Automatic generation of software interfaces for supporting decision-making processes. An application of domain engineering & machine learning

Ph.D. Thesis

Ph.D. Programme of Computer Science
Department of Computer Science and Automation
University of Salamanca (https://ror.org/02f40zc51)
Salamanca, Spain

Candidate
Andrea Vázquez Ingelmo

Supervisors
Francisco José García-Peñalvo, Ph.D.
Roberto Therón Sánchez, Ph.D.

Francisco José García-Peñalvo, Ph.D., Full Professor in the Department of Computer Science and Automation at the University of Salamanca (Spain) and Roberto Therón Sánchez, Ph.D., Full Professor in the same department and university, as supervisors of the Ph.D. thesis entitled "Automatic generation of software interfaces for supporting decision-making processes. An application of domain engineering and machine learning" developed by Andrea Vázquez Ingelmo

Hereby declare that

This Ph.D. thesis, developed in the context of the Ph.D. Programme of Computer Science of the Department of Computer Science and Automation at the University of Salamanca (https://ror.org/02f40zc51), presents enough merits (theoretical and practical) evaluated through the proper assessment, publications, and original proposals to be presented and defended publicly.

Salamanca, Spain, June 2022.

Fran

Firmado digitalmente por GARCIA PEÑALVO FRANCISCO JOSE - 07862615L
Fecha: 2022.12.06 20:57:31 +01'00'

THERON SANCHEZ ROBERTO - 07976246F

Digitally signed by THERON SANCHEZ ROBERTO - 07976246F
Date: 2022.12.07 13:53:11 +01'00'

Francisco José García-Peñalvo, Ph.D.
University of Salamanca

Roberto Therón Sánchez, Ph.D.
University of Salamanca

Automatic generation of software interfaces for supporting decision-making processes. An application of domain engineering & machine learning

Ph.D. Thesis

Ph.D. Programme of Computer Science

Department of Computer Science and Automation

University of Salamanca (https://ror.org/02f40zc51)

Salamanca, Spain

Supervisors

Francisco José García-Peñalvo, Ph.D.

Fran

Firmado digitalmente por GARCIA PEÑALVO FRANCISCO JOSE - 07862615L
Fecha: 2022.12.06 20:58:04 +01'00'

Roberto Therón Sánchez, Ph.D.

THERON SANCHEZ ROBERTO - 07976246F

Digitally signed by THERON SANCHEZ ROBERTO - 07976246F
Date: 2022.12.09 08:53:55 +01'00'

Candidate

Andrea Vázquez Ingelmo

Firmado digitalmente por VAZQUEZ INGELMO ANDREA - 70918831X
Fecha: 2022.12.03 14:25:53 +01'00'

June 2022

Acknowledgments

First, I want to thank my thesis supervisors, from who I learned —and continue learning— the most. This would not have been possible without the support and guidance, both at professional and personal levels, of my brilliant colleagues (and friends) at the GRIAL Research Group, where I basically found my passion, and I cannot be thankful enough for it.

To my amazing family —grandma, aunts, uncles, cousins… and pets—, especially my parents, who always give me all the support —and meals— anyone could wish for.

And to my awesome friends; to Bea, an unparalleled source of good times; the same applies to Marta and our never-ending talks —specially about our addiction to board games and restaurants— and, of course, to our Blasco Holidays with Marina, undoubtedly one of my favorite times of the year. To Isa and our random chats about literally anything, and to Irene, and our therapeutical voice messages. To Misol and the quality time we spent whenever we are together, particularly at the faculty's cafeteria —special mention to Vicente—. To Dena and the countless times she has made me laugh just by laughing herself. To the outstanding plans with Clara, Raquel, and Carmen, and to the cooking days with Laura and Jano —where they cook, and I watch—.

In short, to all the people who have been, explicitly or implicitly, part of my life during this period.

I also want to mention a person that has been following my work before I even started the Ph.D. degree —probably because I told her about it during an interminable thread of direct messages on Twitter—. It seems destiny wanted to join us again before I could end this thesis for me to mention you in these acknowledgments, MJ ♡.

Each of you has made this journey incredibly easy.

Finally, to all those people who indirectly helped me to clear my mind in the meantime: Avril, Nicholas, Angelina, Lydia, Jennifer, Priyanka, Claire, Victoria, Sophie, Deena, Danielle, Nicole, Milagros, Kristen, Regina, Drew, Samantha, David, Lalisa, Cameron, Joe, Alba, Kevin, Blair, Vincent, Charlotte, Paul, Robert, Robyn, Michael, Chelo, María, Ronald, Lucy, and a long etc.

Abstract

Data analysis is a key process to foster knowledge generation in particular domains or fields of study. With a strong informative foundation derived from the analysis of collected data, decision-makers can make strategic choices with the aim of obtaining valuable benefits in their specific areas of action. However, given the steady growth of data volumes, data analysis needs to rely on powerful tools to enable knowledge extraction.

Information dashboards offer a software solution to analyze large volumes of data visually to identify patterns and relations and make decisions according to the presented information. But decision-makers may have different goals and, consequently, different necessities regarding their dashboards. Moreover, the variety of data sources, structures, and domains can hamper the design and implementation of these tools.

This Ph.D. Thesis tackles the challenge of improving the development process of information dashboards and data visualizations while enhancing their quality and features in terms of personalization, usability, and flexibility, among others.

Several research activities have been carried out to support this thesis. First, a systematic literature mapping and review was performed to analyze different methodologies and solutions related to the automatic generation of tailored information dashboards. The outcomes of the review led to the selection of a model-driven approach in combination with the software product line paradigm to deal with the automatic generation of information dashboards.

In this context, a meta-model was developed following a domain engineering approach. This meta-model represents the skeleton of information dashboards and data visualizations through the abstraction of their components and features and has been the backbone of the subsequent generative pipeline of these tools.

The meta-model and generative pipeline have been tested through their integration in different scenarios, both theoretical and practical. Regarding the

theoretical dimension of the research, the meta-model has been successfully integrated with other meta-model to support knowledge generation in learning ecosystems, and as a framework to conceptualize and instantiate information dashboards in different domains.

In terms of the practical applications, the focus has been put on how to transform the meta-model into an instance adapted to a specific context, and how to finally transform this later model into code, i.e., the final, functional product. These practical scenarios involved the automatic generation of dashboards in the context of a Ph.D. Programme, the application of Artificial Intelligence algorithms in the process, and the development of a graphical instantiation platform that combines the meta-model and the generative pipeline into a visual generation system.

Finally, different case studies have been conducted in the employment and employability, health, and education domains. The number of applications of the meta-model in theoretical and practical dimensions and domains is also a result itself. Every outcome associated to this thesis is driven by the dashboard meta-model, which also proves its versatility and flexibility when it comes to conceptualize, generate, and capture knowledge related to dashboards and data visualizations.

Keywords: Data Visualization, Information Visualization, Information Dashboards, Model-Driven Development, Model-Driven Architecture, Software Product Lines, Meta-Modeling, Knowledge Discovery, Graphical User Interfaces, Human-Computer Interaction.

Table of Contents

Table of Figures

Table of Tables

1 Introduction

Our world is data. Our lives are data. Our actions are data. Everything we do, everything we say, everything we think is data. Even the most mundane process is led by data we collect through our senses and data we compute through our brains.

While technical concepts like "data-driven" and "decision-making" can be seen as business jargon, it is necessary to be aware that these concepts influence every aspect of our day-to-day. We carry out decision-making processes when organizing our time to catch the bus promptly or when looking at weather reports to decide if we should bring an umbrella with us on a cloudy day. We are, consciously or not, making data-driven decisions constantly.

But although we face these decisions every day, understanding data is not a simple task nor a straightforward process that can be taken for granted. Several mechanisms are triggered [1] to assist us in the journey of generating brand-new knowledge from tiny morsels of data [2-4]. However, these mechanisms are not infallible. In fact, living in a highly connected society, where data streams are continuously generated, can hamper these processes due to information overload [5, 6].

In this context, technological support is crucial to ease the generation of knowledge in such a convoluted environment with significant quantities of data. Data visualizations and information dashboards are powerful allies when it comes to understanding complex datasets [7-12]. These tools consist of visual displays where raw data is transformed and mapped to visual elements through their properties, such as their position, color, size, etc.

Still, the necessity of drawing conclusions quickly to be part of the current debate can lower our guard when it comes to gaining quality insights from data. This context is a breeding ground for unfair practices that take advantage of this need for immediacy by introducing fake facts, for instance. Moreover, our perception mechanisms can also be tricked –purposely or not– to gain wrong conclusions by taking advantage of cognitive biases [13] or even manipulated data [7, 14].

Data analysis tools such as data visualizations and dashboards must be aware of all these potential issues and overcome them to provide the best experience and insightful, honest outcomes from data. But how can we efficiently introduce these notions and concepts into the design and development processes of these tools?

1.1 Motivation of this research

As introduced, data visualizations and information dashboards are powerful but also complex. A lot of elements need to be involved to deliver effective visual displays of information.

Data visualizations and dashboards are, in the end, a set of visual elements arranged and characterized following the input data. But these elements influence each other, and there are concepts that are not even on the screen but are crucial, such as the user characteristics. Due to this complexity, it is important to rely on expert knowledge when developing data visualizations and dashboards.

However, not every time that users want to create a visualization a domain expert will be available for them to guide the process and apply appropriate design principles.

Several tools have tried to tackle this issue by assisting the user and automating the generation of data visualizations and dashboards through the implementation of generative processes that capture and apply experts' knowledge adapting the visual elements depending on the data and context. This is the case with commercial tools like Tableau, Microsoft Excel, Google Charts, etc.

Although these platforms are very powerful, there is still a problem related to transferring the expert knowledge to practitioners, and with the expressiveness of the obtained visualizations (**Figure 1**).

On the other hand, declarative and imperative programming libraries can improve the expressiveness of the developed data visualizations and dashboards, but, in this case, they usually come with a steep learning curve that hampers the implementation process.

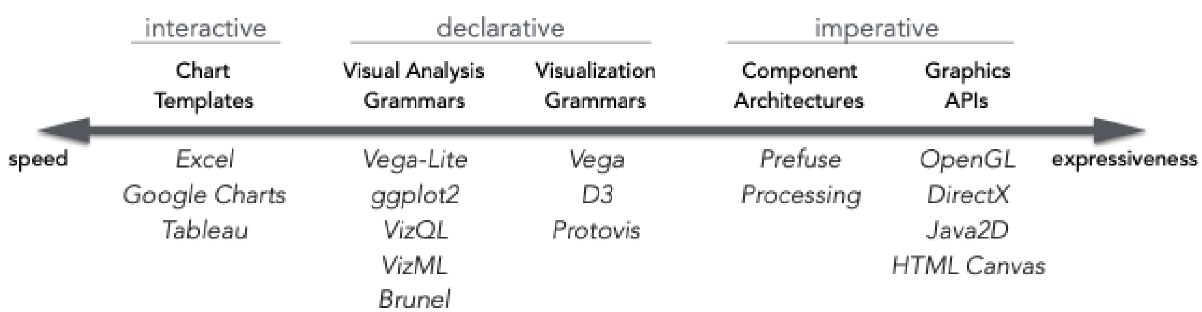


Figure 1. A spectrum of data visualization tools. Source: [15].

In this sense, a generative dashboard pipeline should merge the best of interactive systems and programming languages, offering a good experience for non-expert users, but also a powerful specification to understand every element involved in the final product.

1.2 Context of this research

This thesis has been developed in the GRoup of InterAction and e-Learning (GRIAL) [16, 17] research group and in the context of the Ph.D. Programme of Computer Science of the Department of Computer Science and Automation at the University of Salamanca.

GRIAL is a Recognized Research Group of the University of Salamanca and a Recognized Group of Excellence by the Regional Council of Castille and León. The group is formed by many researchers from different fields of knowledge. Most members have a technical or a pedagogical profile, but there are also members with expertise in e-Learning project management, Humanities, Sciences, etc.

The research activity of the group in these last few years has ranged from purely technical and computing projects to the development of pedagogical methodologies and models of reference in the field of online learning which have gained international recognition and awards.

The main work lines of the GRIAL research group involve:

- Digital humanities.
- eLearning methodologies.
- ICT and educational innovation.
- Information science.
- Interactive learning systems.
- Learning Technologies.
- Quality and assessment in education.
- Social responsibility and inclusion.
- Strategic management of knowledge and technology.
- Technological ecosystems.
- Visual analytics.

- Web engineering and software architecture.

The present thesis is the result of multiple projects in which the author has been involved since joining the GRIAL research group in 2016, namely:

- ***Technological ecosystem for the University Employability and Employment Observatory (OEEU) of the UNESCO Chair in University Management and Policy of the Polytechnic University of Madrid*** [18-22]. Collaboration from 2014 to 2018 with the UNESCO Chair of University Management and Policy of the Polytechnic University of Madrid for the implementation of the technological ecosystem for the University Employability and Employment Observatory, with which the I Barometer of University Employability and Employment in Spain [23] and the Barometer of University Employability and Employment (Master's Edition 2017) [24] have been developed, both of them funded by the Fundación la Caixa.
- ***TE-CUIDA, proposal of a TEchnological Ecosystem to support caregivers (ref. SA061P17)*** [25, 26]. Project funded by the Ministry of Education of the Junta de Castilla y León in the program of support for research projects co-financed by the European Regional Development Fund. Its duration is from 26-7-2017 to 31-12-2019. Resolution ORDER EDU/986/2017, November 8. It seeks to provide support to caregivers, both formal and informal, to improve the quality of care and even reduce the caregiver's burden, thereby facilitating that the elderly person, especially if they have loss of autonomy, can maintain their residence in the community environment and in their own home and maintain the best possible care possible.
- ***A Digital Ecosystem Framework for an Interoperable NEtwork-based Society. (DEFINES) (ref. TIN2016-80172-R)*** [27-32]. Project funded by the Ministry of Economy and Competitiveness in the 2016 call for R&D&I projects of the State Program for Research, Development and Innovation Oriented to the Challenges of Society, with a duration from 1-1-2017 to 31-12-2020. It pursues two main objectives. On the one hand, to propose a

technological ecosystem to support services for corporate knowledge management. On the other hand, to transform the current knowledge management processes and achieve a better adaptation to the context of the Digital Society. Digital Society.

- ***Visual Analytics and Machine Learning for decision-making in Health ecosystems. (AVisSA) (ref. PID2020-118345RB-I00)*** [33-35]. Project funded by the Ministry of Science and Innovation in the 2020 call for R&D&I projects of the State Program for Research, Development, and Innovation 2017-2020, with a duration from 1-10-2021 to 30-09-2025. This project aims at tackling the development of a system of automatic dashboard generation (meta-dashboard) with Domain Engineering and Artificial Intelligence (AI) techniques, to obtain dashboards adapted to each case and application domain from the flow of data in technological ecosystems that automatically adapts to the needs of analysis and knowledge management in heterogeneous contexts. The medical domain is taken as a reference due to its complexity and the diversity of information management needs, which appear in the different medical specialties, to improve these processes within the health system, with a remarkable impact on the decision-making processes.
- ***Design and implementation of a technological ecosystem for research and intelligent data analysis in the Cardiology Department of the Hospital Clínico Universitario de Salamanca*** [35-40]. Collaboration since 2019 with the Cardiology Department of the Hospital Clínico Universitario de Salamanca for the implementation of different platforms related to data management and AI applications in the medical domain.

In terms of the doctoral theses developed within the GRIAL research group, two of them should be highlighted because of their close relationship with the present research.

First, the research carried out by Juan Cruz-Benito "On data-driven systems analyzing, supporting and enhancing Human-Computer Interaction", supervised by Dr. Francisco José García-Peñalvo and Dr. Roberto Therón-Sánchez, both from the University of Salamanca. The main objective of this work is to explore how the collection and analysis of user-computer interaction information can be performed to improve such interaction in several typical scenarios: highly interactive and immersive scenarios, scenarios with many users, scenarios with excessive information and high task complexity [41, 42].

Second, the research carried out by Alicia García-Holgado "Integration analysis of solutions based on software as a service to implement Educational Technological Ecosystems", supervised by Dr. Francisco José García-Peñalvo from the University of Salamanca. This research is focused on providing an architectural framework that allows improving the definition, development, and sustainability of technological ecosystems for learning through model driven engineering [43-46]. This work has also been crucial due to the decision of following a model driven approach during the whole development process of this thesis.

Regarding the PhD Program in Computer Science (https://doctorado.usal.es/es/doctorado/ingenier%C3%ADa-inform%C3%A1tica), it has been proposed by the Department of Computer Science and Automation in collaboration with the Department of Applied Mathematics of the University of Salamanca. The programme aims at training in different areas of research in the field of Computer Science, especially those related to the research lines of the groups that are integrated in the proposal such as: Intelligent systems, software engineering and knowledge engineering, human-computer interaction, data mining, visual analytics and information visualization, robotics, intelligent control, cryptography and information security, mathematical modeling, and numerical analysis.

Two research stays were carried out during the development of this thesis. First, a virtual internship from July 1, 2021, to October 10, 2021, at Østfold University College, Computer Science Department (Halden, Norway). This research stay was focused on validating the meta-model.

Second, as part of the international research stay required to obtain the International Mention for the Ph.D., the author was a visiting scholar at the Department of Computer Graphics Technology of Purdue University (West Lafayette, Indiana, United States of America) from January 10, 2022, to April 14, 2022. The research was related to data visualization applications and the results can be consulted in the last case study of this thesis.

Moreover, the author received three awards for publications related to her thesis:

1. Best paper award in the track International Workshop on Software Engineering for E-Learning (ISELEAR’17) within the International Conference on Technological Ecosystems for Enhancing Multiculturality (TEEM) 2017 held in Cádiz, Spain between October 18-20, 2017. Award granted for the paper “Improving the OEEU's data-driven technological ecosystem's interoperability with GraphQL” developed jointly to J. Cruz-Benito and F. J. García-Peñalvo.
2. Best paper award in the track International Workshop on Software Engineering for E-Learning (ISELEAR’18) within the International Conference on Technological Ecosystems for Enhancing Multiculturality (TEEM) 2018 held in Salamanca, Spain between October 24-26, 2018. Award granted for the paper “Domain engineering for generating dashboards to analyze employment and employability in the academic context” developed jointly to F. J. García-Peñalvo and R. Therón.
3. Best paper award in the track International Workshop on Software Engineering for E-Learning (ISELEAR’19) within the International Conference on Technological Ecosystems for Enhancing Multiculturality (TEEM) 2019 held in León, Spain between October 16-18, 2019. Award granted for the paper “Capturing high-level requirements of information dashboards’ components through meta-modeling” developed jointly to F. J. García-Peñalvo and R. Therón.

4. Conference best paper of the Learning Analytics Summer Institute (LASI Spain) Salamanca, Spain between June 20-21, 2022. Award granted for the paper "A proposal to measure the understanding of data visualization elements in visual analytics applications" developed jointly to F. J. García-Peñalvo, R. Therón, V. Byrd, and J. D. Camba.

Finally, from the economic point of view, this doctoral thesis has received funding from the Spanish *Ministry of Education and Vocational Training* under an FPU fellowship (FPU17/03276).

1.3 Hypotheses and goals

This research aims at exploring the benefits, downsides, and applications of the automatization of information dashboards and data visualizations development processes. Although automatizing processes could yield several gains in different dimensions, it is crucial to understand how automatizing the implementation of dashboards influences their performance as well as their functional and non-functional features.

After all these considerations, the main hypothesis of this work is stated as follows:

H1. Automatizing the development of tailored user interfaces for supporting decision-making processes increments their benefits in terms of **functional** and **non-functional features.**

In other words, the primary goal of the research is to design and implement a **generative framework** for the automatic and **systematic development of information dashboards**, as well as to **discuss the insights** reached from automatizing the generation of these tools. The generative framework needs to involve **tailoring mechanisms** to adapt the layout, visual design, data sources, and interaction mechanisms. Through this approach, the focus is on fostering individualization, usability, and flexibility to **maximize the benefits** derived from the generated tools.

A series of sub-objectives have been posed to reach the main goal. These sub-objectives can be categorized into four main phases:

1. **Conceptualization:**
    - Identification of common characteristics of information dashboards at a meta-level.
    - Identification of connection mechanisms to enable a model-driven approach to build concrete products of the Software Product Line (SPL).
2. **Implementation:**
    - Implementation of mechanisms that foster interoperability to allow the connection of different data sources.
    - Definition and implementation of reusable and configurable core assets to generate specific products of the SPL.
3. **Validation:**
    - Evaluation of the SPL at a generative and functional level.
    - Evaluation of the generated dashboards in terms of usability and expressiveness.
    - Study of the automatic adaptation of the dashboards through AI mechanisms.
4. **Applications:**
    - Study of the integration of the dashboards SPL within different technological ecosystems and case studies.

The outcomes of each phase have been published in different research articles that will be summarized throughout this thesis. **Figure 2** shows a conceptual map of the phases and their related research. As can be observed, the conceptualization-implementation-validation-application processes are iterative, following the action-research methodology, which will be detailed in the following subsection. The continuous evaluation of the outcomes at every stage has resulted in the improvement of the meta-model and the generative pipeline of information dashboards.

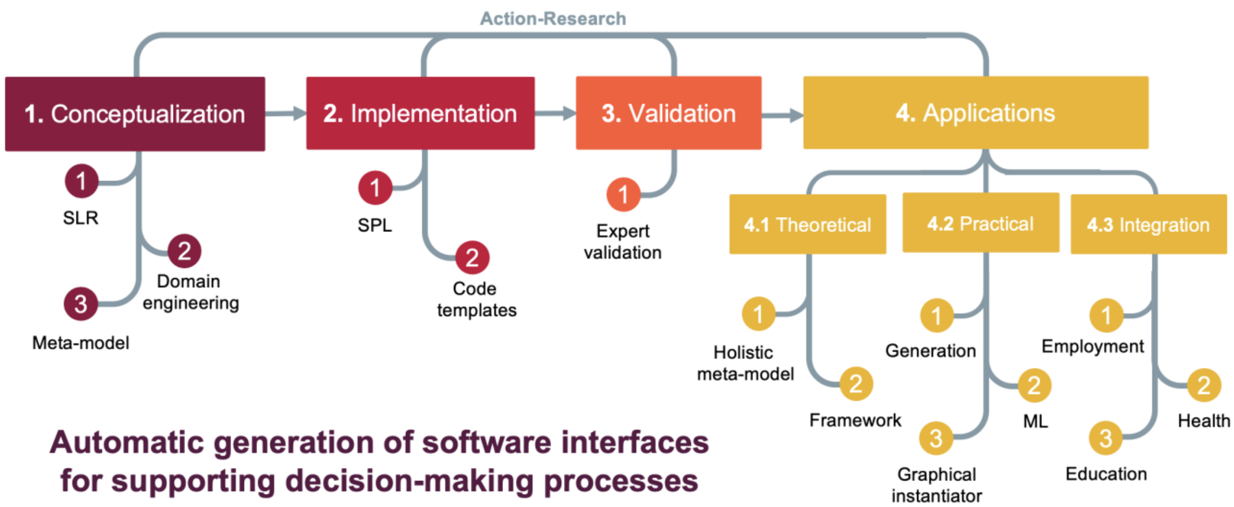


Figure 2. Conceptual map of the objectives and associated research. Source: own elaboration.

Figure 2 presents the previous objectives and a classification of the outcomes from each sub-objective in terms of the publications that are part of this **thesis driven by the articles published during its development**:

1. **Conceptualization:**

    1.1. Systematic Literature Review (SLR):

    - A. Vázquez-Ingelmo, F. J. García-Peñalvo and R. Therón, "Tailored information dashboards: A systematic mapping of the literature," in *Proceedings of the XX International Conference on Human Computer Interaction*

(Donostia, Gipuzkoa, Spain — June 25 - 28, 2019) Article Number 26, New York, NY, USA: ACM, 2019. doi: 10.1145/3335595.3335628 [47].

- A. Vázquez-Ingelmo, F. J. García-Peñalvo, and R. Therón, "Information Dashboards and Tailoring Capabilities - A Systematic Literature Review," *IEEE Access,* vol. 7, pp. 109673-109688, 2019, doi: 10.1109/ACCESS.2019.2933472 [48].

1.2. Application of the domain engineering paradigm to the dashboards' domain[1]:

- A. Vázquez-Ingelmo, F. J. García-Peñalvo and R. Therón, "Domain engineering for generating dashboards to analyze employment and employability in the academic context," in *TEEM'18 Proceedings of the Sixth International Conference on Technological Ecosystems for Enhancing Multiculturality (Salamanca, Spain, October 24th-26th, 2018)*, F. J. García-Peñalvo, Ed. pp. 896-901, New York, NY, USA: ACM, 2018. doi: 10.1145/3284179.3284329 [49].

1.3. Construction of the meta-model for information dashboards and visualizations:

- A. Vázquez-Ingelmo, F. J. García-Peñalvo and R. Therón, "Capturing high-level requirements of information dashboards' components through meta-modeling," in *TEEM'19 Proceedings of the Seventh International Conference on Technological Ecosystems for Enhancing Multiculturality (Leon, Spain, October 16th-18th, 2019)*, M. Á. Conde-González, F. J. Rodríguez-Sedano, C. Fernández-Llamas and F. J. García-Peñalvo, Eds. ICPS: ACM International Conference Proceedings Series, pp. 815-821, New York, NY, USA: ACM, 2019. doi: 10.1145/3362789.3362837 [50].

- A. Vázquez Ingelmo, F. J. García-Peñalvo, R. Therón Sánchez, and M. Á. Conde González, "Extending a dashboard meta-model to account for users'

[1] This publication appears as a case study in the domain of employment and employability, however, it is also framed within the conceptualization phase, so it supports both sections of this work.

characteristics and goals for enhancing personalization,", *Proceedings of LASI-SPAIN 2019. Learning Analytics Summer Institute Spain 2019: Learning Analytics in Higher Education (Vigo, Spain, June 27-28, 2019). CEUR Workshop Proceedings Series,* 2019. [Online]. Available: http://hdl.handle.net/10366/139803 [51].

- A. Vázquez-Ingelmo, F. J. García-Peñalvo, R. Therón, and M. Á. Conde, "Representing Data Visualization Goals and Tasks through Meta-Modeling to Tailor Information Dashboards," *Applied Sciences*, vol. 10, no. 7, 2306, 2020. [Online]. Available: https://www.mdpi.com/2076-3417/10/7/2306 [52].

- A. Vázquez-Ingelmo, F. J. García-Peñalvo, R. Therón, and A. García-Holgado, "Specifying information dashboards' interactive features through meta-model instantiation," in Proceedings of LASI-SPAIN 2020. Learning Analytics Summer Institute Spain 2020: Learning Analytics. Time for Adoption? (Valladolid, Spain, June 15-16, 2020), A. Martínez-Monés, A. Álvarez, M. Caeiro-Rodríguez, and Y. Dimitriadis Eds., (CEUR Workshop Proceedings Series, no. 2671). Aachen, Germany: CEUR-WS.org, 2020, pp. 47-59 [53].

- A. Vázquez-Ingelmo, A. García-Holgado, F. J. García-Peñalvo and R. Therón, "A Meta-modeling Approach to Take into Account Data Domain Characteristics and Relationships in Information Visualizations," in *Trends and Innovations in Information Systems and Technologies, WorldCIST 2021*, vol. 2, Á. Rocha, H. Adeli, G. Dzemyda, F. Moreira and A. M. Ramalho Correia, Eds. Advances in Intelligent Systems and Computing Series Series, no. 1366, pp. 570-580, Cham, Switzerland: Springer Nature, 2021. doi: 10.1007/978-3-030-72651-5_54 [54].

2. **Implementation:**

   2.1. Application of the SPL paradigm following the obtained meta-model entities and relationships:

- A. Vázquez-Ingelmo, F. J. García-Peñalvo and R. Therón, "Automatic generation of software interfaces for supporting decision-making processes. An application of domain engineering and machine learning," in *TEEM'19 Proceedings of the Seventh International Conference on Technological Ecosystems for Enhancing Multiculturality (Leon, Spain, October 16th-18th, 2019)*, M. Á. Conde-González, F. J. Rodríguez-Sedano, C. Fernández-Llamas and F. J. García-Peñalvo, Eds. ICPS: ACM International Conference Proceedings Series, pp. 1007-1011, New York, NY, USA: ACM, 2019. doi: 10.1145/3362789.3362923 [55].

2.2. Code templates as a method to materialize the SPL variability points:

- A. Vázquez-Ingelmo, F. J. García-Peñalvo and R. Therón, "Addressing Fine-Grained Variability in User-Centered Software Product Lines: A Case Study on Dashboards," in *Knowledge in Information Systems and Technologies*, vol. 1, Á. Rocha, H. Adeli, L. P. Reis and S. Costanzo, Eds. Advances in Intelligent Systems and Computing, no. AISC 930, pp. 855-864, Switzerland: Springer Nature, 2019. doi: 10.1007/978-3-030-16181-1_80 [56].

3. **Validation:**

3.1. Expert validation of the dashboard meta-model:

- A. Vázquez-Ingelmo, A. García-Holgado, F. J. García-Peñalvo, R. Therón, and R. Colomo-Palacios, "Content-validation questionnaire of a meta-model to ease the learning of data visualization concepts," presented at the Learning Analytics Summer Institute Spain 2022 (LASI Spain 22), Salamanca, Spain, 20-21 June, 2022 [57].

4. **Applications:**

4.1. Theoretical applications:

4.1.1. Integration of different meta-models: holistic meta-model:

- A. Vázquez-Ingelmo, A. García-Holgado, F. J. García-Peñalvo, and R. Therón, "A meta-model to develop learning ecosystems with support for knowledge discovery and decision-making processes," in 2020 15th Iberian Conference on Information Systems and Technologies (CISTI), 24-27 June 2020 2020, pp. 1-6, doi: 10.23919/CISTI49556.2020.9140986 [58].

- A. Vázquez-Ingelmo, A. García-Holgado, F. J. García-Peñalvo and R. Therón, "A Dashboard to Support Decision-Making Processes in Learning Ecosystems: A Metamodel Integration," in *Proceedings of the 2020 European Symposium on Software Engineering - ESSE 2020 (November 6-8, 2020, Rome, Italy).* International Conference Proceedings Series, pp. 80-87, New York, NY, USA: ACM, 2020. doi: 10.1145/3393822.3432326 [59].

- A. Vázquez-Ingelmo, A. García-Holgado, F. J. García-Peñalvo, and R. Therón, "A Meta-Model Integration for Supporting Knowledge Discovery in Specific Domains: A Case Study in Healthcare," Sensors, vol. 20, no. 15, 4072, 2020. [Online]. Available: https://www.mdpi.com/1424-8220/20/15/4072 [60].

4.1.2. Using the meta-model as a framework:

- A. Vázquez-Ingelmo, A. García-Holgado, F. J. García-Peñalvo, and R. Therón, "Dashboard Meta-Model for Knowledge Management in Technological Ecosystem: A Case Study in Healthcare," *Proceedings,* vol. 31, no. 1, 2019, doi: 10.3390/proceedings2019031044 [61].

- A. Vázquez-Ingelmo, F. J. García-Peñalvo and R. Therón, "Aggregation Bias: A Proposal to Raise Awareness Regarding Inclusion in Visual Analytics," in *Trends and Innovations in Information Systems and Technologies, WorldCIST 2020*, vol. 3, Á. Rocha, H. Adeli, L. P. Reis, S. Costanzo, I. Orovic and F. Moreira, Eds. Advances in Intelligent Systems and Computing Series Series, no. AISC 1161, pp. 409-417, Cham, Switzerland: Springer Nature, 2020. doi: 10.1007/978-3-030-45697-9_40 [62].

- A. Vázquez-Ingelmo, F. J. García-Peñalvo, R. Therón, D. Amo Filvà, and D. Fonseca Escudero, "Connecting domain-specific features to source code: towards the automatization of dashboard generation," Cluster Computing, vol. 23, no. 3, pp. 1803-1816, 2020, doi: 10.1007/s10586-019-03012-1 [63].

- A. Vázquez-Ingelmo, F. J. García-Peñalvo, and R. Therón, "Towards a Technological Ecosystem to Provide Information Dashboards as a Service: A Dynamic Proposal for Supplying Dashboards Adapted to Specific Scenarios," Applied Sciences, vol. 11, no. 7, art. 3249, 2021, doi: 10.3390/app11073249. [64]

4.2. Practical applications:

4.2.1. Generation of information dashboards:

- A. Vázquez-Ingelmo and R. Therón, "Beneficios de la aplicación del paradigma de líneas de productos software para generar dashboards en contextos educativos," RIED. Revista Iberoamericana de Educación a Distancia, vol. 23, no. 2, pp. 169-185, 2020, doi: 10.5944/ried.23.2.26389 [65].

- A. Vázquez-Ingelmo, F. J. García-Peñalvo and R. Therón, "Generating Dashboards Using Fine-Grained Components: A Case Study for a PhD Programmes," in *Learning and Collaboration Technologies. Design, Experiences. 7th International Conference, LCT 2020, Held as Part of the 22nd HCI International Conference, HCII 2020, Copenhagen, Denmark, July 19–24, 2020, Proceedings, Part I*, P. Zaphiris and A. Ioannou, Eds. Lecture Notes in Computer Science, no. 12205, pp. 303-314, Cham, Switzerland: Springer Nature, 2020. doi: 10.1007/978-3-030-50513-4_23 [66].

- A. Vázquez Ingelmo, A. García-Holgado, H. Hernández-Payo, F. J. García-Peñalvo, and R. Therón Sánchez, "Following up the progress of doctoral students and advisors' workload through data visualizations: a case study in a PhD program," *Proceedings of LASI-SPAIN 2021. Learning Analytics Summer Institute Spain 2021: Learning Analytics in times of COVID-19: Opportunity from crisis (Barcelona, Spain, July 7-9, 2021). CEUR Workshop Proceedings Series,* 2021. [Online]. Available: http://ceur-ws.org/Vol-3029/paper06.pdf [67].

4.2.2. Application of Machine Learning to the generation of information dashboards:

- Vázquez-Ingelmo, A., García-Holgado, A., García-Peñalvo, F. J., & Therón, R. "Proof-of-concept of an information visualization classification approach based on their fine-grained features," *Expert Systems*, e12872, 2022, doi: 10.1111/exsy.12872 [68].

4.2.3. Design and implementation of a graphical instantiation platform based on the dashboard meta-model and following the model-driven architecture

- A. Vázquez Ingelmo, F. J. García-Peñalvo, and R. Therón, "MetaViz - A graphical meta-model instantiator for generating information dashboards and visualizations," Journal of King Saud University - Computer and Information Science, In Press, doi: https://doi.org/10.1016/j.jksuci.2022.09.015 [69].

4.3. Integration in real-world environments:

4.3.1. Integration of the generative pipeline of dashboards in the employment and employability context:

- A. Vázquez-Ingelmo, J. Cruz-Benito, F. J. García-Peñalvo, and M. Martín-González, "Scaffolding the OEEU's Data-Driven Ecosystem to Analyze the Employability of Spanish Graduates," in *Global Implications of*

Emerging Technology Trends, F. J. García-Peñalvo Ed. Hershey, PA, USA: IGI Global, 2018, pp. 236-255 [18].

- A. Vázquez-Ingelmo, J. Cruz-Benito, and F. J. García-Peñalvo, "Improving the OEEU's data-driven technological ecosystem's interoperability with GraphQL," in *Proceedings of the 5th International Conference on Technological Ecosystems for Enhancing Multiculturality*, Cádiz, Spain, 2017, New York, NY, USA: Association for Computing Machinery, p. Article 89, doi: 10.1145/3144826.3145437 [21].
- J. Cruz-Benito, J. C. Sánchez-Prieto, A. Vázquez-Ingelmo, R. Therón, F. J. García-Peñalvo, and M. Martín-González, "How Different Versions of Layout and Complexity of Web Forms Affect Users After They Start It? A Pilot Experience," Cham, 2018: Springer International Publishing, in Trends and Advances in Information Systems and Technologies, pp. 971-979 [19].
- J. Cruz-Benito, A. Vázquez-Ingelmo, J. C. Sánchez-Prieto, R. Therón, F. J. García-Peñalvo, and M. Martín-González, "Enabling Adaptability in Web Forms Based on User Characteristics Detection Through A/B Testing and Machine Learning," *IEEE Access,* vol. 6, pp. 2251-2265, 2018, doi: 10.1109/ACCESS.2017.2782678 [41].
- A. Vázquez-Ingelmo, F. J. García-Peñalvo, and R. Therón, "Domain engineering for generating dashboards to analyze employment and employability in the academic context," presented at the Proceedings of the Sixth International Conference on Technological Ecosystems for Enhancing Multiculturality, Salamanca, Spain, 2018. [Online]. Available: https://doi.org/10.1145/3284179.3284329 [49].
- A. Vázquez-Ingelmo, F. J. García-Peñalvo, and R. Therón, "Taking advantage of the software product line paradigm to generate customized user interfaces for decision-making processes: a case study on university

employability," PeerJ Computer Science, vol. 5, e203, 2019, doi: 10.7717/peerj-cs.203 [70].

4.3.2. Integration of the generative pipeline of dashboards in the health context:

- A. Vázquez-Ingelmo *et al.*, "A platform for management and visualization of medical data and medical imaging," in *Proceedings TEEM'20. Eighth International Conference on Technological Ecosystems for Enhancing Multiculturality (Salamanca, Spain, October 21st - 23rd, 2020)*, F. J. García-Peñalvo, Ed. ICPS: ACM International Conference Proceedings Series, New York, NY, USA: ACM, 2020. doi: 10.1145/3434780.3436652. [40].

- F. J. García-Peñalvo *et al.*, "Application of Artificial Intelligence Algorithms Within the Medical Context for Non-Specialized Users: the CARTIER-IA Platform," *International Journal of Interactive Multimedia and Artificial Intelligence*, vol. 6, no. 6, pp. 46-53, 2021, doi: 10.9781/ijimai.2021.05.005 [36].

- A. García-Holgado *et al.*, "User-Centered Design Approach for a Machine Learning Platform for Medical Purpose," in *HCI-COLLAB 2021*, Sao Paulo, Brazil, 8-10, September 2021, Cham, Switzerland: Springer International Publishing, in Human-Computer Interaction, pp. 237-249, doi: 10.1007/978-3-030-92325-9_18 [38].

- A. Vázquez-Ingelmo *et al.*, "Bringing Machine Learning Closer to Non-Experts: Proposal of a User-Friendly Machine Learning Tool in the Healthcare Domain," in *Proceedings TEEM'21. Ninth International Conference on Technological Ecosystems for Enhancing Multiculturality (Barcelona, Spain, October 27th – 29th, 2021)* ICPS: ACM International Conference Proceedings Series, pp. 324-329, New York, USA: ACM, 2021. doi: 10.1145/3486011.3486469 [39].

- A. Vázquez-Ingelmo *et al.*, "A platform to support the visual analysis of the SALMANTICOR study outcomes: conveying cardiological data to lay users," in *Proceedings TEEM'21. Ninth International Conference on Technological Ecosystems for Enhancing Multiculturality (Barcelona, Spain, October 27th – 29th, 2021).* ICPS: ACM International Conference Proceedings Series, pp. 335-341, New York, USA: ACM, 2021. doi: 10.1145/3486011.3486471 [37].

- F. J. García-Peñalvo, A. Vázquez-Ingelmo, and A. García-Holgado, "Fostering Decision-Making Processes in Health Ecosystems through Visual Analytics and Machine Learning," presented at the 9th International Conference on Learning and Collaboration Technologies, Virtual, June 28, 2022 [34].

- F. J. García-Peñalvo *et al.*, "KoopaML: A graphical platform for building machine learning pipelines adapted to health professionals," *International Journal of Interactive Multimedia and Artificial Intelligence,* In Press [35].

4.3.3. Integration of the generative pipeline of dashboards in the education context:

- A. Vázquez-Ingelmo, F. J. García-Peñalvo, R. Therón, V. Byrd, and J. D. Camba, "A proposal to measure the understanding of data visualization elements in visual analytics applications," presented at the Learning Analytics Summer Institute Spain 2022 (LASI Spain 22), Salamanca, Spain, 20-21 June, 2022 [71].

1.4 Methodology

This subsection covers the methodologies employed throughout the thesis, including the general methodological framework to carry out the research as well as the paradigms followed to design and develop the generative pipeline of information dashboards. An overview of the followed methodology has been published and can be consulted in **Appendix I**. Automatic generation of software interfaces for supporting decision-making processes. An application of domain engineering and machine learning [55].

1.4.1 Action-research methodology

Due to the mixed nature of the artifacts and proposed scenarios of this research, this thesis has been carried out following an iterative process where the knowledge gained through past experiences and the outcomes of the different cycles is crucial for the following stages. The Action-Research methodological framework [72] will be followed to accomplish this process.

Kemmis posed Action-Research [73] as an inquiry method carried out by the participants in social situations with the aim of improving and understanding their own social practices and their contexts.

Later, McTaggart & Kemmis described the characteristics of this methodology. The Action-Research methodology is based on a cyclic spiral of research and actions composed of a series of phases and sequences [74].

Therefore, Action-Research is an iterative process where each cycle provides an output that will be the input for the next cycle.

The iterative nature of the methodology enables the researcher to address previously identified problems, thus obtaining more refined solutions.

However, as previously represented in **Figure 2**, it is necessary to formalize the problem to be addressed to be able to start the Action-Research cycles. Similar problems and previously developed solutions have been studied to understand the context and the current state of the field. The methodology used for this step (a SLR) is detailed in the next section.

Once the problem is formalized, two Action-Research cycles are proposed to develop a proposal for generating dashboards and evaluating them in real contexts. Evaluation is necessary to obtain feedback to improve the proposal.

The chosen framework for software development is an agile approach based on SCRUM [75]. This framework provides the necessary processes, rules, practices, roles, and artifacts to increase the productivity of development teams through an iterative and incremental software development cycle [76].

A mixed methods research approach has been employed to evaluate the artifacts. The research has been conducted using both quantitative and qualitative methods [77], leveraging the two perspectives to obtain a wider view of the results to face the next Action-Research cycles.

1.4.2 Systematic Literature Review

As introduced above, an SLR [78] is a powerful method to gain knowledge about previous solutions and similar problems. The SLR helps in the contextualization of the problem to be solved and provides new research lines by identifying weaknesses and strengths in previous solutions.

The SLR is conducted under the guidelines proposed by Kitchenham [79]. Following the [79, 80] guidelines, the SLR is composed of three main phases: planning, conducting, and reporting the study.

However, before planning the review, a preliminary search was performed to verify that there were no recent reviews about the target topic. If any recent SLR were found, there would not be any necessity to conduct a new one.

This preliminary search was performed using different electronic databases (Scopus, Web of Science (WoS), IEEE Xplorer and Springer) and using terms related to literature reviews ("SLR", "systematic literature review", etc.), as well as terms related to the target of the review ("dashboards").

The result of the previous search confirmed that, at the time of performing the queries, there were not any previous SLR about tailored dashboards, so the necessity of performing a literature review was justified.

1.4.3 Meta-modeling and Software Product Lines

Given the complexity of the dashboards' design processes, it is necessary to understand their domain deeply. Dashboards can present different features, different visual designs, different purposes, etc. However, dashboards also share common features that are always present.

These common features can be abstracted to obtain generic schemas or models that can help with the domain understanding and systematic reuse of software components. The technique for identifying shared properties and variabilities within a specific domain is called domain engineering [81-86].

Domain engineering is based on knowledge reuse regarding some specific domain. This approach is an essential phase of the SPL paradigm [83, 87]. This methodology allows the reuse of software components and their configuration to match certain requirements; that is why identifying common features and variabilities is an essential step.

Once the domain has been studied, it is possible to develop a generic model (a meta-model) that captures every abstract property of dashboards, as well as the relationships among the identified entities.

Meta-models are crucial artifacts in model-driven development (MDD) paradigms [88-90], as they allow mapping entities from high-abstraction levels to more detailed entities and even source code through transformations.

The Object Management Group (OMG) proposes the model-driven architecture (MDA) as a guideline to implement this approach. This architecture provides a framework for software development that employs models to describe and define the target system [91]. The main difference between MDD and MDA is that the OMG proposal uses a set of standards: meta-object facility (MOF), unified modeling language (UML), XML (Extensible Markup Language) metadata interchange (XMI), and query/view/transformation (QVT).

The MDA framework is composed by four architectural layers. Each layer represents a level of abstraction of the represented entities. The most abstract layer (M3 level) is known as the **meta-meta-model level**. This layer defines basic structures and concepts to represent less-abstract layers as well as itself, and it can be implemented with the mentioned MOF standard.

The M2 level, namely **meta-model level**, complies with the meta-meta-model and represents abstract entities and relationships. Meta-models can be seen as Domain Specific Languages (DSL) that express common and generic features of the target domain.

The M1 level, defines **models** that instantiate and specify the abstract features contained in the meta-model, and its syntaxis must comply with the M2 level.

Finally, the M0 level represent real-world applications based on a previously defined M1 model.

Figure 3 summarizes the MDA framework layers.

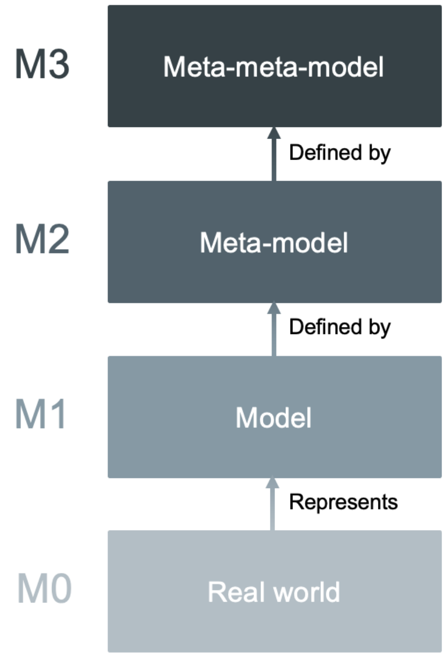


Figure 3. Levels of the MDA framework. Source: own elaboration.

The combination of the MDD and SPL paradigms increases productivity in terms of development processes, but also knowledge reuse, as the study of the domain is represented by the meta-model. Moreover, both methodologies provide mechanisms to address several requirements from different profiles and contexts, which is crucial in this domain.

1.5 Document structure

This manuscript is organized into 6 chapters and 36 appendixes. The present section introduces the dissertation with the research context, motivation, goals, and methodologies followed to carry out this research.

Chapter 2 presents the state-of-the-art of automatic generation of information dashboards with tailoring capabilities. A SLR and mapping were carried out to frame the possibilities of creating a generative pipeline focused on improving the user experience and understanding.

Chapter 3 details one of the main artifacts of this research: the dashboard meta-model. This chapter describes the domain engineering process and the iterations made until reaching the final version of the meta-model.

Chapter 4 presents the results derived from the validation and application of the dashboard meta-model into different contexts, including theoretical, practical, and real-world applications.

Chapter 5 discusses all the obtained results, while Chapter 6 presents the general conclusions of the research, future research lines, and the achievements obtained by the author while carrying out the thesis.

The first 35 appendixes, on the other hand, include every published paper with the results derived from this thesis, while the last appendix (**Appendix AK**. Resumen extendido: Generación automática de interfaces software para el soporte a la toma de decisiones. Aplicación de ingeniería de dominio y machine learning) contains an extended abstract of this document in Spanish.

2 State-of-the-art of information dashboards and tailoring capabilities

As introduced in the previous chapter, tailoring capabilities are vital factors to tackle the fact that there is no "one size fits all" related to information dashboards because not every user has the same knowledge, goals, interests, or preferences when visualizing data.

However, before planning new methodologies for the automatic generation of tailored information dashboards, it is crucial to analyze previous findings that deal with this issue to find caveats and challenges new solutions could address.

For these reasons, a SLR on how tailoring capabilities are achieved in the domain of information dashboards [92-94] has been carried out. This process aims to provide a comprehensive view of existing solutions, their limitations, and methods employed to offer suitable dashboard configurations to specific users.

Apart from the obtained landscape of solutions, the other main outcome of the literature review is a critical analysis of the methodologies and architectures found in the selected papers. This kind of analysis offers a good starting point for designing and implementing the first proposal of a system for the automatic generation of information dashboards.

The mapping and SLR can be consulted at **Appendix G**. Tailored information dashboards: A systematic mapping of the literature [47] and **Appendix H**. Information dashboards and tailoring capabilities – A systematic literature review [48], respectively. This section provides the updated analysis and outcomes of the systematic review[2].

2.1 Methodology

The systematic process to conduct the literature review follows the SLR methodology by Kitchenham and Carters [79, 80]. The SLR has been complemented with a systematic literature mapping following the method proposed in [95]. The mapping results provide a quantitative analysis regarding the state-of-the-art of the target domain.

The SLR comprises three main phases: planning, conducting, and reporting the study [79, 80]. This subsection describes the protocol followed during the SLR to enable the traceability of the outcomes.

2.1.1 Review and planning process

Research questions

The questions raised to analyze the state-of-the-art of tailoring capabilities in information dashboards are organized into three categories: technical aspects (RQ1-RQ4), artificial intelligence (AI) application (RQ5), evaluation of the solutions (RQ6).

- **RQ1.** How have existing dashboard solutions tackled the necessity of tailoring capabilities?

[2] The update was carried out on May 18, 2022.

- **RQ2.** Which methods have been applied to support tailoring capabilities within the dashboards' domain?
- **RQ3.** How do the proposed solutions manage the dashboard's requirements?
- **RQ4.** Can the proposed solutions be transferred to different domains?
- **RQ5.** Has any artificial intelligence approach been applied to the dashboards' tailoring processes, and, if applicable, how have these approaches been involved in the dashboards' tailoring processes?
- **RQ6.** How mature are tailored dashboards regarding their evaluation?

On the other hand, the mapping questions focus on categorizing and analyzing the collected solutions quantitatively.

- **MQ1.** How many studies have been published over the years?
- **MQ2.** Who are the most active authors in the area?
- **MQ3.** What type of papers are published?
- **MQ4.** To which contexts have been the variability processes applied? (BI, learning analytics, etc.).
- **MQ5.** Which are the factors that condition the dashboards' variability process?
- **MQ6.** What is the target of the variability process? (Visual components, KPIs, interaction, the entire dashboard, etc.).
- **MQ7.** At which development stage is the variability achieved?
- **MQ8.** Which methods have been used for enabling variability?
- **MQ9.** How many studies have tested their proposed solutions in real-world environments?

The review scope was defined following the PICOC method [96] along with the posed research questions.

- **Population (P):** Software solutions.
- **Intervention (I):** Provide support to tailor (information) dashboards.
- **Comparison (C):** No comparison intervention in this study, as the primary goal of the present SLR is to analyze existing approaches regarding tailoring capabilities and gain knowledge about them.
- **Outcomes (O):** Information dashboard proposals.
- **Context (C):** Environments related to data visualization and (or) decision making (in academia, industry, etc.).

Inclusion and exclusion criteria

A series of inclusion (IC) and exclusion criteria (EC) to select articles that could answer the RQs and dismiss those unrelated to the review scope.

- **IC1.** The paper describes a dashboard solution (proposal, architecture, software design, model, tool, etc.) AND
- **IC2.** The solution is applied to information dashboards AND
- **IC3.** The solution supports or addresses tailoring capabilities (customization, personalization, adaptation, variation) regarding information dashboards AND
- **IC4.** The tailoring capabilities of the dashboard are related to its design, components, or KPIs AND
- **IC5.** The papers are written in English or Spanish AND
- **IC6.** The articles are published in peer-reviewed Journals, Books, or Conferences AND

- **IC7.** The publication is the most recent or complete of the set of related publications regarding the same study.

The exclusion criteria are derived from the inclusion criteria as their opposite.

- **EC1.** The paper does not describe a dashboard solution (proposal, architecture, software design, model, tool, etc.) OR
- **EC2.** The solution is not applied to information dashboards OR
- **EC3.** The solution does not support or address tailoring capabilities (customization, personalization, adaptation, variation) regarding information dashboards OR
- **EC4.** The tailoring capabilities of the dashboard are not related to its design, components, or KPIs OR
- **EC5.** The papers are not written in English or Spanish OR
- **EC6.** The articles are not published in peer-reviewed Journals, Books, or Conferences OR
- **EC7.** The publication is not the most recent or complete of the set of related publications regarding the same study.

Search strategy

Four electronic databases (Scopus, Web of Science (WoS), IEEE Xplore, and SpringerLink) were selected to perform the search. The selection process used the following criteria:

- It is a reference database in the research scope.
- It is a relevant database in the research context of this literature review.
- It allows using similar search strings to the rest of the selected databases and using Boolean operators to enhance the outcomes of the retrieval process.

The search concepts employed to build the search query are detailed in [48]. The search was conducted on January 22, 2019, for the **first version** and on May 18, 2022, for the **updated version**.

Query strings

Scopus

```
TITLE-ABS-KEY ((meta-dashboard*) OR ((dashboard*) W/10 (custom* OR personal* OR adapt*
OR flexib* OR config* OR tailor* OR context-aware OR generat* OR compos* OR select* OR
template* OR driven)) OR ((dashboard*) AND ( (heterogeneous OR different OR diverse OR
dynamic) W/0 (requirement* OR stakeholder* OR user* OR need* OR task* OR necess*))  ))
AND NOT TITLE-ABS-KEY (car OR vehicle OR automo*) AND NOT DOCTYPE(cr)
```

Web of Science

```
TS=((meta-dashboard*) OR ((dashboard*) NEAR/10 (custom* OR personal* OR adapt* OR
flexib* OR config* OR tailor* OR context-aware OR generat* OR compos* OR select* OR
template* OR driven)) OR ((dashboard*) AND ((heterogeneous OR different OR diverse OR
dynamic) NEAR/0 (requirement* OR stakeholder* OR user* OR need* OR task* OR necess*))))
NOT TS= ( car OR vehicle OR automo* )
```

IEEE Xplore

```
(((meta-dashboard) OR ((dashboard) NEAR/10 (custom* OR personal* OR adapt* OR flexib*
OR tailor OR tailored OR configurable OR context-aware OR generation OR generated OR
generative OR composed OR composition OR selection OR selecting OR template OR driven))
OR ((dashboard) AND ((heterogeneous OR different OR diverse OR dynamic) NEAR/0
(requirement OR stakeholder OR user OR need OR task OR necessities)))) AND NOT (car OR
vehicle OR automo*))
```

SpringerLink

```
((meta-dashboard*) OR ((dashboard*) NEAR/10 (custom* OR personal* OR adapt* OR flexib*
OR config* OR tailor* OR context-aware OR generat* OR compos* OR select* OR template*
OR driven)) OR ((dashboard*) AND ((heterogeneous OR different OR diverse OR dynamic)
NEAR/0 (requirement* OR stakeholder* OR user* OR need* OR task* OR necess*))))
```

Quality criteria

Another set of criteria was also defined to assess each work's quality to answer the RQs, before including them in the final literature review. Each criterion can be scored with three values: 1 (the paper meets the criterion), 0.5 (the paper partially meets the criterion), and 0 (the paper does not meet the criterion).

1. The research goals of the work are focused on addressing the variability, adaptability, customization, or personalization of an information dashboard to improve individual user experience (UX).
    - *Partial: not every research goal tries to address UX through tailoring capabilities.*
2. A software solution that supports the variability of the dashboard components is presented.
    - *Partial: the software supports customization of the dashboard but is not the focus*
3. A model, framework, architecture, or any software engineering artifact that addresses the dashboard components' variation and interaction methods is adequately exposed.
    - *Partial: a model, framework, architecture, or any software engineering artifact is exposed but not detailed, i.e., the nature of the referred elements is mentioned, but their internal structures and details are not further explained.*
4. The employed methods or paradigms to achieve tailoring capabilities are appropriately described.
    - *Partial: the employed methods or paradigms to achieve tailoring capabilities are partially described, i.e., the methodology is mentioned, not detailed in the application context.*
5. The context or domain of application of the dashboard is described.
    - *Partial: the context or domain of application is mentioned but not detailed.*
6. The proposed solution has been tested with real users.
    - *Partial: real users have used it and tested its functionality, but no further testing has been performed.*
7. Issues or limitations regarding the proposed solution are identified.
    - *Partial: problems or limitations are mentioned but not detailed.*

Each paper can obtain a maximum of 7 points regarding its quality following this methodology. This 0-to-7 score was transformed into a 0-to-10 scale, and the seven value was chosen as the threshold for including a paper into the final synthesis. If on a 0-to-10 scale, a paper obtains a score of fewer than seven points, it will be dismissed from the review as it did not meet a minimum quality to answer the stated research questions.

2.1.2 Data extraction process

Once the search was performed –on January 22, 2019, for the **first version** and on May 18, 2022, for the **updated version**–, the paper selection process was carried out through the following procedure:

1. The raw results (i.e., the records obtained from each selected database) were gathered in a GIT repository[3] and arranged into a spreadsheet[4]. A total of **2185** papers were retrieved: **595** (**254** + **341**) from Web of Science, **1035** (**501** + **534**) from Scopus, **192** (**97** + **95**) from IEEE Xplore, and **363** (**182** + **181**) from SpringerLink.
2. After organizing the records, duplicate works were removed. Specifically, 755 records were removed, retaining 1430 works (65.45% of the raw records) for the next phase.
3. The maintained papers were analyzed by reading their titles, abstracts, and keywords and applying the inclusion and exclusion criteria. 1327 papers were discarded as they didn't meet the requirements, retaining 103 articles (7.20% of the unique papers retrieved) for the next phase.
4. The selected 103 papers were read in detail and further analyzed. The papers were scored regarding their quality to answer the research questions using the quality assessment checklist described in the previous section. One paper

[3] https://github.com/AndVazquez/slr-tailored-dashboards/tree/master/update-2022
[4] https://bit.ly/3KvwygV

was added after checking the references of the assessed works, leaving 104 records for this quality assessment phase.

5. After applying the quality criteria, a total of 30 papers (2.10% of the unique papers retrieved and 29.12% of the full text assessed articles) were selected for the present review. Although 36 papers were above the 7-score threshold, six records were finally discarded during the last phase. This exclusion was because the six works were previous or partial versions of other studies found within the retrieved records. The decision was to keep the more complete and/or more recent work.

The update process added 9 more works to the original SLR's included articles (23). However, two of these 9 works were more complete or recent versions of the originally 23 articles included, so the two old versions [49, 97] were replaced with the newer ones, resulting in the final 30 included papers.

The PRISMA flow diagram has been employed to detail the data extraction process. Specifically, the PRISMA 2009 [98] flow diagram was used for the first version of the SLR (**Figure 4**), and the detailed paper selection can be consulted in **Appendix H**. Information dashboards and tailoring capabilities – A systematic literature review [48]. In the case of the updated version, the PRISMA 2020 [99, 100] guidelines were followed. **Figure 5** shows the PRISMA 2020 flow diagram for the updated version of the SLR and the paper selection procedure described at the beginning of this subsection.

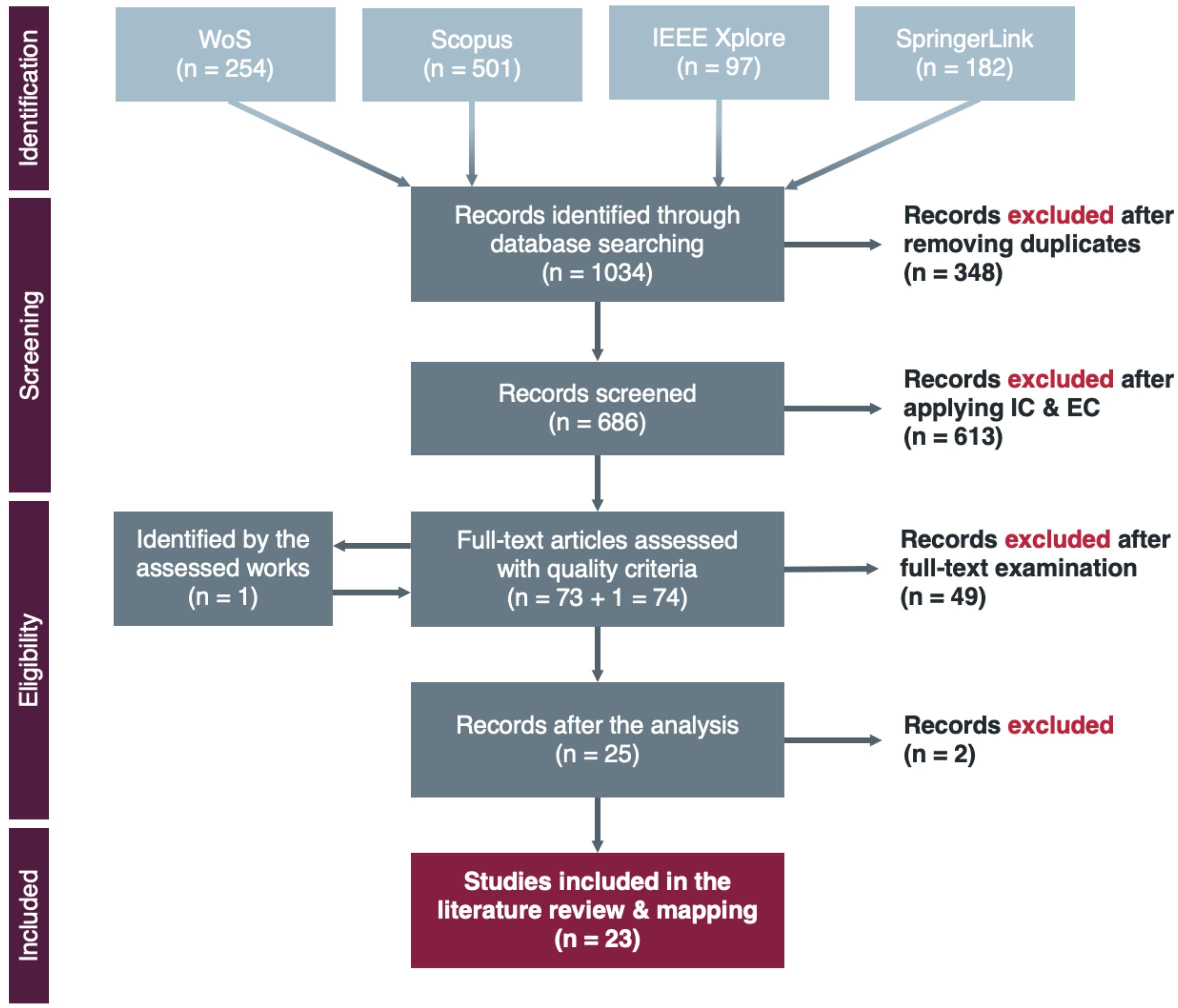


Figure 4. Phases and results of the first review process carried out on January 22, 2019, using the PRISMA 2009 flow diagram. Source: own elaboration.

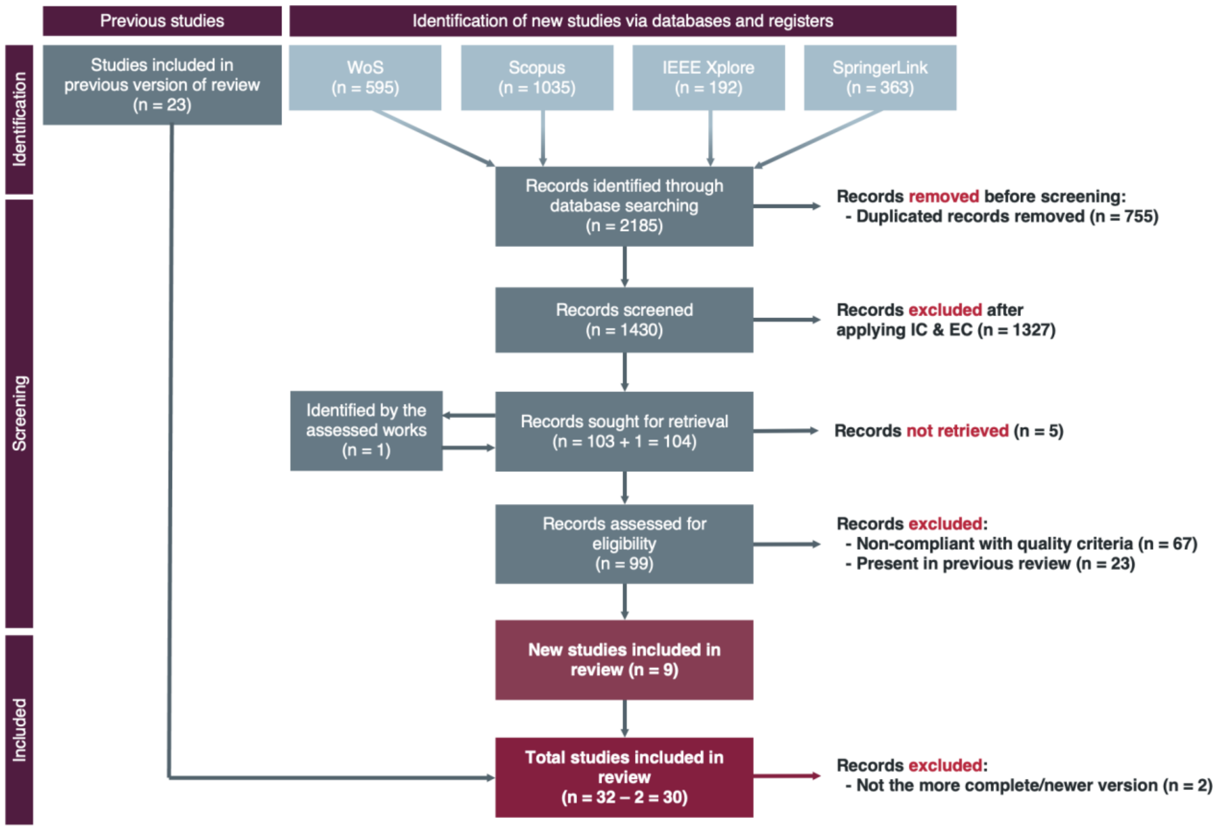


Figure 5. Phases and results of the updated review process carried out on May 18, 2022, using the PRISMA 2020 flow diagram. Source: own elaboration.

2.2 Results of the systematic literature mapping

This section presents the updated mapping results of the collected records. A Jupyter notebook (http://jupyter.org) based on the work developed by Cruz-Benito http://bit.ly/2tS9JgF was employed to support the analysis process of the raw data.

MQ1. How many studies have been published over the years?

The results cover from 2011 to 2018, with a work placed in 2007 [101]. A few records were published in 2011 [102, 103], 2012 [104], 2013 [105], 2014 [106-108] and 2016 [109, 110]. However, most records are distributed between 2017 [111-116] and 2018 [117-

121], with six and seven papers, respectively. The update of the SLR also shows that this field has the period of the first SLR (2018) and the current version (2022), with two works from 2019 [70, 122], three from 2020 [123-125], and four from 2021 [68, 126-128].

The number of selected papers per year can be consulted in **Figure 6**.

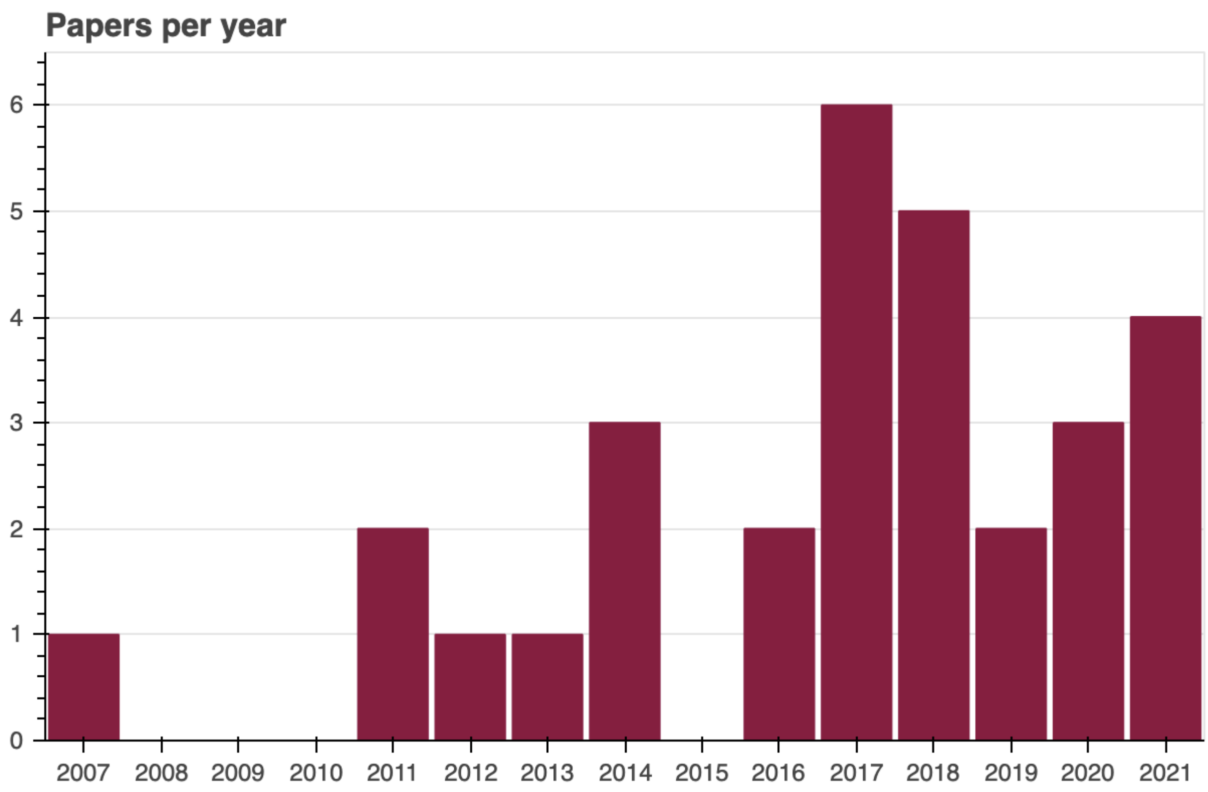


Figure 6. Distribution of papers per year. Source: own elaboration.

MQ2. Who are the most active authors in the area?

Five authors have more than one record within the retrieved results. On the one hand, Kintz presents a model-driven solution for generating dashboards [104, 113]; in one case, it shows the semantic description language. The other one gives an extension to consider user roles in the dashboards' generation process.

On the other hand, Van Hoecke is one of the authors of two publications regarding dynamic monitoring dashboards through semantic technologies [110, 123].

Finally, Vázquez-Ingelmo, García-Peñalvo, and Therón (author and supervisors of this thesis, respectively) describe the application of the SPL paradigm for generating employability dashboards [70] and also for classifying potentially misleading visualizations [68].

The rest of the authors appear only once in this mapping study. **Table 1** shows all authors and their number of papers in the scope of this literature mapping. Some authors also had more than one paper related to tailored dashboards. However, they were omitted because of the exclusion criteria EC7, so the most recent and complete paper about their study made it to the final phase.

Table 1. Authors' addressing variability on dashboards. Source: own elaboration.

Author	Total
García-Peñalvo, F.J.; Kintz, M.; Therón, R.; Van Hoecke, S; Vázquez-Ingelmo, A.	2
Amer-Yahia, S.; Arjun,, S.; B. Mayer; Barros, R.; Bastidas, V.; Bederson, B.B.; Belo, O.; Bezerianos, A.; Borges, M. R. S.; Bose, J.; Bouarour, N.; Cabrera, C.; Cardoso, A.; Castelnovo, C.; Chan, A.L.; Chowdhary, P.; Chua Zhen Liang, D.; Chua, G.G.; Ciucanu, R.; Collet, P.; Correia, H.; Da Col, S.; Dabbebi, I.; Danaisawat, K.; Dantas, V.; De Paepe, D.; Elias, M.; Elmqvist, N.; Filonik D., Medland R., Foth M., Rittenbruch M.; Furtado, V.; García-Holgado, A.; Garlatti, S.; George, S.; Gilliot, J.M.; Guo, S.; Hautte, S.V.; Hruška, T.; Huys, C.; Hynek, J.; Iksal, S.; Janssens, O.; Ji, M.; Karstens E.; Khunkornsiri, T.; Kochanowski, M.; Koetter, F.; Kukolj, S.; Kumar, K.; Lavoue, E.; Logre, I.; Magnoni, L.; Majstorović, B.; Mak, M.T.; Mariani, L.; May, M.; McGuinness, D.L.; Michel, C.; Mihaila, G.; Min Chim Lim, P.; Miotto, G.L.; Mobilio, M.; Moens, P.; Mosser, S.; Nascimento, B. S.; Noonpakdee, W.; Ongenae, F.; Orlovskyi, D., Kopp, A.; Palpanas, T.; Pastushenko, O.; Petasis G.; Phothichai, A.; Pinel, F.; Pinheiro, P.; R. Weinreich; Radovanović, S.; Riganelli, O.; Riveill, M.; Rodrigues, P.; Rojas, E.; Santos, H.; Siong Ng, W.; Sloper, J.E.; Soare, M.; Soni, S. K.; Sousa Pinto, J.; Steenwinckel, B.; Triantafillou A.; Tundo, A.; Van Herwegen, J.; Verborgh, R.; Verstichel, S.; Vieira Teixeira, C.J.; Vivacqua, A. S.; Yalcin, M.A.; de Walle, R.V.	1

MQ3. What type of papers has been published?

Each consulted electronic database provides the metadata to answer this mapping question. According to the inclusion and exclusion criteria, only peer-reviewed papers (either in journals, conferences, books, or workshops) are included. The complete list of types regarding the analyzed records can be consulted in **Table 2**.

Table 2. Papers grouped by type of publication. Source: own elaboration.

Type	Total	Papers
Conference paper	21	[102] [103] [104] [105] [106] [107] [108] [109] [110] [111] [112] [113] [115] [116] [117] [120] [121] [122] [125] [126] [127]
Article	9	[101] [114] [118] [119] [70] [123] [124] [128] [68]

MQ4. To which contexts have been the variability processes applied?

Dashboards can be used in any domain; the only requirement is to have enough data to visualize. Regarding customizable and/or personalized dashboards, Business Intelligence (BI) is the most common application domain, followed by Internet of Things (IoT), services monitoring, Learning Analytics (LA), and generic solutions (**Table 3**).

Table 3. Papers grouped by target domain. Source: own elaboration.

Domain	Total	Papers
Business Intelligence	9	[101, 103, 104, 106, 113, 115, 117, 122, 126]
IoT	4	[108, 110, 123, 124]
Services monitoring	3	[112, 118, 125]
Learning Analytics	2	[114, 116]
Generic	2	[119, 127]

Domain	Total	Papers
Communication	1	[68]
Disaster situations	1	[120]
Economics	1	[121]
Emergency management	1	[109]
Energy monitoring	1	[105]
Interface evaluation	1	[128]
Microservices monitoring	1	[111]
Physics	1	[103]
Sensor monitoring	1	[107]
Social sciences	1	[70]

MQ5. Which are the factors that condition the dashboards' variability process?

One of the first steps to perform a variability process is determining the factors that will condition the dashboards' variation, i.e., the customization and/or personalization stage inputs. Most of the included papers use user preferences to modify the dashboard appearance and functionality (**Table 4**).

Table 4. Papers grouped by variability factors. Source: own elaboration.

Factor	Total	Papers
User preferences	21	[68, 70, 102, 103, 105, 107, 109-112, 114, 115, 117-119, 122-125, 127, 128]
Data structure	6	[115, 116, 119, 121, 126, 127]
Business process	3	[101, 104, 113]
User role	2	[101, 113]
Design guidelines	2	[117, 128]

Factor	Total	Papers
Usage profiles	1	[106]
Data sources	2	[108, 110]
Goals	2	[104, 113]
User description	1	[116]
Analysis scenario	1	[116]
User abilities	1	[120]

MQ6. What is the target of the variability process?

Variability processes have a target that will change or be modified after the variation has been accomplished. In the case of dashboards, several elements could be the target of the variation: visualization types, layout, displayed data, visual design (i.e., color palettes, font sizes, etc.), and even interaction (pan, zoom, etc.) or functionalities (filters, exportation, etc.). **Table 5** lists the different variability targets identified in the included papers.

Table 5. Papers grouped by the target of the variability. Source: own elaboration.

Target	Total	Papers
Visualization types	28	[68, 70, 101-112, 114-119, 121-124, 126-128]
Layout	24	[68, 70, 101-112, 114-119, 124-128]
Displayed data	27	[68, 70, 101-119, 121-125, 127]
Visual design	2	[118, 120]
Interaction	2	[104, 120]
Functionalities	1	[70]

MQ7. At which development stage is the variability achieved?

The modification of dashboard features can be performed at different stages. In this case, four steps were identified: compile-time, run-time, pre-configuration time (i.e., a phase before the creation of the dashboard in which the end-user or any other stakeholder defines its configuration), and user-configuration time (i.e., at run-time, but the user is responsible for the configuration of its dashboard).

Pre-configuration and user-configuration seem preferred to customize or personalize the dashboards (**Table 6**).

Table 6. Papers grouped by variability stage. Source: own elaboration.

Stage	Total	Papers
Pre-configuration	13	[68, 70, 101, 104, 107, 113, 115, 117, 118, 122, 124, 125, 128]
User-configuration	9	[102, 103, 105, 109, 111, 112, 114, 121, 123]
Run-time	8	[106, 108, 110, 116, 119, 120, 126, 127]
Compile-time	1	[108]

MQ8. Which methods have been used for enabling variability?

A set of methods have been identified through the included papers. The most repeated method consists of configuration wizards to allow users to tailor their dashboards. Some solutions give extra support to these wizards with visual mapping to ease the selection of proper visualizations given the data structure to be visualized [103, 109, 119, 121]. Other common methods are configuration files, agents, SPL, and model-driven development. The detailed list of methods can be consulted in **Table 7**.

Table 7. Papers grouped by variability methods. Source: own elaboration.

Method	Total	Papers
Configuration wizard	10	[102, 103, 105, 109, 111, 114, 119, 121-123]
Visual mapping	5	[103, 109, 119, 121, 126]
Model-driven	5	[101, 104, 113, 124, 125]
Configuration files	4	[112, 118, 122, 128]
SPL	3	[68, 70, 107]
Agents	2	[106, 108]
Pre-defined templates	2	[101, 117]
Semantic reasoner	2	[110, 123]
Inclusive user modeling	1	[120]
Context-aware generator	1	[116]
Indicator ontology	1	[115]
Knowledge graphs	1	[115]
Machine Learning	1	[127]

MQ9. How many studies have tested their proposed solutions in real environments?

The last mapping question is regarding the performed tests on the included dashboard solutions. Most (16) of the solutions have been tested in real-world scenarios involving real data and real users, while 6 have not been tested with real users or real data (**Table 8**). Four solutions have been partially tested in a real-world scenario, i.e., they have been tested with real data but not with real users or vice versa.

Table 8. Papers grouped in terms of testing. Source: own elaboration.

Tested?	Total	Papers
Yes	13	[102, 103, 105, 107-110, 113, 116, 117, 119-121]
Partially	11	[68, 70, 101, 114, 122-128]
No	6	[104, 106, 111, 112, 115, 118]

2.3 Results of the SLR

RQ1. How have existing dashboard solutions tackled the necessity of tailoring capabilities?

The selected works were categorized in terms of their tailoring process. Each paper was analyzed to answer the questions that would characterize them into a specific category. The final classification can be seen in **Figure 7**.

Most of the selected works are framed in the category of "customizable," meaning that the tailoring process of the dashboard is driven by explicit user requirements [68, 70, 103, 105, 107, 111, 112, 114, 117, 118, 122, 124, 125, 128].

Customizable solutions involve manual approaches mainly (which will be detailed in RQ2), meaning that users need to perform a set of explicit actions to tailor their dashboard according to their needs [102, 105, 111, 114, 122].

However, not only manual user interactions are employed for arranging the tool, some of these customizable dashboards involve generative or automatized approaches through the specification of configuration files [112, 118, 128], models [70, 107], or pre-defined templates [117].

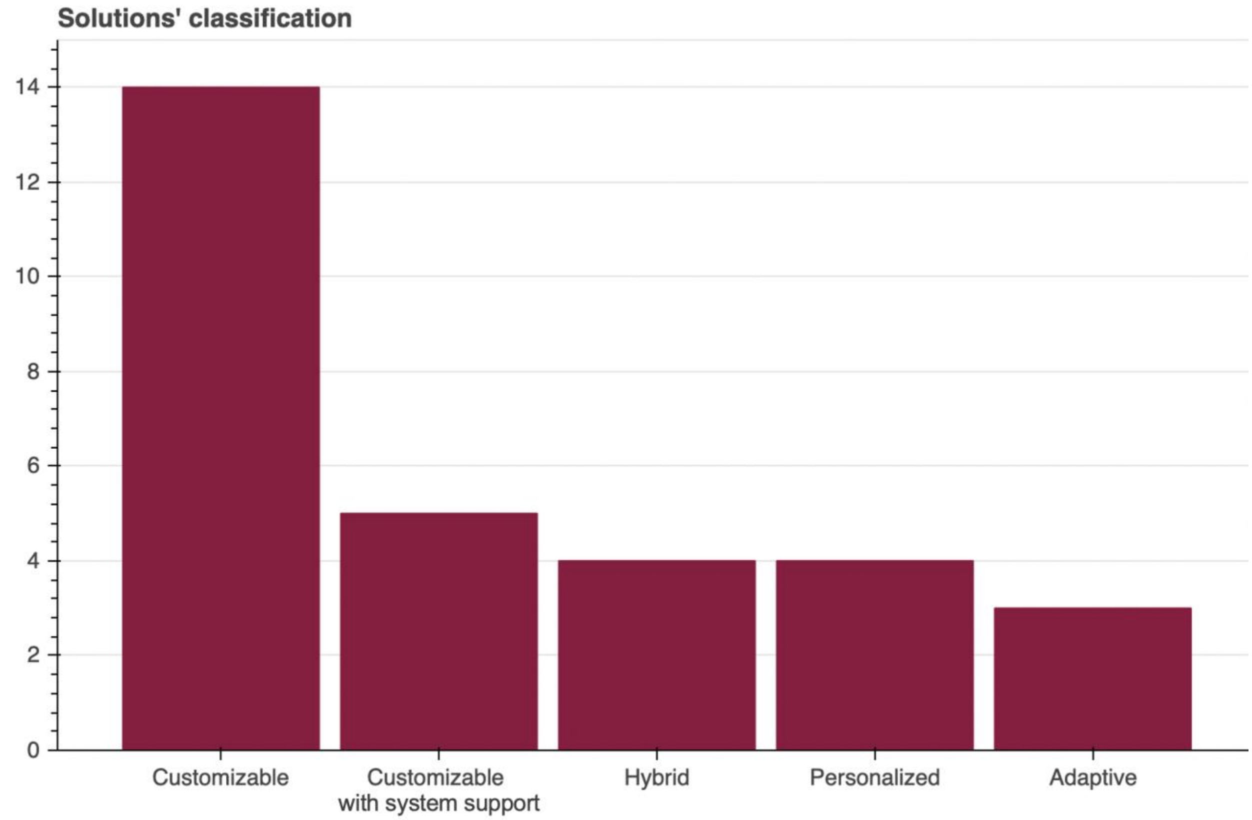


Figure 7. Classification of the retrieved solutions in terms of their tailoring method. Source: [48], own elaboration.

Although these solutions involve automatization, the inputs (configuration files, models, templates, etc.) of the generative pipelines still need to be filled with the user requirements. In the end, these generative approaches add an abstraction layer that helps users to configure their dashboards without requiring programming skills. For these reasons, these solutions are also classified as customizable dashboards.

In contrast with **customizable** solutions, **personalized** solutions infer a suitable configuration based on implicit data about users, tasks, or goals [101, 104, 113, 120].

Adaptive solutions, on the other side, can adapt themselves at run-time based on environmental changes. These environmental changes include the analysis of user queries [106], interaction history [116], and explicit user feedback [127]. The last solution takes advantage of machine learning (ML) to adapt the dashboard's views depending on the user's interactions.

Other tailored solutions have been identified, as they cannot be framed on the last categories (**customizable**, **personalized**, or **adaptive**). On the one hand, solutions identified as "**hybrid**" are mainly personalized or adaptive dashboards that allow the user to have the last word regarding the dashboard configuration, or that need user actions to complete the tailoring process. Works in this category analyze the data sources to personalize or adapt the dashboards' visualization types [108, 126] or indicators [110, 115], but allow users to customize or create the final display.

On the other hand, there are customizable solutions that can assist and help users build their dashboards according to a series of factors, identified as **customizable with system support** solutions. Visual mapping is the preferred method to assist users in the selection of the best visualization types for their dashboards [103, 109, 119, 121]. Also, [123] presents an extension of [110] in which the semantic reasoner supports and guides the implementation of the dashboard through a graphical interface.

Classifying these tools regarding their tailoring capabilities is complex. The selected papers describe too many different solutions implemented through various methods with other goals, so this classification of tailored dashboards should be seen as a spectrum, allowing the existence of dashboards that mix features of different approaches. However, framing them in distinct categories allows better understanding regarding existing solutions and the current state of the present field. **Table 9** summarizes this categorization.

Table 9. Classification of the solutions regarding their tailoring capabilities. Source: own elaboration.

Type	Total	Papers
Customizable	14	[105, 107, 111, 112, 117, 118] [68, 70, 103, 114, 122, 124, 125, 128]
Customizable with system support	5	[102, 109, 119, 121, 123]
Personalized	4	[101, 104, 113, 120]
Hybrid	4	[108, 110, 115, 126]
Adaptive	3	[106, 116, 127]

RQ2. Which methods have been applied to support tailoring capabilities within the dashboards' domain?

The preferred method for customizing dashboards is by using **configuration wizards** that support the users' decisions when building their customized dashboards without programming skills. For example, [102, 105, 111, 114, 122] uses graphical user interfaces that ease the selection of widgets and the data to be displayed.

Configuration wizards are also the preferred method for customizable dashboards with system assistance, in conjunction with visual mapping methods that ease the selection of visualization types given the data types or structure [103, 109, 119, 121, 123]. Although it is considered a hybrid solution, authors in [126] also make use of mapping (by using thresholds) to match data sources to specific visualizations. Users configure their dashboards based on their needs and the system then provides feedback to support the customization process and potentially obtain more effective dashboards.

Another common method to customize dashboards is to configure them by using structured **configuration files** [112, 118, 122], which also allow users to tailor their dashboards with a higher level of abstraction (through JSON files, XML files, etc.) through richer and more domain-specific syntaxes than programming languages. Although [122] uses a graphical interface to select widgets, the final dashboard specification and data schema are stored as JSON files. In this case, a series of parameters are set to render a concrete and functional dashboard.

Some works also take advantage of **the SPL** paradigm [68, 70, 107] or **Model-Driven Development (MDD)** [101, 104, 113, 124, 125].

For example, in [125], a meta-model is used to transform the definition of different KPIs into an arranged set of visualizations. In this case, the meta-model only references the layout of the dashboard. The final rendering process is performed through external tools, such as Kibana or Grafana, as the authors mention.

These paradigms are used to finally generate a dashboard that fits previously defined feature models (in the case of SPL) or meta-models (in the case of MDD).

A similar MDD approach is followed in [116], although authors do not explicitly indicate that they followed this paradigm. In this case, to generate the dashboard, a **context-aware generator** with users' data and visualization models as inputs oversees the generation of the dashboard instances. Still, the internal features of the dashboard generator are not detailed.

Regarding adaptive solutions, **agents** are a common method for managing changing requirements [106, 108]. **Machine learning** has also been used in [127] to adapt the dashboard and to learn from users' interactions to show better data visualizations subsequently.

Other methods found in the selected papers enclose **inclusive user modeling** for adapting the dashboard interface to the user abilities [120], **semantic reasoners** for selecting appropriate data sources and compositions [110, 123], and **knowledge graphs and ontologies** to adapt the dashboards to the target data domain [115].

RQ3. How do the proposed solutions manage the dashboard's requirements?

The second research question shows that configuration wizards are popular methods to manage these requirements by giving the user the responsibility of building their own dashboard based on their necessities.

These solutions allow users to customize their displays while using their dashboards freely, thus performing the tailoring process at user-configuration time (i.e., at run-time, but with the user's intervention through explicit actions). All solutions found that use a configuration wizard approach [102, 103, 105, 109, 111, 114, 119, 121, 122] manage individual user requirements by implementing **authentication and account management services**, associating each user to their dashboard configuration persistently. This user management approach is also applied to other solutions found, like in [120], where a user creates an account and fills a questionnaire about their abilities to access their personalized view based on the previous information finally. Also, in [106], users' behavior and events need to be stored to adapt the display.

However, these works do not further discuss the storage method nor the possibility of storing different versions of a user dashboard over time, which could be very useful for collecting the evolution of the preferences or user behavior. This fact also applies to [127]. Although interactivity with the system needs to be captured to trigger the recommendation process, it is only mentioned that the feedback from the user is stored, but no further details are given.

On the other hand, various selected works take advantage of **structured files** or **models** to hold individual dashboard requirements that finally serve as inputs of generators that provide the configured dashboard instance meeting the original specifications. In this category fall those solutions based on configuration files [112, 118, 128], data models [108], context models [116], software product lines [68, 70, 107], or model-driven development [101, 104, 113, 124, 125]. In this case, user requirements are managed "outside" the dashboard systems before their exploitation and stored within individual files or models.

In the case of [117], no requirement management is explicitly performed, as the pre-defined templates enclose general requirements collected from the gathering and analysis phase. The same happens in [126], which only references the characteristics of the datasets to personalize the dashboards, but the final dashboard implementation is undisclosed.

The remaining solutions use semantic reasoners [110, 123] and knowledge graphs [115] to manage the dashboards' information requirements, but the management of the end-users' requirements is not further discussed.

RQ4. Can the proposed solutions be transferred to different domains?

Most of the solutions can be transferred to different domains. This is the case of solutions in which data sources can be uploaded or specified [103, 106, 119, 122, 126, 127], solutions based on MDD [101, 104, 113, 124, 125] or SPL strategies [68, 70, 107], and some solutions based on configuration files [112, 128].

In the case of [126], although the application domain is business intelligence, the applied thresholds refer to abstract features of the datasets, such as their size, so this solution could be employed in other contexts. This is also the case of [127], in which the ML recommendation process is triggered by user interactions and the suggested visualizations are adapted to the target dataset, no matter the domain.

Some solutions allow freedom when configuring the dashboards, but only within the original domain (environmental performance [105], micro-services monitoring [111, 118], emergency situations [109], learning analytics [114, 116], physics [102], economics [121].

The works that focus on sensor monitoring [110, 123] and device clouds [108] employ methodologies that could be reused for other domains. Still, in the end, the dashboard solutions would need to be built from zero to adapt them to new domains.

The remaining solutions are tightly coupled to its original context, which is the case of [120] –the adaptation is focused on users' physical abilities–, [117] –the templates are related to specific areas of BI–, and [115] –a specific Smart City ontology is employed to tailor the dashboards–.

As a clarification, it is worth stating that every methodology employed in the selected papers could be applied to develop dashboards in different data domains. However, the purpose of this research question is to identify the most flexible and powerful solutions regarding their abstraction and, therefore, their potential reuse to other domains in an automatized manner (i.e., avoiding developing the same solution for new domains manually).

RQ5. Has any artificial intelligence approach been applied to the dashboards' tailoring processes, and, if applicable, how have these approaches been involved in the dashboards' tailoring processes?

Only a few works have applied or mentioned AI when presenting their dashboard solutions. The most explicit application of AI can be found in [127], in which authors use a Multi-Armed Bandits (MBAs) reinforcement learning model to improve the recommendations and adaptation of the dashboards' visual components based on the users' feedback.

In [106], the Apriori algorithm [129] is used to compute association rules, a technique from the data mining field. This solution takes advantage of "pairs of events that have happened in sequence" that fed the Apriori algorithm to obtain a set of if-then rules that will be used to restructure the dashboard in terms of the presented data and visualization types employed. In a study referencing those mentioned above [130], the same authors specify that their solution also supports the restructuration of the dashboards through other methods, like Markov chains or top-k queries, but they do not detail these processes.

Also, in [110, 123], a semantic reasoner is employed to discover potentially interesting data compositions through a knowledge base and semantically annotated visualization and data services. However, no details about the implementation of the reasoner are addressed in this work.

Finally, although it is not explicitly involved in the dashboard generation process, the work exposed in [68] focuses on labeling and training ML models to obtain a classifier and detect potentially misleading visualizations.

Other papers mention the possibility of introducing AI techniques, like [128], to rate the generated dashboards through classification algorithms. Still, the authors state that is out of the scope of the paper and refer to [131] as an inspiration. There is also a work that mentions inference methods [116] to provide a suitable dashboard given the context, user description, and analysis scenario, although no further details are given, nor the inference method named.

RQ6. How mature are tailored dashboards regarding their evaluation?

Only 11 articles mention any kind of user testing. The testing methodologies for each of these works are summarized in **Table 10**. There are some works, like [104, 108, 109, 113] that mention user testing outcomes, however, they do not describe the methods employed nor the samples' sizes. Also, it should be underlined that they collected user feedback but do not detail the specific method.

Table 10. Summary of user-testing methods applied in the retrieved articles. Source: own elaboration.

Article	Interview	Survey	Data exploration	Other	# Participants
[105]	Yes	-	-	Expert evaluation	5 experts 13 participants
[111]	Yes	Yes	-	-	15 participants
[108, 109]	-	-	-	Undisclosed	Undisclosed
[117]	-	-	-	Multi-criteria evaluation [132]	40 enterprises
[104, 113]	-	-	-	User feedback	2 participants
[103]	Yes	-	Yes	-	7 novice users 8 BI experts

[119]	Yes [133]	-	Think-aloud protocol	-	6 novice users
[114]	Yes [134]	-	Yes	-	12 participants
[121]	-	Yes	Yes	-	60+ participants

The solutions presented in [68, 70, 101, 102, 106, 107, 110, 112, 115, 116, 118, 120, 122-128] did not mention any formal testing regarding end-users' perceptions about the dashboard solutions, mentioning these evaluations as future work. Some of these proposed tools were tested in real-world scenarios to prove their applicability and functionalities, but this research question is focused on user perceptions on the solutions.

2.4 Conclusions

The SLR aimed at identifying current trends and solutions within the domain of study of the present Ph.D. thesis: the automatic generation of tailored information dashboards.

The research questions covered relevant aspects to consider when addressing generative workflows of dashboards. With the collected information, it is possible to select the best strategy to implement approaches that tackle the automatic generation of these tools.

By virtue of the SLR outcomes, the decision is to follow a meta-modeling approach to conceptualize the generative framework and the SPL paradigm to materialize and transform abstract features into source code. The analysis of the retrieved articles has proved that these two approaches are feasible in this domain; almost one third of the selected works -8 out of 30- employ one of these two paradigms.

However, the feasibility of the solutions was not the only object of study. Other attributes, like flexibility and evolving capabilities, possibility to transfer the solutions to any data domain, traceability of the dashboard requirements, and the potential to integrate AI algorithms to adapt the dashboard features to environmental changes were also under the focus of this review.

On the other hand, other challenges and open research paths have been identified during the analysis of the selected works. For example, a few works mention leveraging Machine Learning (ML) models to unburden users from complex tasks such as configuring the dashboard layout. However, these applications are not detailed or are in their first development stages.

This research line is very promising, as these methodologies could yield several benefits to assist users and provide them with useful guidelines to learn and understand how to design effective dashboards and visualizations. In this sense, choosing a meta-modeling approach is also suitable for applying ML methods, as these models require structured data to learn from. Instances of the meta-model can be provided as inputs to identify patterns that make specific configurations useful, efficient, effective, usable, etc.

In short, carrying out this analysis have provided clear evidence that solutions following meta-modeling and/or SPL paradigms meet these properties, concluding that the versatility of these solutions provides a great starting point for implementing a generative dashboard system.

3 Dashboard meta-model

This chapter addresses the main outcome of the present research: the dashboard meta-model. According to the results of the SLR, using a meta-modeling approach has proved to be suitable in the information dashboards' domain for a series of relevant factors.

First, it enables the abstraction of commonalities. Although information dashboards can present disparities in their design and look very different at first sight [92], they are developed using common, low-level elements [135].

Second, the meta-model provides structures for these low-level features to arrange them into a set of entities and relationships, capturing how the different elements that comprise information dashboards influence each other.

Third, this approach supports the automatic generation of products through methodologies such as Model-Driven Architecture (MDA) and Model-Driven Development (MDD), or Software Product Lines (SPLs). This approach implies that the obtained meta-model can be translated into source code to develop real-world, functional products.

Finally, adopting a model-driven generative approach entails traceability of every design decision taken until obtaining the final product, from theoretical specifications (model instances) to tangible system features (dashboards' code). Traceability is crucial in the context of information dashboards and visualizations. As will be discussed in subsequent sections, this attribute improves the transparency of the whole design process, which could result in a better understanding of data visualizations. In addition, it enables easier version control of each model instance, keeping the evolution of individual dashboard requirements.

In addition to the above, meta-modeling has many more benefits: increase in flexibility, faster developments, reusability of core assets, reusability of knowledge, etc. [136]. The suitability of a model-driven approach in this context led to the proposal of a dashboard meta-model to tackle the automatic generation of software interfaces for supporting decision-making processes.

The rest of this chapter details the developed dashboard meta-model and the followed domain engineering methodology to obtain the final version (Section 3.1), the approach conceived to implement a generative pipeline (Section 3.2), and the conclusions derived from the application of this proposal to the information dashboards' domain (Section 3.3).

3.1 Dashboard meta-model

The dashboard meta-model consists of a series of elements, properties, and relationships among them. As stated in Chapter 2, the dashboard meta-model is framed within the MDA paradigm [137].

The dashboard meta-model is part of the four-layer meta-model architecture proposed by the OMG, in which a model at one layer is used to specify models in the layer below [18]. **Figure 8** shows the correspondence of the dashboards domain with the followed MDA paradigm [89].

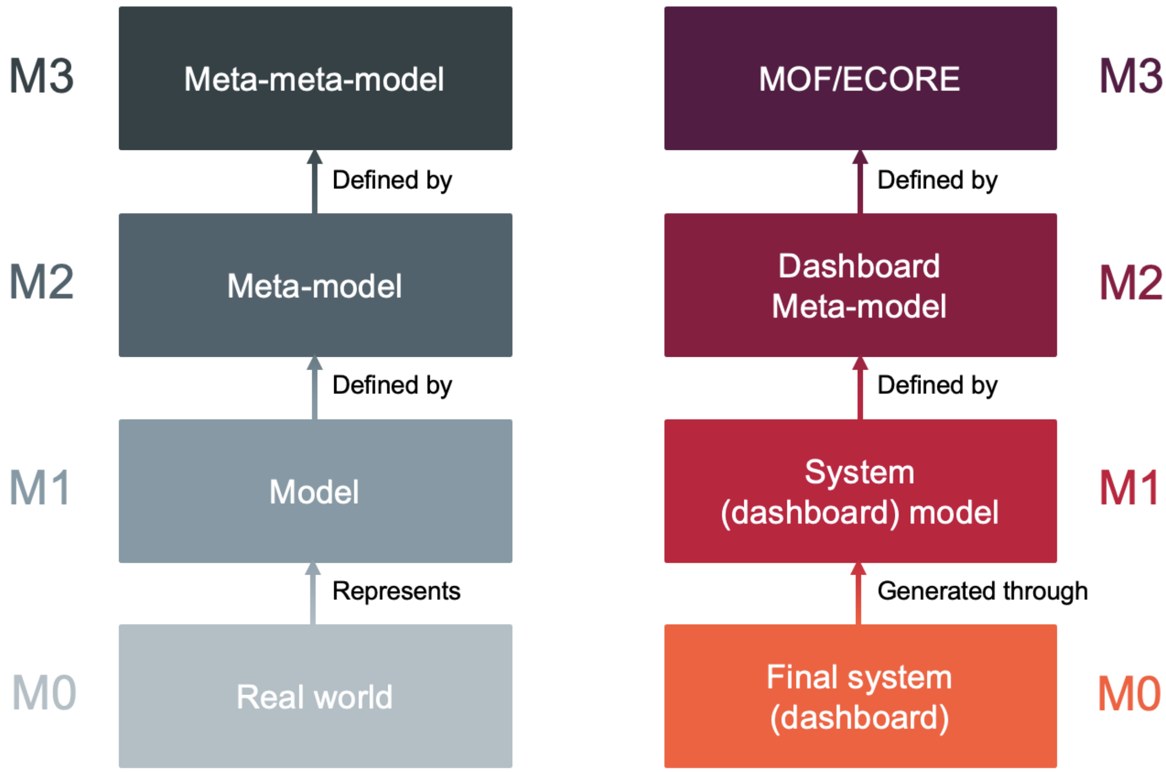


Figure 8. Correspondence of the MDA framework levels with the followed approach in the dashboards and data visualizations domain. Source: own elaboration.

The dashboard meta-model is an M2 model, which is defined by a meta-meta-model, and it will be used to instantiate dashboard model which, in turn, will be transformed into real-world, functional dashboards.

The first version of the dashboard meta-model was an instance of MOF; however, it was finally transformed into an instance of Ecore using Graphical Modelling for Ecore included in Eclipse Modeling Framework (EMF).

The development of the meta-model has been subject to several iterations with the aim of improving the captured domain elements. **Figure 9** shows an overview of the improvements made in each iteration. The first two iterations were focused on identifying the static, tangible elements of data visualizations and dashboards (layout, visual components, resources, etc.). The next two iterations, on the other hand, deal with the user characteristics and their intents with the dashboard. The fourth iteration aims at modeling interactive, dynamic behavior between the dashboard elements. Finally, the last improvement included more complex and higher-level concepts like the domain of the data to be displayed. An animated overview of this evolution can be consulted at https://youtu.be/ZyAZIRZXogc.

As it can be seen, the meta-model development process was incremental, which helped in focusing on the entities of the dashboard domain one by one to identify better their properties and relationships.

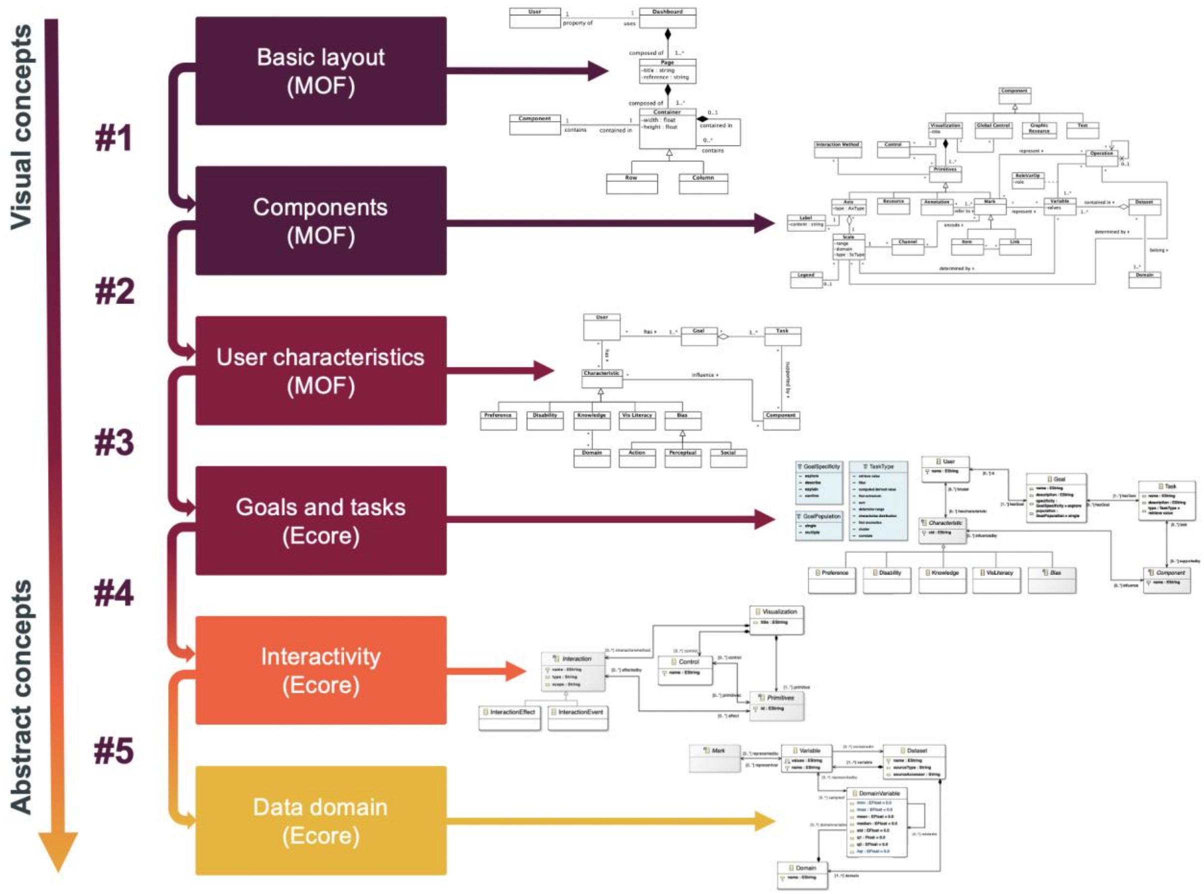


Figure 9. Iterations in the development of the dashboard meta-model. Source: own elaboration.

The following subsections detail the meta-model development process towards obtaining the current version of the meta-model. All these iterations have been published in [49-54, 70].

3.1.1 Basic layout

The first iteration of the meta-model development process was focused on identifying the basic skeleton that makes up information dashboards and visualizations. This version of the meta-model was employed to develop dashboards within the employment and employability domain. This case study is detailed in Section 4.4.1, and the associated publications at **Appendix E**. Domain engineering for generating dashboards to analyze employment and employability in the academic context and **Appendix L**. Taking advantage of the software product line paradigm to generate customized user interfaces for decision-making processes: a case study on university employability [49, 70]. The very first version of the dashboard meta-model included the layout specification and the user entity (**Figure 10**).

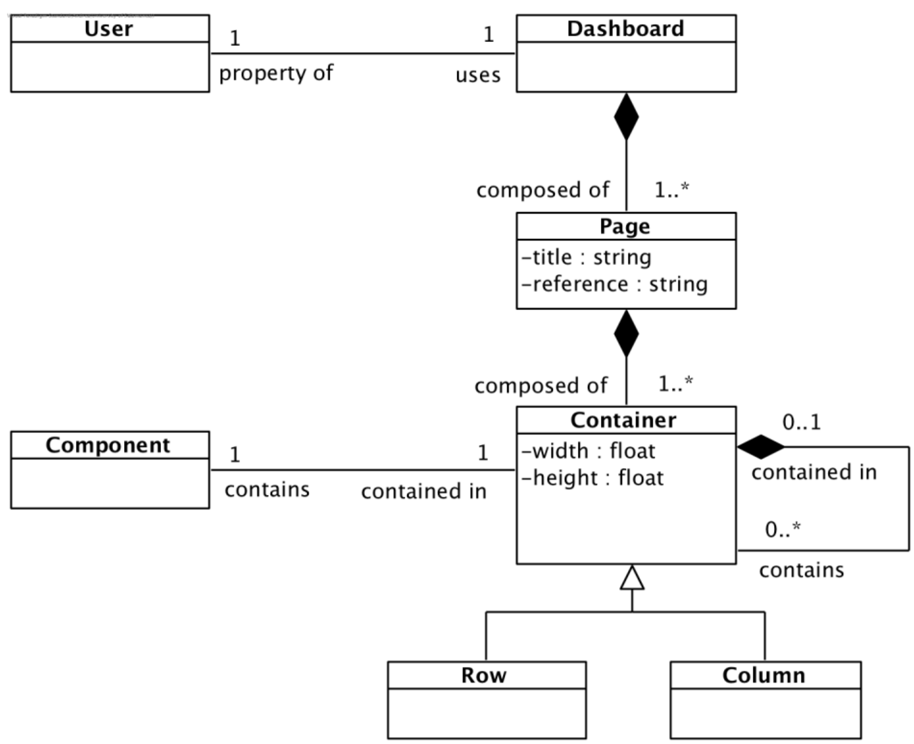


Figure 10. Dashboard meta-model (Initial version). Source: own elaboration, published in [70].

The meta-model was defined as an instance of MOF and captures the highest-level entities involved in the domain. In this case, the <<User>> uses a <<Dashboard>>, which is composed of one or more pages that contain one or more containers (divisions of the screen with a width and height). Containers can be specified as rows or columns, and they can contain, in turn, more containers recursively. Finally, each container holds a <<Component>>, which references any kind of resource in the dashboard: data visualizations, text, images, etc.

3.1.2 Including the components' specification

The second iteration included the detailed specification of dashboards' components, specifically data visualizations [50]. This part is very complex because several primitive elements and combinations are involved when building these tools. To capture these elements, different types of visualizations were analyzed to identify common and abstract features among them.

As can be seen in **Figure 11**, this meta-model excerpt extends the initial version's <<Component>> entity. This version is still an instance of MOF.

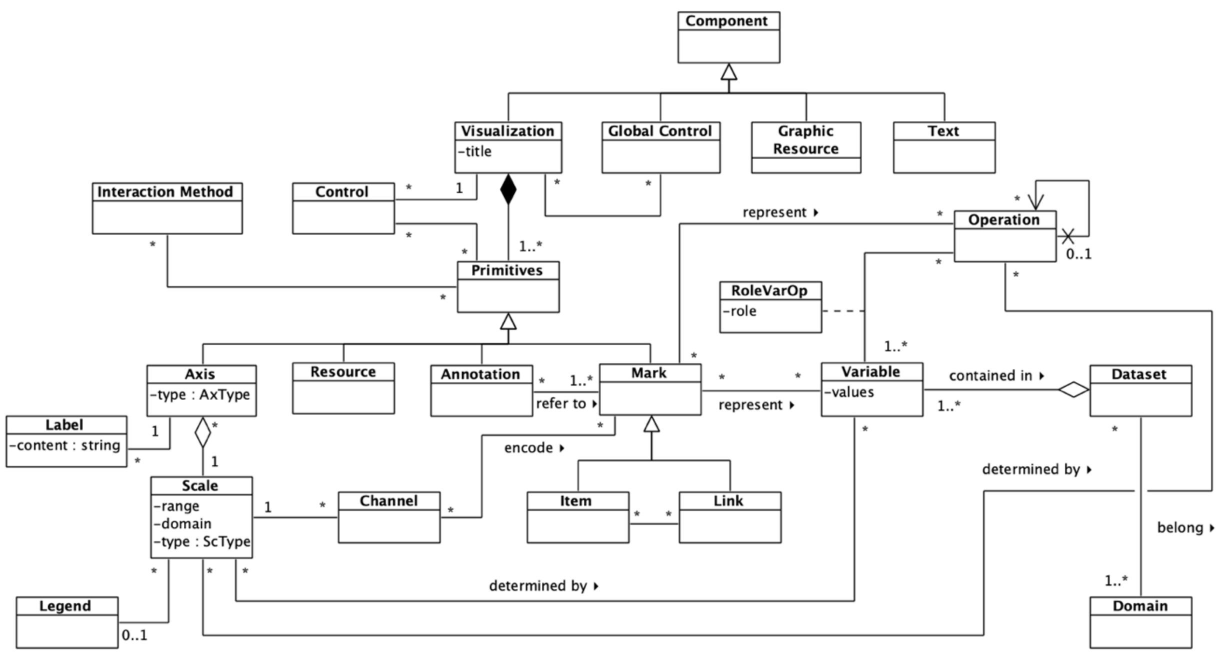


Figure 11. Dashboards' components specification (increment #1). Source: own elaboration, published in [50].

Specifically, the features of the meta-model have been identified through domain engineering [81-84, 86], a review of visualization grammars [135, 138], and the outcomes of the previous literature review [48].

These methodologies and resources were complemented with an example-driven approach [139], providing an approach to identify common fine-grained components of information dashboards and data visualizations.

First, a component in a dashboard display does not have to be necessarily a visualization. Some of the dashboard's containers can hold graphical resources (e.g., images or illustrations) or text, to provide a context to the displayed information or instructions about how to employ the tool. But in the end, the main components of dashboards are the information visualizations that present the domain's data.

On the other hand, components can also be controls or tools that can affect several visualizations at once. For example, filters that allow to select or highlight data points that meet certain conditions among every visualization in the dashboard are identified as “global control” in the meta-model.

A visualization can be affected by the aforementioned global controls, and by "local" controls (i.e., controls that only affect a specific visualization). This distinction allows having control of visualizations both on global and local levels, thus letting users explore data more freely. In this case, a control is understood as any explicit handler that allows modifications on visualizations at any dimension: displayed data, design, visual encoding, etc.

Moreover, a visualization can be decomposed into lower-level elements that are shared among all the potential instances. That is why the meta-model reflects that a visualization is composed of one or more primitives. The <<Primitive>> class is a high-level class that encompasses different elements.

These elements can be axes, annotations, marks, and resources (images, text, etc.). But before detailing the meaning of these low-level components, it is important to clarify that the visualizations' local controls can affect these primitives; as introduced, a control allows the modification of the visualizations, that is, their primitives, which are who hold the actual information.

In addition, primitives of a given visualization can also be modified by the available interaction methods. For example, a visualization that allows zooming will change the primitives when this interaction method is employed. Once these classes and associations have been clarified, each primitive will be detailed.

First, one of the most important primitives regarding visualizations are axes. Axes contain information about the scales and thus, about some properties that can influence the appearance of a visual mark, as it will be explained. Axes can take different forms, which are encoded as a meta-class attribute (*type*); for example, an axis can be linear or radial, presenting curvature in its presentation.

Axes can be labeled to clarify their role, or the variable being represented. A meta-class <<*Label*>> is included in the diagram to reflect this capability.

On the other hand, an axis is always associated with a unique scale. A unique axis cannot represent more than one scale at once; however, a scale can be represented in several axes, providing redundant information, for instance.

Scales have a domain, a range, and a type, the last referring to the nature of the data. Given the data properties, associated scales can be linear, ordinal, nominal, logarithmic, etc.; the selection of a proper scale is essential in the information visualization field, so the mentioned attributes are necessary for the meta-model. In addition, these attributes are common to any scale, so they are worth to be included in an abstract representation of an information visualization. Scales can be associated with a legend to improve the understandability of the visualization.

Relevant visualization elements have been explained so far, but their backbone is the visual encodings of data, that is, the marks that contain actual information about different data variables.

There are popular terms to refer to the representation of data elements in data visualizations, but the most used among the literature are marks and visual channels or visual encodings [140-143]. Marks are shapes with properties that represent different values through their properties. In this sense, marks can be items or links. Items represent nodes, points, series, zones, etc., and links represent connections, containments, etc., among items [143].

Marks represent data variables contained in a dataset or the outcomes of operations (arithmetic operations, aggregations, etc.), based on the PTAH meta-model presented in [107]. To visually represent the values, these can be encoded through different visual properties: position, size, length, color, opacity, angle, curvature, etc. These visual properties are referred to visual channels or visual encodings [140-143]. The same channel can be employed to encode different marks. Channels are associated with a scale, which will map variables' or operations' values to specific channel values.

Annotations are also considered in the meta-model, as they can be crucial elements in declarative visualizations, where the main focus is on explaining values rather than on exploring them [144]. Annotations can refer to different marks, and a mark can be affected by zero or more annotations.

Regarding the previously mentioned operations, it is also important to bear in mind the role of the different variables that might take part in an operation. For example, an aggregation would have groups and a target. For this reason, an association class (*RoleVarOp*) that models the role of a variable within an operation has been included. Moreover, a recursive association in the *Operation* class has been modeled to support chained operations between variables.

Finally, datasets can be associated with different domains, a concept that will be thoroughly described in Section 3.1.5.

Figure 12 shows an example of how the previous entities have been identified in real world data visualizations through the followed domain engineering [81-84, 86] and example-driven [139] approach.

Figure 12. Identification of commonalities in data visualizations following a domain engineering approach. Source: own elaboration.

The work related to this section is published and can be consulted in **Appendix J**. Capturing high-level requirements of information dashboards components through meta-modeling [50].

3.1.3 Including user characteristics

Including the audience of the dashboard is crucial to delivering a truly effective tool. As literature points out, there is not a "one size fits all" when we talk about dashboards and data visualizations. Some visual designs are more understandable for lay users and others are more efficient for users with higher visual literacy [145].

For these reasons, modeling users is essential as they influence the elements that will be displayed in the dashboard. This extension is focused on the <<User>> entity. In the initial version, the <<User>> class represents a high-level user, but none of his or her characteristics are represented nor detailed. However, the user should be defined in terms of different significant and influential aspects to support a personalized dashboard design, thus being necessary to extend this meta-model with more elements regarding the users' characteristics and goals, as well as defining the relations of these aspects with the dashboard's components.

Given that, the extended dashboard meta-model is presented in **Figure 13**, also as an instance of MOF. Firstly, a new concept arises, <<Goal>>. A user employing a dashboard must have at least one goal, however implicit. Even users that want to explore data casually have a goal (that is, exploring a dataset). That is the reason for the "one or more" (1..*) multiplicity. In turn, a goal can belong to any user, and users can share common and general purposes, explaining the "zero or more" (*) multiplicity on that side of the relation.

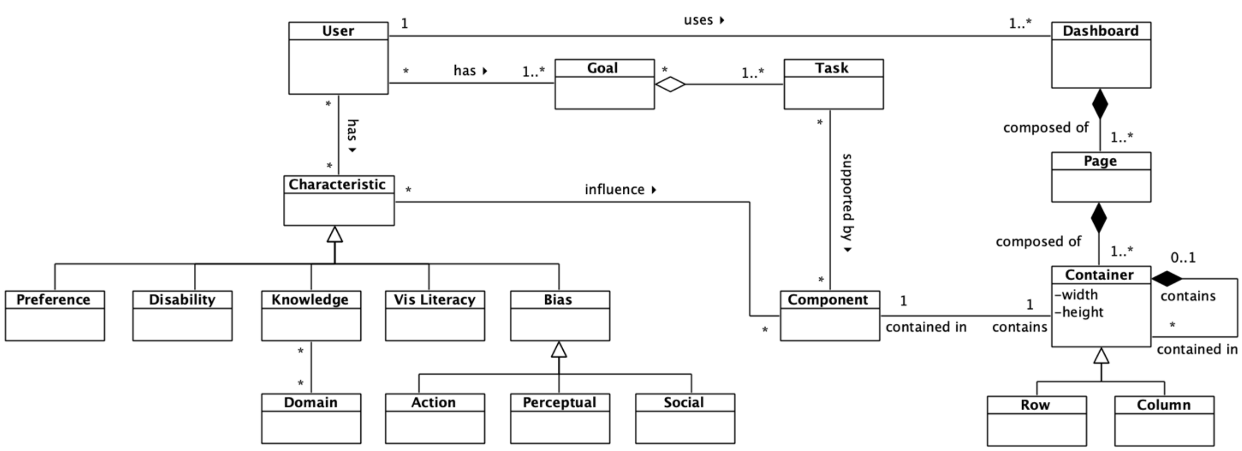


Figure 13. Users' specification (increment #2). Source: own elaboration, published in [51].

On the other hand, a goal can be broken down into individual and more specific tasks [146]. Simple goals can be accomplished by performing one task, e.g., if a particular goal is "to know which USA city has the largest number of inhabitants," a straightforward yet necessary task could be "to sort USA cities by population number," meaning that the dashboard components must support sorting capabilities.

However, more complex goals might involve several specific and chained tasks such as "to understand why there has been a business income loss within the last six months," which could involve applying different tasks to different dimensions of the data to reach insights about the stated problem. That is the reason why the dashboard's components need to support the identified responsibilities to enable them.

Finally, a user can have zero or more identified characteristics, given the fact that, at a certain point, there could be no user data available of the possible dimensions. These characteristics can belong to zero or more users, as different users can share general characteristics. Characteristics can be of a different kind; preferences, disabilities, knowledge about different domains, visualization literacy, and bias (action, perceptual, or social bias). These characteristics can influence the dashboard's components to adapt them and, therefore, to match the identified user aspects.

The work related to this section is published and can be consulted in **Appendix K**. Extending a dashboard meta-model to account for users' characteristics and goals for enhancing personalization [51].

3.1.4 Detailing goals and tasks

Goals and tasks were introduced in the last iteration, but it is also important to characterize them into structured and well-defined elements that can be used to add rules and recommendations. Another important step taken in this iteration was the specification of the metamodel in Ecore.

Although a slight addition, having information regarding the typology of tasks and the structure of goals is essential to building a robust model that can be instantiated into specific products. In addition, it would also ease the process of defining data collection tools by determining the necessary data and structures that should be gathered.

First, to include the analysis goal framework [147] into the meta-model, four attributes have been added to the <<Goal>> entity. These attributes are a name to identify the goal, its specificity, and its population. The fourth attribute is a description to complement the previous information if needed. The specificity attribute is an enumeration of the four values described in [147]: explore, describe, explain, and confirm, while the population enumeration has two values: single or multiple. The included attributes characterize the user's goals and support their structuration by classifying the goal intent through its specificity and population.

Given the flexibility of the analysis goal framework and the possibility of connecting it with other existing lower-level task taxonomies, the <<Task>> class has been complemented with three attributes. Two of the attributes are also a name and a description to enrich the specification of the task. The last attribute is the task type, which can be one of the ten low-level analytical tasks depicted in [148]: retrieve value, filter, compute a derived value, find extremum, sort, determine a range, characterize distribution, find anomalies, cluster and correlate. The extension of the meta-model is shown in **Figure 14**.

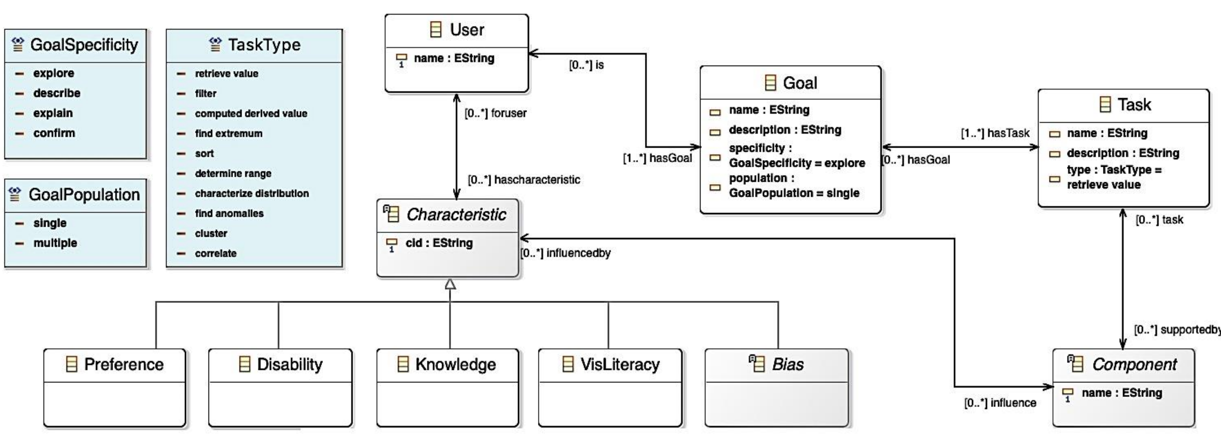


Figure 14. Goals and tasks characterization (increment #3). Source: own elaboration, published in [52].

The work related to this section is published and can be consulted in **Appendix O**. Representing Data Visualization Goals and Tasks Through Meta-Modeling to Tailor Information Dashboards [52].

3.1.5 Including interaction

Interactive behavior is also a crucial feature of data visualizations and information dashboards. These kinds of mechanisms allow users to drill-down into data and to identify patterns more clearly by highlighting elements across views, filtering data points, or getting a detailed view of the represented points without cluttering the display [149].

However, interaction patterns are highly diverse. They can involve the user clicking some parts of the dashboard. They can also involve hovering, brushing, etc. And, on the other hand, they can provoke different effects, such as highlighting some point, showing a tooltip, filtering the data, etc.

All these possibilities must be captured through the meta-model in an abstract and coherent manner. Following the domain engineering approach [81, 85], a set of conceptual classes have been identified across dashboards from different domains to model interaction patterns. These classes are depicted in **Figure 15**.

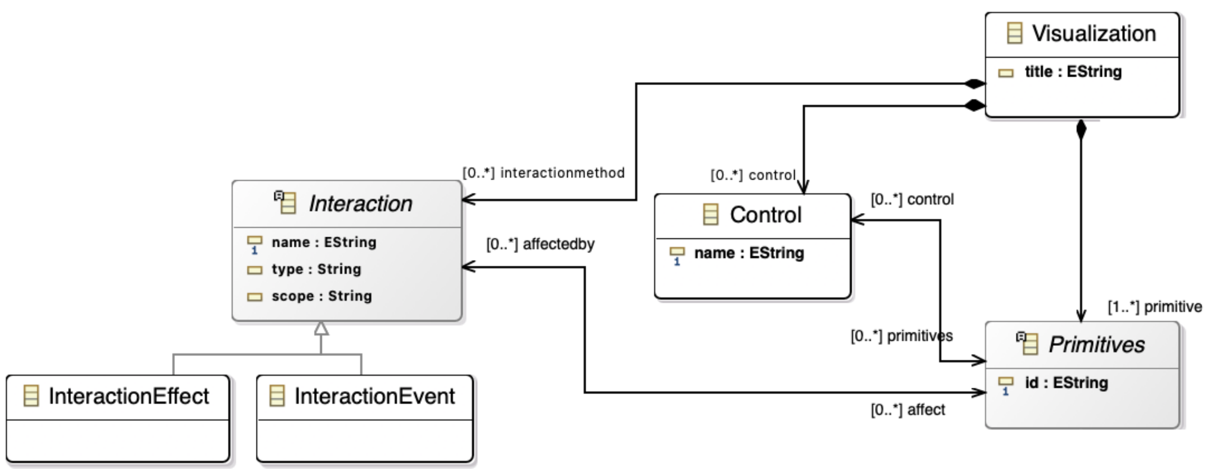


Figure 15. Interaction patterns specification (increment #4). Source: own elaboration, published in [53].

Information visualizations are composed of different elements. Mainly, these visualizations are composed of basic primitives, like visual marks, axes, scales, visual channels, etc. When interacting with a visualization, these primitives will be affected, for example, by changing their colors to highlight them or by showing a tooltip.

Three classes have been identified to reflect interactions in the meta-model. The *Interaction* class, which represents the interaction pattern to be applied to a specific primitive of the visualization. This class is abstract and can be of two types: an event or an effect. This distinction is necessary to represent which events to capture and which effects to apply to the visualization's primitives. For example, clicking in one of the bars from a bar chart is an event, and highlighting that bar (by varying its style) when selected is an effect. With these conceptual classes, it is possible to combine different specifications to obtain fully functional and interactive dashboards.

The work related to this section is published and can be consulted in **Appendix T**. Specifying information dashboards' interactive features through meta-model instantiation [53].

3.1.6 Including data domain characteristics

The last improvement of the dashboard meta-model made during the development of this thesis involved higher-level concepts of the dashboards' domain. The entities included in this iteration are related to the datasets' domain characteristics. The relevance of these entities was found during the execution of an experiment for classifying information visualizations through ML (the complete experiment and its outcomes are detailed in Section 4.3.2).

One of the outcomes of the experiment was the fact that it is necessary to identify relevant features to characterize the domain and to materialize those characteristics in useful visualizations. In this case, the domain is defined as a set of attributes that statistically describe the variables involved in that domain.

Specifically, a new class named *<<DomainVariable>>*, which represents data variables that are part of the data domain was included in the meta-model. This class is associated with a domain (in the meta-model, the class *<<Domain>>*) and also with the class *Variable*, which represents a variable that belongs to a dataset to be displayed through the visualization. The <<*Variable*>> entity is seen as a sample of the <<*DomainVariable*>> entity.

The domain variable enables users to perform analytic tasks on the visualization and reach insights, because the user has information about which values are normal, which values are outliers, or which tendency is being developed.

If users see a visualization with information from a domain which they do not fully understand, their conclusions might be wrong. But also, if practitioners do not fully understand the data domain of a visualization they are developing, they could end up with a misleading graphic. Another example about this concern is given. When visualizing information in X, Y coordinates it is necessary to select the domain of the scales in both axes; different scale extents might distort the whole data story being told. **Figure 16** shows an example of the same data visualized through different Y-axis scales.

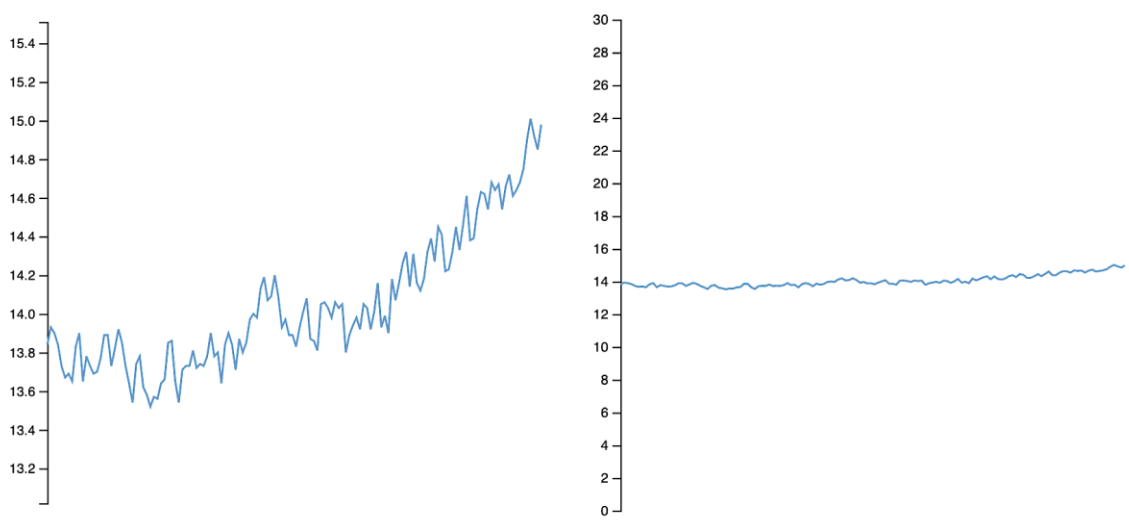


Figure 16. Example of the effect of different scales when visualizing data. Both figures represent the same set of data points. Source: own elaboration, based on the example from [150], published in [54].

If people are not aware of the data domain, the first graph can be seen as misleading for not starting the Y-axis at the zero value. Starting the Y-axis at the zero value (as in the second graph), gives the audience the impression that the temperature change over time is very small and, indeed it is (in absolute terms) [150].

However, in this case, being aware that the tendency and average temperatures of the world over the last decades provides the context to understand that a change of 1ºC in average temperature is a huge increment in this domain, conveying a whole different story. So, although the first graph does not comply to Tufte's lie factor [151], it is more honest than the second in terms of representing data framed in this domain.

For these reasons, the following attributes were introduced in the <<*DomainVariable*>> entity: mean, standard deviation, median, first quartile, third quartile, interquartile range, maximum and minimum. These values not only describe statistically the variable, but also they help in characterizing their distribution [152], as they give a notion of its dispersion, skewness and what values can be considered as outliers.

Another association regarding the <<*DomainVariable*>> entity was included with the goal of considering the possibility of representing context in a visualization. In this case, the context is identified as additional information related to a variable. For example, household income could be related to the COVID-19 incidence rate [153], and including that information in a visualization about COVID-19 incidence rate provides context to the data to be displayed.

By including a reflexive association on the <<*DomainVariable*>> class, the possibility to identify and materialize relationships among variables from a different or the same domain (for example, because they are correlated) is enabled.

The inclusion of this relationship to provide the notion of context allows the accountability of potentially relevant variables to include in a visualization before selecting its technical features. These new additions can be inspected in **Figure 17**.

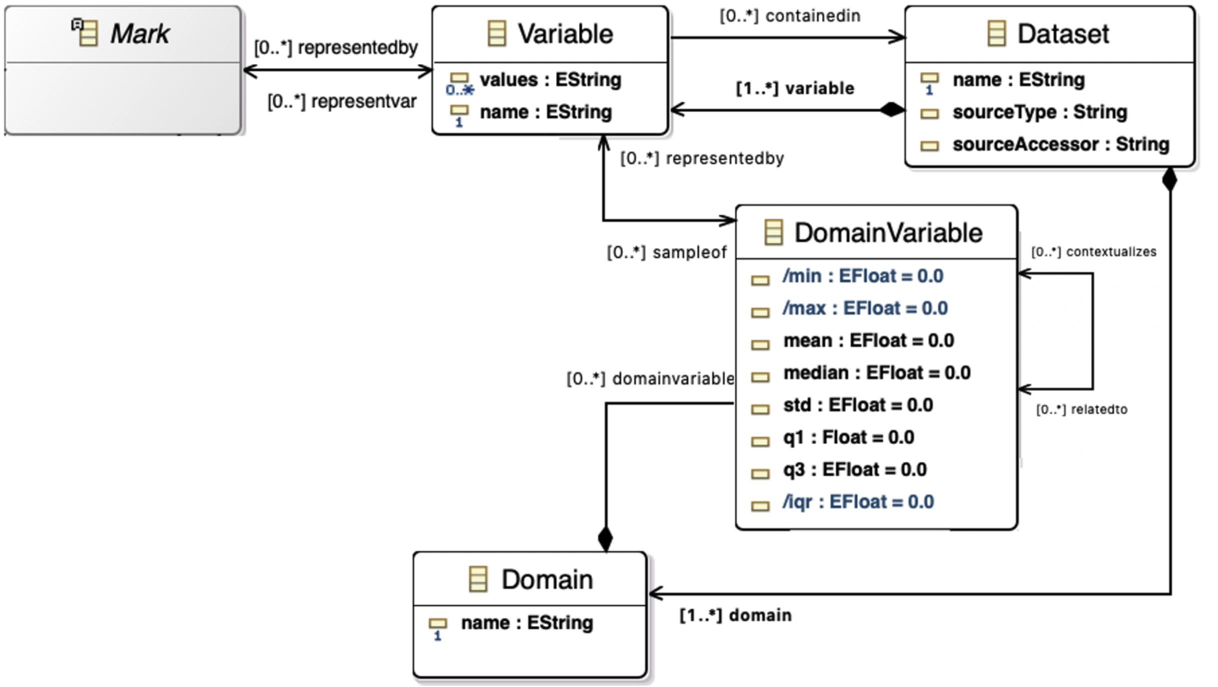


Figure 17. Specification of context and domain characteristics in the meta-model (increment #5). Source: own elaboration, published in [54].

The work related to this section is published and can be consulted in **Appendix Z**. A Meta-Modeling Approach To Take Into Account Data Domain Characteristics and Relationships In Information Visualizations [54].

3.1.7 Final version of the meta-model

After the five iterations, the final version of the dashboard meta-model includes every element detailed before, divided into three main sections: user, layout, and components. The current version of the dashboard meta-model is presented in **Figure 18**. A high-resolution version of the dashboard meta-model can be consulted at https://zenodo.org/record/5788981 [154].

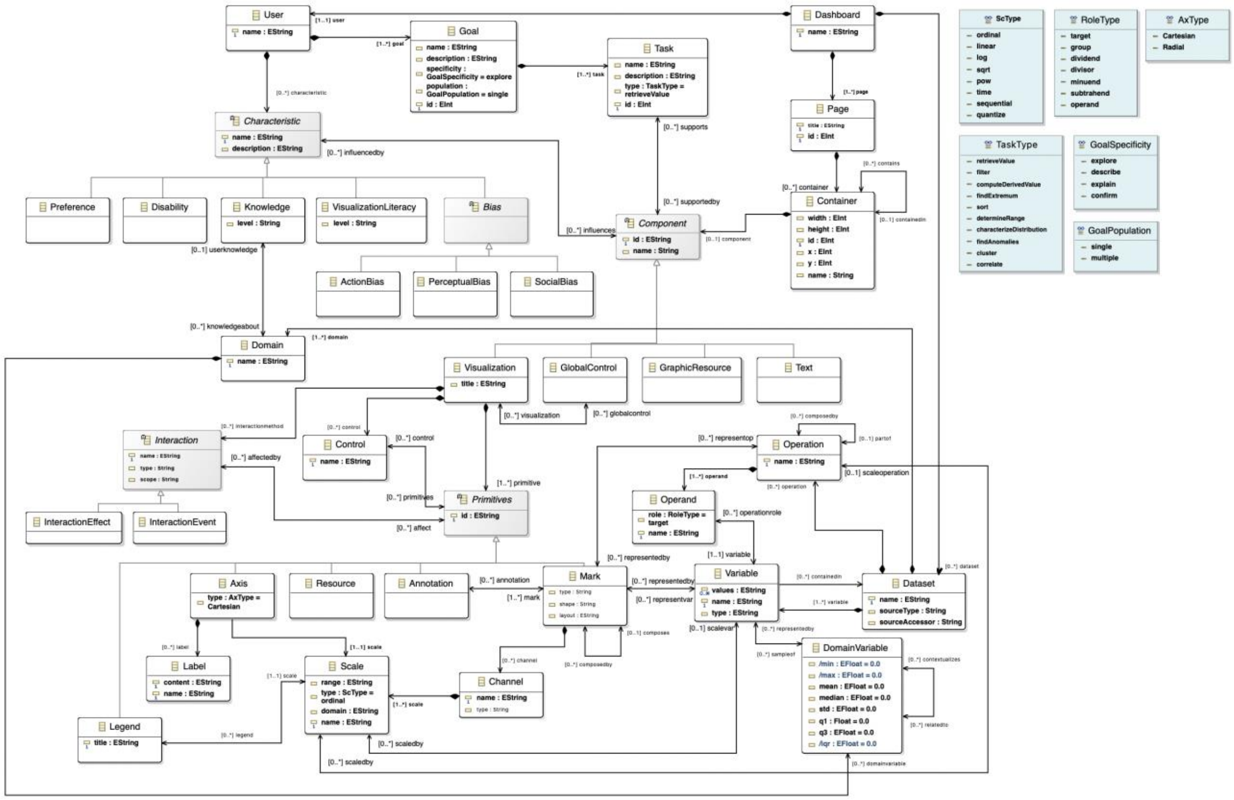


Figure 18. Final version of the dashboard meta-model [154]. Source: own elaboration, published in https://zenodo.org/record/7037624.

3.2 Generative pipeline

The generative pipeline of information dashboards was developed along with the meta-model. The implementation of the automatic generator of dashboards was also part of the domain engineering process to refine the meta-model, obtaining new dimensions to consider in the abstract level (meta-model) before instantiating them through the pipeline.

The generative pipeline follows an SPL approach that relies on the meta-model to identify the commonalities and variability points in the domain. The next subsections detail the methodology to implement these variability points and the final approach to generate functional information dashboards using the meta-model.

3.2.1 Methodology

There are several options to materialize the variability of an SPL. It is important to choose wisely given the requirements of the product line itself (i.e., the complexity of the software to develop, its number of features, their granularity requirements, etc.). Generally, at the code level, the variability points that correspond to a specific feature will be spread across different source files [155].

That is why separating concerns at the implementation level is essential to avoid the variability points to be scattered, as this feature dispersion would decrease code understandability and maintainability. Implementing each feature in individual code modules can help with this separation of concerns [155], but it is difficult to achieve fine-grained variability through this approach. A balance between code understandability and granularity should be devised to choose both a maintainable and highly customizable SPL.

An analysis of the suitability of different approaches for implementing variability points was performed in [56], and can be consulted in **Appendix F**. Addressing fine-grained variability in user-centered software product lines: a case study on dashboards After discussing the pros and cons of methods like conditional compilation [156-158], frames [155, 159, 160], aspect-oriented programming [161-164] or template engines [165], among others [155], the chosen implementation technique for the generative pipeline was to use a template engine.

The decision was made due to the fine granularity that can be achieved through this method, which is necessary to materialize even the slightest variability on the visualization components. Another factor for choosing this technique lies in the straightforward way of implementing variability regarding the products' features and its language-independent nature.

Framing technology could also be a potential solution within the dashboards' domain, but the decision of designing a DSL to wrap the features at a higher level made the use of code templates a more suitable solution, providing complete freedom to define the syntax of the DSL (specification x-frames are based on a fixed syntax

[160], which could result in lack of flexibility for this work's approach) as the directives within the templates can be fully parameterized.

The selected template engine was Jinja2 (*http://jinja.pocoo.org/docs/2.10/*) given its rich API and powerful features such as the possibility of defining macros, importing them, defining custom filters and tags in addition to its available basic directives (loops, conditions, etc.).

3.2.2 Approach

After developing the meta-model and selecting the methodology to implement the dashboard SPL, the final step was to put together these two paradigms to implement the generative pipeline.

While meta-modeling enables to identify the fine-grained features of information dashboards, the SPL paradigm provides the means to materialize the identified features into code, and thus, into real-world, functional tools.

Specifically, the combination of the meta-model entities and the template-based code rendering approach offers a great framework for this matter because both are focused on decomposing conceptual elements (in the case of meta-modeling) and code (in the case of code templates) into primitive elements.

Considering this, the approach to creating the generative pipeline is based on the idea of matching the meta-model entities and relationships with source code fragments. These fragments are, in turn, encapsulated into macros that can be conditionally included to render specific dashboard features.

Figure 19 shows an example of how the identified entities in the meta-model match specific code fragments framed within macros and conditions. These code fragments are glued together through a Python generator that renders the final code considering the input model (which is, in turn, an instance of the meta-model).

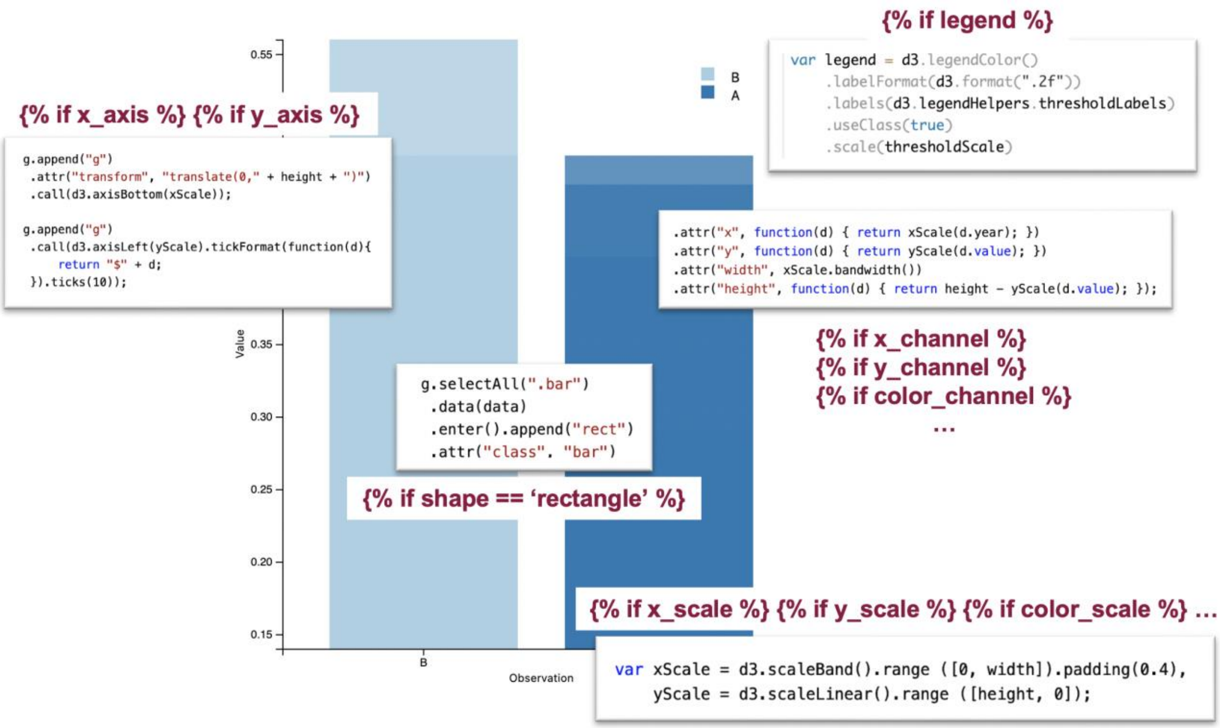


Figure 19. Using macros to label and materialize fine-grained features of information dashboards into code. Source: own elaboration.

In this case, the base languages employed to build the core assets of the SPL (that is, the language enclosed within the template engine directives) are HTML and JavaScript (visualizations are specifically built with the D3.js library [166]). **Figure 20** shows an example of how the rendering process of code templates results in JavaScript code. However, this templating approach is language-independent in terms of the content of the files to be rendered, so this approach can be reproduced with other programming languages if needed.

Figure 20. Excerpt of a code template rendering process. Source: own elaboration.

3.3 Conclusions

This chapter details a crucial output of this thesis: the dashboard meta-model. A meta-modeling approach was chosen as the best fit for automating the generation process of information dashboards, based on previous literature, and supported by the SLR findings. Meta-models along with domain engineering provide a powerful strategy to analyze and identify commonalities in complex domains, such as in the dashboards' domain.

The meta-model has been subject to different iterations before obtaining the final version. These increments not only enabled to focus on specific domain entities at a time but also to review and solve potential issues that might arise in later iterations. The first iterations were focused on the tangible elements of dashboards (layout, resources, visual components, etc.), followed by a couple of iterations to address the characterization of the dashboard audience (the user).

The last two increments tackled higher-level concepts, like interaction patterns (which are dynamic behaviors that are not explicitly tangible) and the data domain characteristics (which is implicit knowledge about the context in which real-world datasets are framed).

Developing the meta-model not only provided an artifact to build a generative framework for information dashboards but also increased the author's knowledge regarding this domain, which is another well-known benefit of using domain engineering: to capture, generate and reuse knowledge.

This approach also drove the decision of using code templates as the method to materialize the variability points of the SPL. Code templates were selected given their suitability in this context and resemblance to meta-modeling in the philosophy of encapsulating and decomposing complex entities into primitive elements. Matching the meta-model entities with their respective code fragments allowed the development of a dashboard generator, which is the milestone associated with the implementation phase in **Figure 2**. This milestone marks the beginning of the validation and application phases of the developed artifacts.

The next chapter will present the findings derived from the validation and application of both the meta-model and the automatic generator of dashboards into theoretical and practical scenarios as well as in real-world contexts. The diversity of scenarios in which the meta-model and the generator can be integrated to improve processes related to the development of data visualizations and information dashboards hints at the potential of bringing these methodologies together.

4 Results

This chapter provides different scenarios, applications, and outcomes derived from the meta-model. Developing the meta-model has not only resulted in the implementation of a code generator but also has made it possible to apply this artifact as a framework for conducting data visualization-related research.

The diverse setups designed to study the benefits of the meta-modeling approach from different perspectives provide hints at the versatility in terms of potential applications of the meta-model.

Different theoretical applications of the meta-model were carried out, which made possible its validation, its use as a roadmap to design information dashboards, and its integration with other meta-models.

On the other hand, practical applications aim to prove the viability of the meta-model-based generative pipeline in real-world scenarios, which also serve to improve the meta-model as new concepts and dimensions can be identified through these case studies.

Finally, the meta-model and generative pipeline were integrated in three real-world scenarios to test the capability of the meta-model and the SPL approach to adapt information dashboards to different and not-related contexts. The first scenario is in the domain of employment and employability, the second is related to healthcare and finally, the third scenario explores the use of the dashboard meta-model as a learning resource.

The following studies are driven by the hypothesis **H1** introduced in Chapter 1. The proposed scenarios aim to provide feedback regarding the benefits –in terms of functional and non-functional features– of using both the meta-model and the generative pipeline to tailor data visualizations and information dashboards.

4.1 Meta-model validation

One of the goals of developing the presented meta-model is to provide the foundations to understand the data visualization and information dashboards domain. However, as discussed in the previous chapter, this domain is complex, resulting, in turn, in a complex meta-model.

For these reasons, a validation study of the meta-model's content was conducted to test the coherence, relevance, and clarity of the different sections of this artifact. By validating the meta-model, it is possible to identify potential limitations and drawbacks of the dashboard domain representation and address them before using this artifact to instantiate real-world dashboards.

First, the meta-model quality framework proposed in [167] was applied to check the quality of the Ecore version of the dashboard meta-model before and after the modifications were introduced. The meta-model was compliant with the thirty features of the framework related to design, best practices, naming conventions, and metrics, which proves its quality.

Second, an expert validation of the dashboard meta-model's final version was carried out. This validation aims to check if the dashboard meta-model's sections are clear, coherent, and relevant. In particular, expert judgement [168] was applied: (1) selection of the experts; (2) provenance of guidelines to the experts by email; (3) collection of each answer –via online; and (4) analysis of the results.

An online questionnaire was created with six different sections of the meta-model (dashboard layout, user characteristics, goals and tasks, user and dashboard relationships, data visualization primitives, and data domain and operations), in addition to the whole meta-model. Each section was scored in terms of the mentioned dimensions using a 1-4 scale, where one implies that the section does not meet the criterion, and four that it highly meets the criterion. The rubric employed to score the different dimensions based on previous works on content validation by experts [169].

The questionnaire was configured in a customized installation of LimeSurvey and sent by e-mail to different domain experts and was available from July 10, 2021, to May 31, 2022. Eleven experts completed the questionnaire:

- Four women and seven men.
- All eleven experts are from Spain and Norway (6 and 5, respectively) and academicians in model-driven development, modeling, and information visualization.

A brief explanation of the meta-model section and a "Yes/No" question to test if the representation meets the intended goal of the section was also added. Finally, the questionnaire also included an open text field to collect any qualitative feedback of justification that experts might have. The detailed specification of the study procedure can be consulted in **Appendix AG**. Content-validation questionnaire of a meta-model to ease the learning of data visualization concepts [57].

Table 11 summarizes the results obtained from each expert regarding each meta-model section. The results yielded high and medium scores for every section and dimension and only one "low relevance" score in the dashboard layout section.

Table 11. Results of the expert validation. Attributes key: CL=Clarity, CO=Coherence, RL=Relevance. Scores key: H=High, M=Medium, L=Low. Source: own elaboration.

	E1	E2	E3	E4	E5	E6	E7	E8	E9	E10	E11
Dashboard layout	CL: M CO: H RL: H	CL: H CO: H RL: L	CL: H CO: H RL: H	CL: H CO: H RL: H	CL: H CO: H RL: H	CL: M CO: M RL: H	CL: H CO: H RL: H	CL: H CO: H RL: H	CL: M CO: H RL: H	CL: M CO: L RL: L	CL: H CO: H RL: M
User characteristics	CL: M CO: M RL: H	CL: H CO: H RL: M	CL: H CO: H RL: M	CL: H CO: H RL: H	CL: H CO: H RL: H	CL: H CO: H RL: H	CL: H CO: H RL: H	CL: H CO: H RL: H	CL: H CO: H RL: H	CL: M CO: M RL: M	CL: H CO: H RL: H
Goals and tasks	CL: M CO: H RL: H	CL: M CO: M RL: M	CL: H CO: H RL: M	CL: M CO: H RL: H	CL: H CO: H RL: H	CL: M CO: M RL: M	CL: H CO: H RL: H	CL: H CO: H RL: H	CL: H CO: H RL: H	CL: H CO: H RL: H	CL: H CO: H RL: H
User and dashboard	CL: H CO: H RL: H	CL: M CO: M RL: M	CL: H CO: H RL: M	CL: H CO: H RL: H	CL: H CO: H RL: H	CL: M CO: M RL: M	CL: H CO: H RL: H	CL: H CO: H RL: H	CL: H CO: M RL: H	CL: H CO: H RL: H	CL: H CO: H RL: H
Visualization primitives	CL: M CO: H RL: H	CL: M CO: M RL: M	CL: H CO: H RL: M	CL: H CO: H RL: H	CL: H CO: H RL: H	CL: H CO: H RL: L	CL: H CO: H RL: H	CL: H CO: H RL: H	CL: H CO: M RL: H	CL: H CO: H RL: H	CL: H CO: H RL: H
Data domain & operations	CL: M CO: M RL: M	CL: M CO: M RL: M	CL: H CO: H RL: M	CL: M CO: H RL: H	CL: H CO: H RL: H	CL: M CO: H RL: L	CL: H CO: H RL: H	CL: H CO: H RL: H	CL: H CO: H RL: H	CL: M CO: M RL: M	CL: M CO: H RL: H
Whole meta-model	CL: H CO: H RL: H	CL: M CO: M RL: M	CL: H CO: H RL: H	CL: H CO: H RL: H	CL: H CO: H RL: H	CL: M CO: M RL: M	CL: H CO: H RL: H	CL: H CO: H RL: H	CL: H CO: H RL: H	CL: H CO: H RL: H	CL: H CO: H RL: H

The results yielded good scores (high and medium scores) and only five "low" scores related to the relevance and coherence of the dashboard layout section (due to its simplicity) and the relevance of the visualization primitives and data domain (due to the high level of detail which could be "too low level for end-users").

Other comments of the scores were related to the multiplicity of some associations, such as the [1..1] multiplicity between the *User* class and the *Dashboard* class. Expert 1 commented that "usually dashboards can be accessed by multiple users." Another comment pointed out the lack of detail of the *Dataset* entity in the meta-model.

Regarding modeling user characteristics, some issues were found regarding free value strings. Expert 1 suggested using enumerations instead. Also, modeling user biases was seen as a complex topic, which indeed it is. However, representing them in the meta-model set the foundations for accounting them while designing information dashboards.

On the other hand, another participant indicated that the goal-task model represented in the meta-model could be overly simplified.

Regarding the data operations, an expert commented that “some charts will present the data as is, others will aggregate the data or execute a certain operation. It does not seem scalable to implement all the operations directly on the dataset which the model presented as-is seems to indicate”. Expert 4 pointed out the necessity of accounting for text operations.

In general, most of the experts indicated that the meta-model is complex and hard to read, and that this complexity should be addressed through explanations and instantiation examples.

The results obtained from this expert validation are accessible at https://zenodo.org/record/7037651.

4.2 Theoretical applications

This subsection provides a set of theoretical applications of the dashboard meta-model. These applications are focused on integrating the meta-model with other OMG’s M2 models and on using this artifact as a framework to drive the conceptualization of dashboards at a high level.

4.2.1. Integration with other meta-models

As stated in the methodology section, the dashboard meta-model was developed as an M2 meta-model following the four-layer meta-model architecture proposed by the Object Management Group (OMG) [18]. Using an OMG’s M2 meta-model also unlocks the possibility of integrating the dashboard meta-model with other M2 meta-models. In fact, the dashboard meta-model provides the specification of a visual analytics tool that ultimately can be included in any data-driven ecosystem.

To test this approach, the dashboard meta-model was integrated into a learning ecosystem meta-model [58, 59]. In this case, the dashboard is a part of the learning ecosystem, which is based on a meta-model defined and validated in previous works. The first version of the learning ecosystem meta-model is based on MOF, and the last validated version is an instance of Ecore [10]. Both versions are M2 models. The model has served as a map to develop and deploy the ecosystem in a real-world context.

The integration of both meta-models is possible because both are Platform Independent Models (PIM) at M2 layer (**Figure 21**), although one is instantiated from Ecore (learning ecosystem meta-model) and the other from MOF (dashboard meta-model). To get the holistic meta-model, the dashboard meta-model was transformed into an instance of Ecore using Graphical Modelling for Ecore included in EMF. The changes introduced during this transformation are detailed in [58].

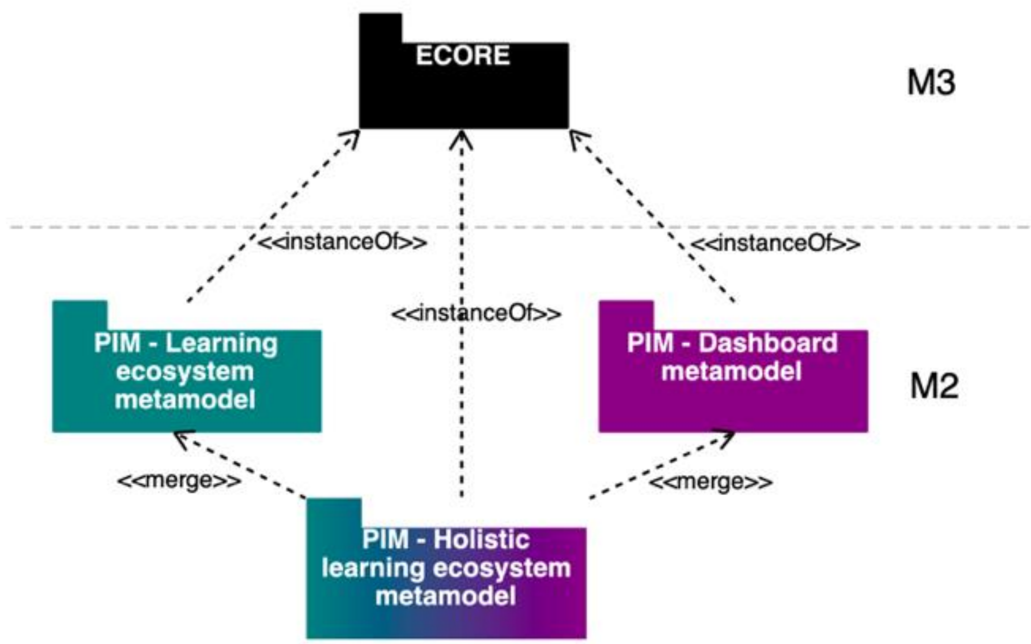


Figure 21. Meta-models organized in the four-layer metamodel architecture. Source: own elaboration, published in [59].

Although the learning ecosystem meta-model proposed solves most of the problems associated to the definition and development of these technological solutions, there are some issues related to the analysis of the information flows and the support to decision-making processes that should be improved.

Connecting both meta-models resulted in the holistic meta-model included in **Figure 22**.

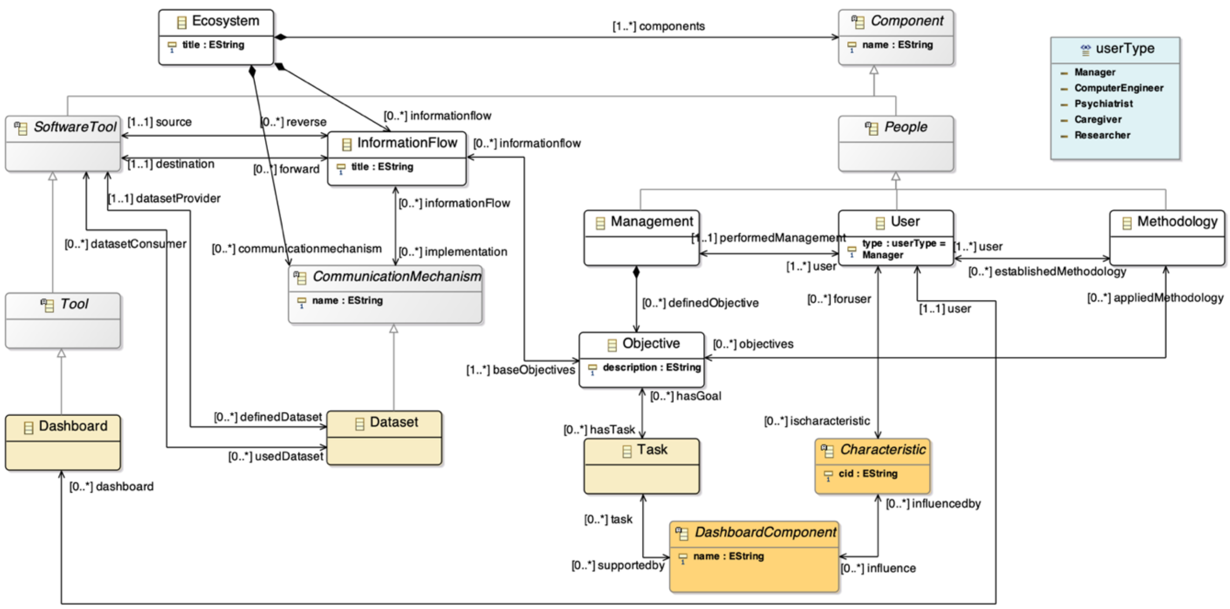


Figure 22. Connection between both meta-models. Source: own elaboration, published in [59].

These two M2-level meta-models are connected by some elements present both in the dashboard meta-model and the ecosystem meta-model. On the one hand, it has been justified the need to include users in the dashboard meta-model because they are the drivers and consumers of the displayed data. The human factor also plays a crucial role in the learning ecosystem meta-model because the technology is defined and evolved to support the users' needs.

On the other hand, there are two relevant elements shared in both meta-models too. The dashboard *Goals* (within the dashboard meta-model) are represented as *Objectives* within the learning ecosystem meta-model. These elements are represented by a set of *Tasks*, and *Information Flows*, respectively. The relevance of these entities is that they are the core of the meta-model because they frame the required components to achieve the goals or objectives set.

To connect both meta-models, the dashboard's *Goal* is merged with *Objective*. The connection between *Goal* and *User* in the dashboard meta-model is replaced by the association between *User* and *Objective* through the *Management*. In this sense, all the goals that support the definition of the dashboard are connected to the management decisions in the ecosystem.

Regarding the *Dashboard*, the main class to instantiate the dashboard meta-model is connected with the learning ecosystem as a *Tool*. Besides, the connection between *User* and *Dashboard*, which has a particular impact on the dashboard meta-model, is included in the proposal. The information flows and tasks are different concepts, so merging them is impossible. For this reason, the *Task* entity is included in **Figure 22**.

On the other hand, a new communication mechanism is included to implement the information flows, the *Dataset*, to represent the integration between the dashboard and other software tools in the learning ecosystem. Also, the dashboard *Component* has been renamed as *DashboardComponent* to distinguish it from the learning ecosystem *Component*.

Finally, the connection between dashboard Characteristic and User appears in the new proposal. Tasks are supported by the dashboard's elements, which are also influenced by the user characteristics to match his or her information requirements.

The dashboard meta-model provides a skeleton that can be adapted to instantiate concrete dashboard solutions. The role of the dashboard is to support decision-making processes through visual analysis. Including the users within the meta-model is crucial, as their goals and data requirements are the drivers of the dashboard configuration process.

This integration was tested by instantiating an M1 model from the holistic meta-model (**Figure 23**). The M1 meta-model was focused on providing a health ecosystem to support caregivers [60].

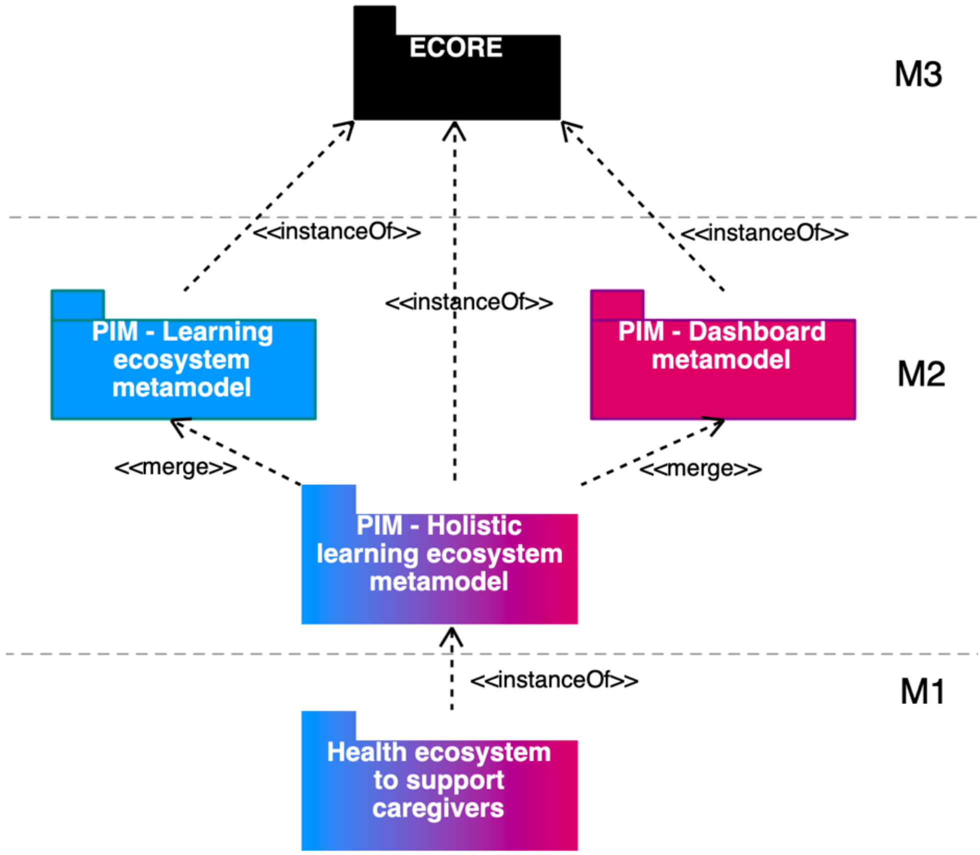


Figure 23. Methodology employed to integrate the learning ecosystem meta-model and the dashboard meta-model organized in the four-layer meta-model architecture of the OMG. Source: own elaboration, published in [60].

A dashboard component was designed to achieve different information goals related to the management of the caregivers. Specifically, two main goals were identified:

- To analyze the relationship between the attention given by relatives and the patient's health.
- To gain insights about the workload of the caregivers.

The prototype for this dashboard was instantiated from the dashboard meta-model and it is presented in **Figure 24**.

Figure 24. Dashboard component prototype for the health ecosystem. Source: own elaboration, published in [60].

The publications associated to this section can be consulted at **Appendix P**. A Meta-Model Integration for Supporting Knowledge Discovery in Specific Domains: A Case Study in Healthcare, **Appendix R**. A meta-model to develop learning ecosystems with support for knowledge discovery and decision-making processes, and **Appendix U**. A Dashboard to Support Decision-Making Processes in Learning Ecosystems: A Metamodel Integration [58-60].

4.2.1. Meta-model as a conceptual framework

The target of the following studies is to test the suitability of the meta-model as a conceptual tool to design data visualizations and dashboards. Although these studies do not rely completely on the dashboard generator, they test how the meta-model can provide guidelines to conceptualize dashboards and data visualization services adapted to domain-specific requirements.

Conceptual instantiation

The dashboard meta-model represents common features found in data visualization tools. However, to obtain a functional dashboard, it is necessary to instantiate the meta-model into a M1 model following the OMG's architecture [18] described in the methodology section, and then transform it into a M0 model, i.e., the functional product. To test if the meta-model supports the instantiation of real-world dashboards at the M1 level, a visual analytics component was designed to support knowledge management in a health ecosystem for caregivers [61].

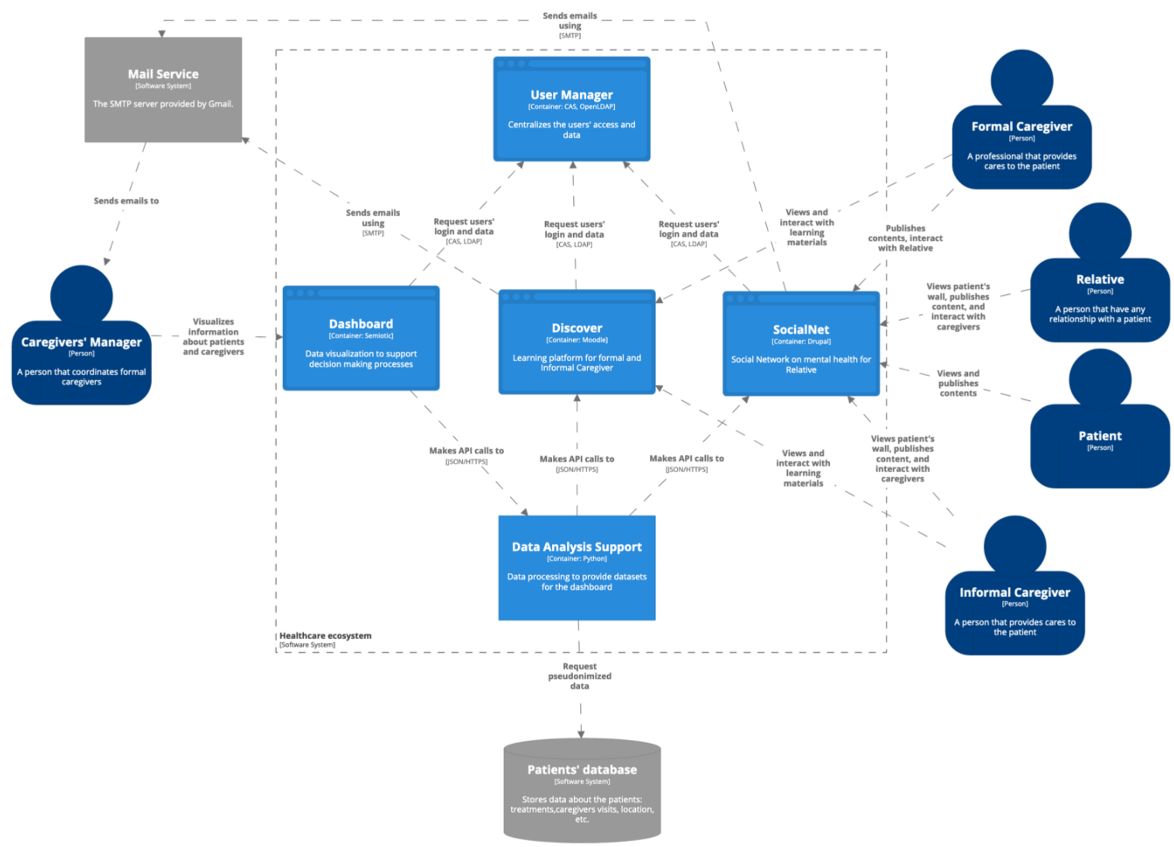


Figure 25. Healthcare ecosystem architecture using the C4 model. Source: own elaboration, published in [61].

The aim of the technological ecosystem for caregivers is to support the learning and knowledge management processes to develop and enhance the caregiving competences both at home and in care environments of formal and informal caregivers [43]. In this context, a dashboard, and a data analysis support component

(**Figure 25**) to assist decision making processes was introduced. These components are focused on supporting caregivers' managers to make decisions about the workload of the caregivers, and the activity of the patients based on the insights from the different components of the ecosystem.

A series of goals were extracted from the caregivers' manager profile. Caregivers' managers have a series of objectives regarding the analysis of the collected data. These requirements can be classified into two main goals: insights about the patients' relatives and insights about the workload of the caregivers. The generated knowledge from the reached insights can support decision-making regarding the distribution of caregivers to reduce workload and the impact of the patients' relatives on their healthcare.

Figure 26 shows the instantiation of the meta-model's section that addresses the users' goals. On the one hand, the first goal regarding relatives has the following description: "To analyze the relationship between the attention given from relatives and the patient's health". To achieve that goal, two lower-level tasks arise: observe the relatives' data and observe the patients' data. Given the fact that the goal asks for "analyzing a relationship", a dashboard component must support the mentioned tasks to enable managers to reach insights. The selected component for this matter is a scatter chart.

On the other hand, it is necessary to provide support to the second goal: "To gain insights about the workload of the caregivers". Workload can be visualized in different ways and can involve different variables. In this case two components have been selected to support the second goal: a heat map and a tree map to let managers identify patterns or relevant data points regarding the caregivers' distribution along time and among patients.

Each instantiated visualization displayed in **Figure 26** can be decomposed on primitives that support the introduced requirements. For example, the scatter chart can be composed by different primitives that will hold different data dimensions encoded through different channels (**Figure 27**). In this case, two data variables will be encoded through the scatter chart: the patient's health and the patient's given attention (by his/her relatives). These two variables are encoded through two linear scales that will position each data point (circles) within the visualization container. The scales are visible through two axes: X Axis and Y Axis. Each axis has a label to ease the readability of the visualization.

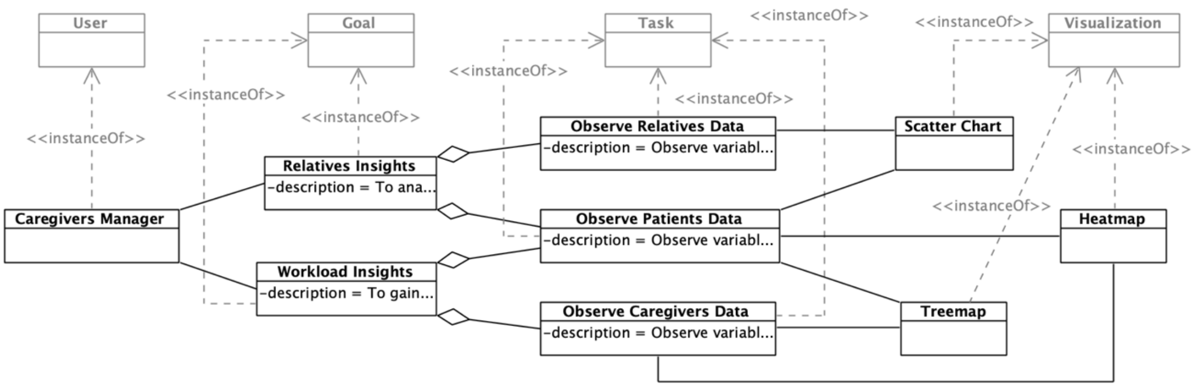


Figure 26. M1 model for a dashboard following the user goals and tasks. Source: own elaboration, published in [61].

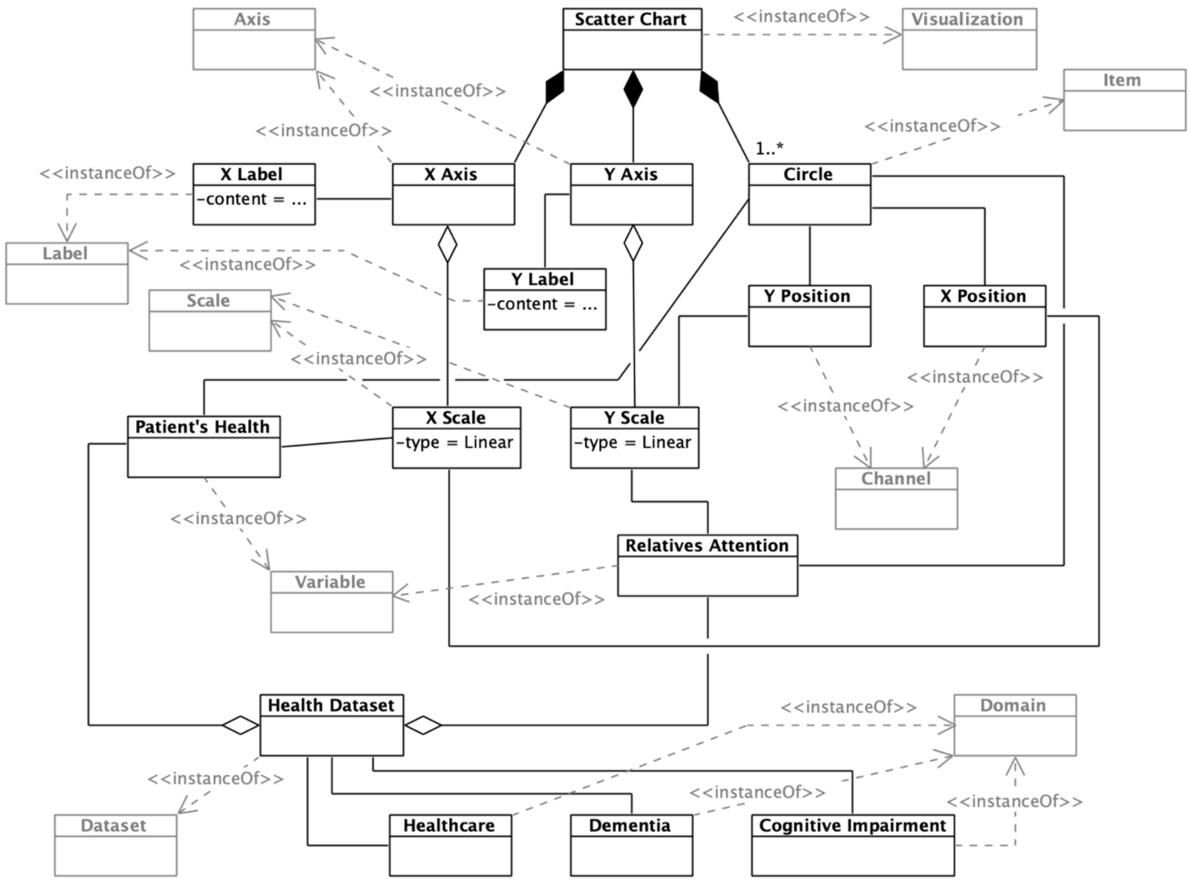


Figure 27. M1 model for a scatter chart. Source: own elaboration, published in [61].

The whole instantiation process for this case study is detailed in **Appendix M**. Dashboard meta-model for knowledge management in technological ecosystem: a case study in healthcare [61].

Bias identification

Delegating decisions solely in data might turn out to be a two-edged sword. Data not only can be wrong or false, but it can also be incomplete, and making decisions using wrong data leads to wrong decisions. There are several cases in which relying on the wrong data has provoked undesired results, mostly because of data bias or even algorithmic bias [170-172].

In this sense, visualizations could hide data issues by lifting the attention from the analysis process carried out on the raw data to the discovered patterns. Patterns can be seen as shortcuts that tell users properties about the data, for example, if there are correlations among the visualized variables [173]. But visual analysis shouldn't be reduced to just the identification of patterns and to trust them blindly, because patterns can likewise lead to wrong conclusions [174].

However, identifying flaws in data is not a trivial task; bias, beliefs, and uncertainty can show up both at data collection time and analysis time, resulting in distorted insights. For these reasons, bias is included as a part of the meta-model's user entity. Although a complex concept, it is important to account for bias when designing data visualizations and information dashboards.

In this context, the notion of automatizing the identification of bias through a study focused on detecting aggregation issues [62] was explored. A simple workflow has been considered to automatically seek for issues involving the Simpson's paradox and underrepresentation of categories, a problem in which the patterns individually identified in different data groups contradict the patterns found when these groups are combined [175, 176].

Each categorical variable is considered as a potentially influencing variable. As it will be discussed, this methodology is limited to the available variables within the dataset. If the whole dataset has a small set of categories, the results would not be as useful as it could be with a richer dataset.

The workflow follows a naïve approach to detect Simpson's paradoxes [177]:

1. Every possible grouping at any possible level is computed on categorical values to obtain a set of potential disaggregation variables.
2. When the user visualizes data, the current aggregation level is retrieved (i.e., the categorical columns used to group the data).
3. These data are then grouped by the variables identified in the first step.

4. The results of the performed disaggregation are sorted and compared with the original scenario (i.e., the aggregated data values) trend.
5. If the disaggregation results differ from the originally aggregated results (a threshold can be defined to specify which proportion of values differ from the original trend to be considered in the paradox), the Simpson's paradox is contemplated for the disaggregated attributes.

However, even visualizing the disaggregated data by the identified attributes in the fifth step, there could still be aggregation issues if data are aggregated by a function such as the mean, mode, ratios, etc. These functions can, in turn, distort the reality of data.

To avoid relying on aggregation functions, when the detected instances of the Simpson's paradox are inspected, a sunburst diagram complements the display to give information about the raw data sample sizes regarding the disaggregated values. The primary purpose is to have another perspective of data, drawing attention over potential underrepresentation or overrepresentation in datasets.

This research can set the foundations for a better characterization of the bias concept in the meta-model, as well as for the definition of OCL rules to mitigate its side effects. The detailed proposal can be consulted in **Appendix Q**. Aggregation Bias: A Proposal to Raise Awareness Regarding Inclusion in Visual Analytics [62].

Automatization of the dashboard instantiation process

As outlined in the methodology section, the employed framework involves two software engineering approaches: meta-modeling and the software product line paradigm. The combination of meta-modeling to obtain a domain abstraction with the SPL philosophy of systematically reusing assets provides a powerful framework for building families of products.

However, although the SPL paradigm can support the generation of dashboards, the features still need to be manually selected. The configuration process can be a challenge, as users might not know what dashboard configuration better suits their needs, and thus, the dashboard designer would not be able to select the appropriate dashboard features. In this case, the configuration process asks for automation.

For these reasons, a workflow to address the challenge of automatically configuring a product line of dashboards based on user necessities is proposed. First, data characteristics need to be taken into account to build appropriate charts [144], as some visual encodings are not suitable for some types of variables.

However, while a chart can be correctly built, it can be inappropriate if its context and audience are not considered. User goals are crucial, as visual metaphors, supported tasks or displayed data depend on what are the user's data necessities, as well as users' characteristics, because the visualization literacy, domain knowledge or even bias, can compromise the users' insight delivery process.

In the end, the problem is summarized as the necessity of tuning the SPL core assets parameters to optimize the user experience and the effectiveness of the generated dashboard.

Users' goals, characteristics, and datasets must be structured into machine-readable documents to allow their processing and to automatize the process. The output as well, in this case, a selection of features, can be transformed into a structured file readable by the code generator which will inject the specific parameter values into the templates to generate the final dashboard.

A proof-of-concept for integrating inferred configurations into the dashboards generative pipeline was carried out. CompassQL [178] was selected as the visual mapping/recommendation engine to test the viability of this workflow. CompassQL enables the automatic selection of channels given the dataset and the variables to visualize.

The integration of this recommendation engine is made through the code templates that support the SPL. CompassQL recommendations are obtained and stored and structured into the configuration files that will drive the generation process.

In this sense, the design of the generative pipeline allows the connection of any kind of recommendation engine if the configuration files comply with the meta-model entities and relationships. Configuration files can be seen as M1 models that are instantiated from the meta-model, and the final code M0 models obtained from the previous M1 models.

The same applies for the code templates. The content of these templates could be directives of any programming language. This improves the flexibility regarding the selection of technologies for specific contexts.

The detailed research for this proof-of-concept can be consulted in **Appendix N**. Connecting domain-specific features to source code: towards the automatization of dashboard generation [63].

Dashboards as a Service

As described in Chapter 3, the meta-model and generative pipeline provide means to generate information dashboards on demand through the configuration of fine-grained features. However, designing a dashboard involves other low-level tasks such as cleaning data, selecting the right data encodings or dashboard configurations, generating source code, etc.

The results derived from the last subsection's study proved the viability of encapsulating different tasks and using their outputs as inputs for the next task, as it was the case of the selection of proper visual encodings to subsequently generate a dashboard using the obtained specification.

Following this idea, a technological ecosystem to manage the generation of information dashboards that can be tailored to fine-grained features is proposed. Applying an ecosystem approach to this matter benefits users by providing access to a whole generative pipeline for information dashboards, with components that interact and collaborate among them to offer powerful features, but also with independent services for more specific tasks.

In fact, this ecosystem would support knowledge management, because all the tacit knowledge associated with dashboard design processes, design decisions, data transformations, etc., would be managed by the different components.

The ecosystem is planned as a holistic set of well-defined components that provide unitary services, but that can also be combined to obtain a complete pipeline. Every service has well-defined interfaces that enable the connection of information flows among them. These services provide support for the generation of information dashboards that compile with the previously presented meta-model.

One of these services is the dashboard generator (**Figure 28**), based on plain JavaScript through the D3.js framework to allow better integration with external services avoiding other dependencies. The dashboard generator service accepts HTML requests containing information about the visualization component to craft. Specifically, this service is developed as an Application Programming Interface (API) in which the input is a JSON object with the configuration of an entire dashboard or a single visualization:

- Information about the dataset or datasets to be displayed. Data sources could be external APIs or files.
- The disposition or layout of the elements.
- The features of the visualization:
    - Number and type (X position, Y position, size, color, etc.) of visual channels.
    - Visual mark type (bar, circle, topographic, arc, etc.).
    - Dataset's variables to be represented.
    - Interaction events and effects [53].

The service processes this JSON object; then, the source code is generated using the previous section's code templates. These source code files are returned to the client, which could embed them in its own applications or use them standalone.

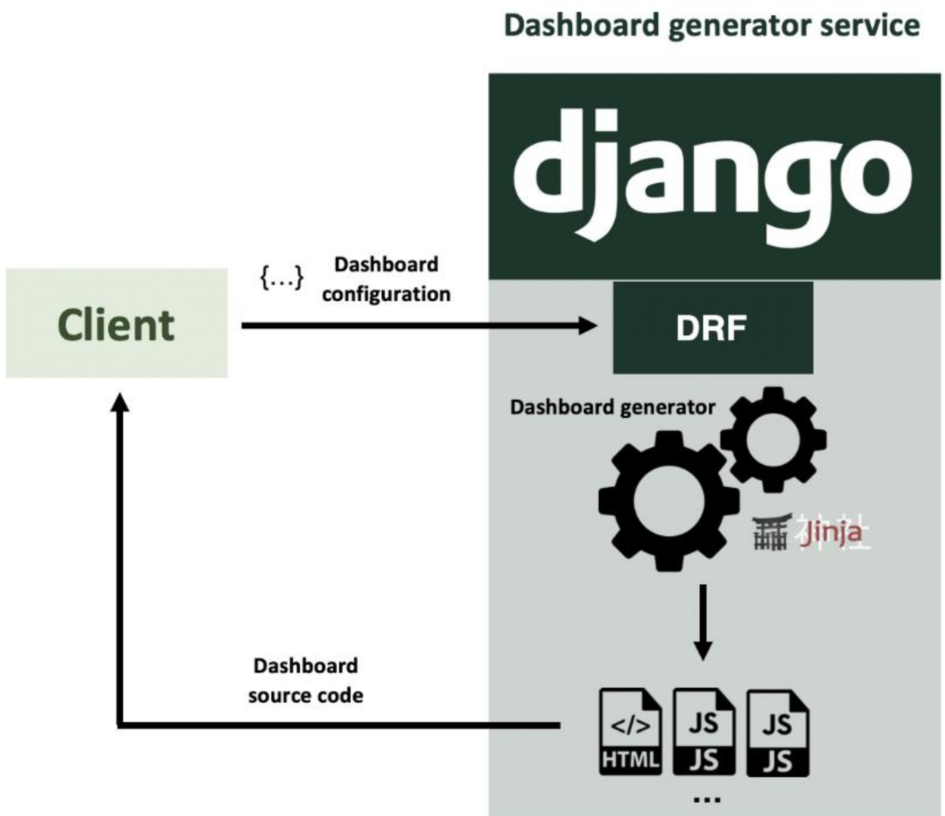


Figure 28. Schematic view of the dashboard generator service architecture. Source: own elaboration, published in [64].

The detailed architecture definition is published in [64] and can be consulted in **Appendix W**. Towards a Technological Ecosystem to Provide Information Dashboards as a Service: A Dynamic Proposal for Supplying Dashboards Adapted to Specific Scenarios. This approach has been successfully employed in the development of some practical applications and case studies described in the following sections.

4.3 Practical applications

This subsection describes three practical applications of meta-modeling and SPLs in the domain of data visualization and information dashboards. The following studies focus on how to transform the meta-model (M2 model) into an instance adapted to a specific context (M1 model), and how to finally transform this later model into code (M0 model), i.e., the final, functional product.

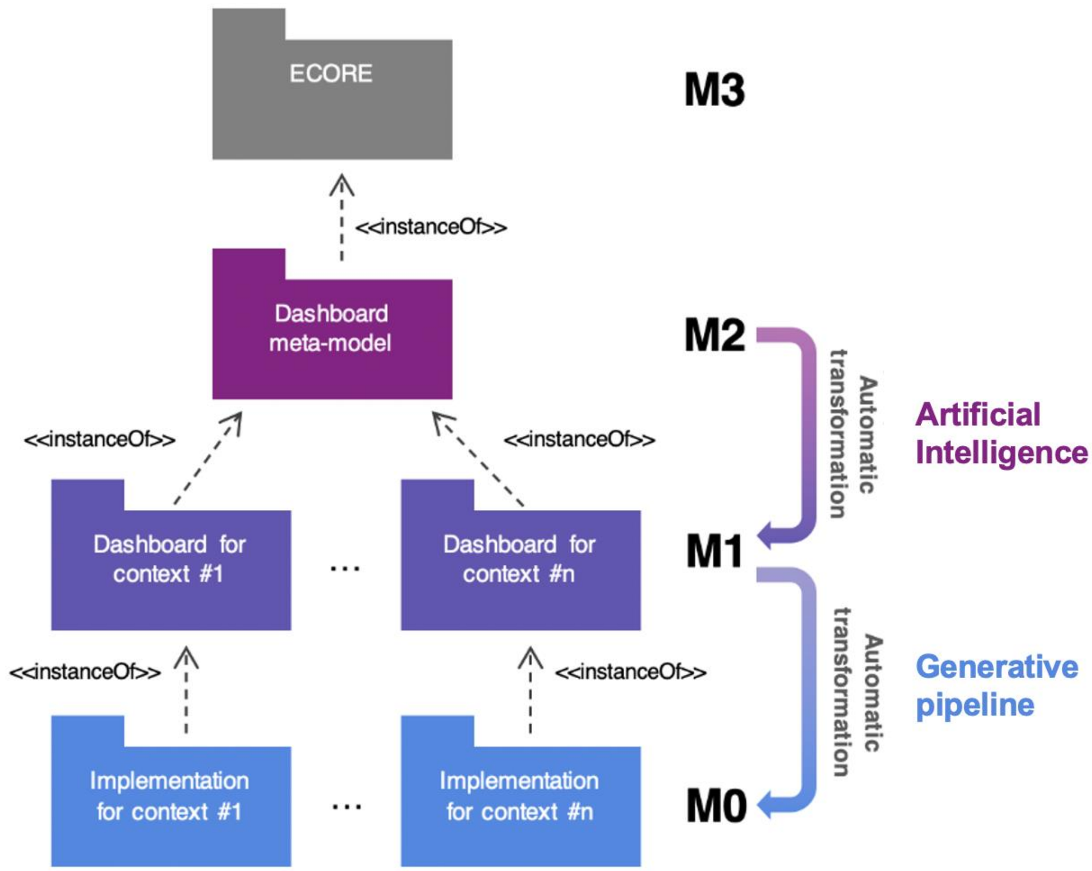


Figure 29. Location of the different models following the MDA architecture [89]. Source: own elaboration.

In this regard, the main challenge was the automatization of the model transformations. The M1-M0 transformations have been addressed through the previously described generative pipeline, while the M2-M1 transformations have been explored through the application of AI approaches (**Figure 29**). The final product derived from these studies is a graphical instantiator whose architecture and interface are driven by the meta-model and enables both the instantiation and the generation of information dashboards visually.

4.3.1. M1-M0 transformations: Automatic generation of code

One of the main goals of combining the MDD and SPL paradigms was to automate the generation of information dashboards. In this sense, the focus is put on the transformation of a model instantiated from the dashboard meta-model into real, functional code. The generative pipeline employed to tackle this transformation was previously detailed in Chapter 3.

First, the automatic generation of information dashboards was tested in the context of education and Learning Analytics (LA) [65] by using part of the records from the dataset collected in [179]. This dataset contains demographic, interactions, and performance data of Open University (OU) students.

Then, a basic information dashboard was generated. The dashboard contains two visualizations representing the scores of the tasks performed by each student and the range of scores (minimum and maximum) per task, respectively (**Figure 30**).

The details of this work can be consulted in **Appendix V**. Beneficios de la aplicación del paradigma de líneas de productos software para generar dashboards en contextos educativos [65]. The results derived from this proof-of-concept set the foundations to generating more complex dashboards based on real-world requirements.

In this sense, a case study to generate information dashboards in the context of a Ph.D. Programme [66] was carried out. First, a requirement elicitation process was conducted to model the dashboards that would be generated.

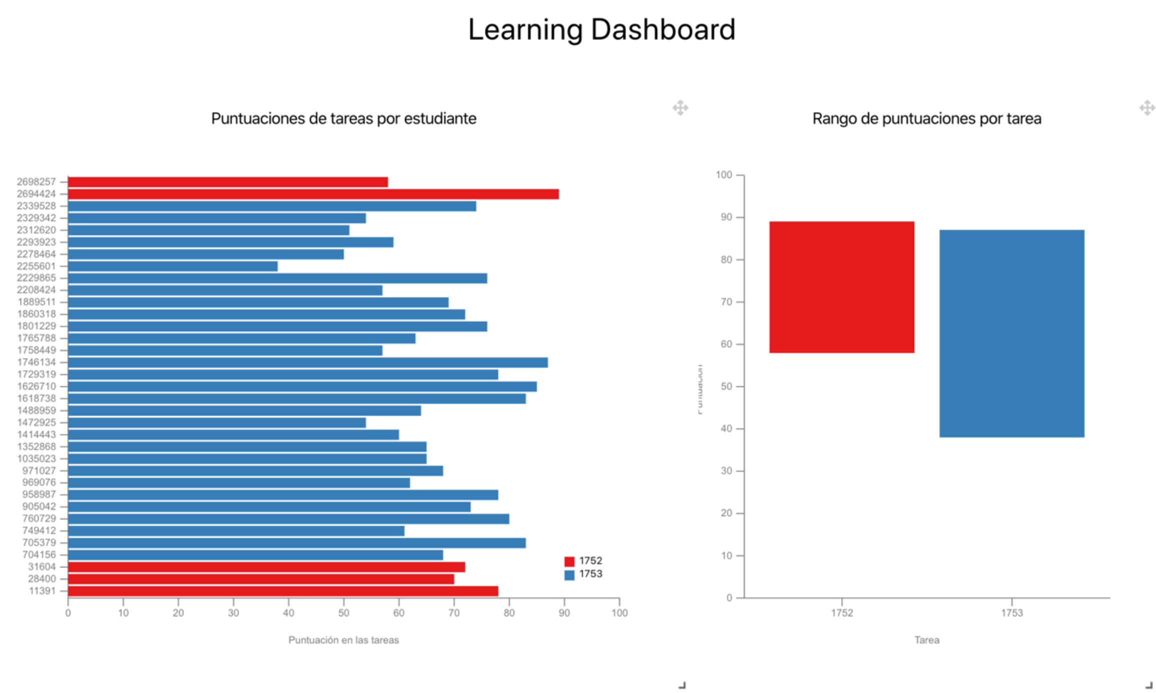


Figure 30. An information dashboard generated using the SPL approach with LA data. Source: own elaboration, published in [65].

The requirements elicitation process included an interview with a member of the Ph.D. Programme's quality committee to understand which data could be displayed in a potential information dashboard.

Once the interview was carried out, an instrument to collect information regarding the users' information requirements was designed. First, some demographic variables are collected to contextualize the sample: age, gender, and birthplace.

The next section focuses on the collection of the user situation within the Ph.D. Programme: role, research lines, Ph.D. modality, and academic year (in the case of Ph.D. students) and the number of Ph.D. thesis being directed (in the case of Ph.D. advisors). Questions regarding the usage of the Ph.D. portal were also included in this section to understand how users employ this platform.

Finally, the last section included questions regarding users' past experiences with information visualization and regarding the users' information requirements for a hypothetical Ph.D. Programme dashboard.

Following the results, three information dashboards were generated using the generative pipeline: a dashboard for students (**Figure 31**), for advisors (**Figure 32**), and for managers (**Figure 33**).

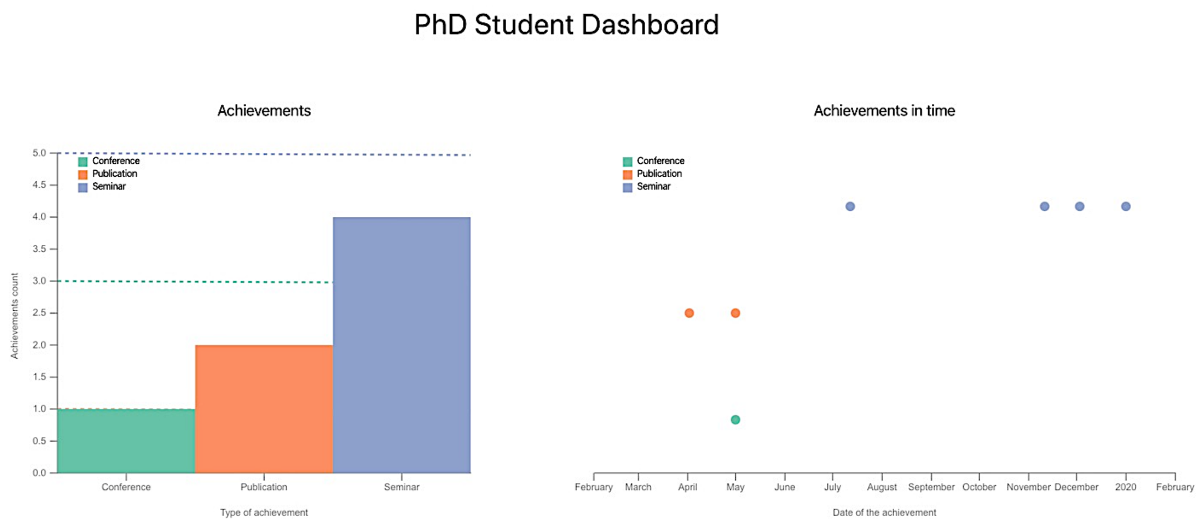


Figure 31. Ph.D. student dashboard proposal. Source: own elaboration, published in [66].

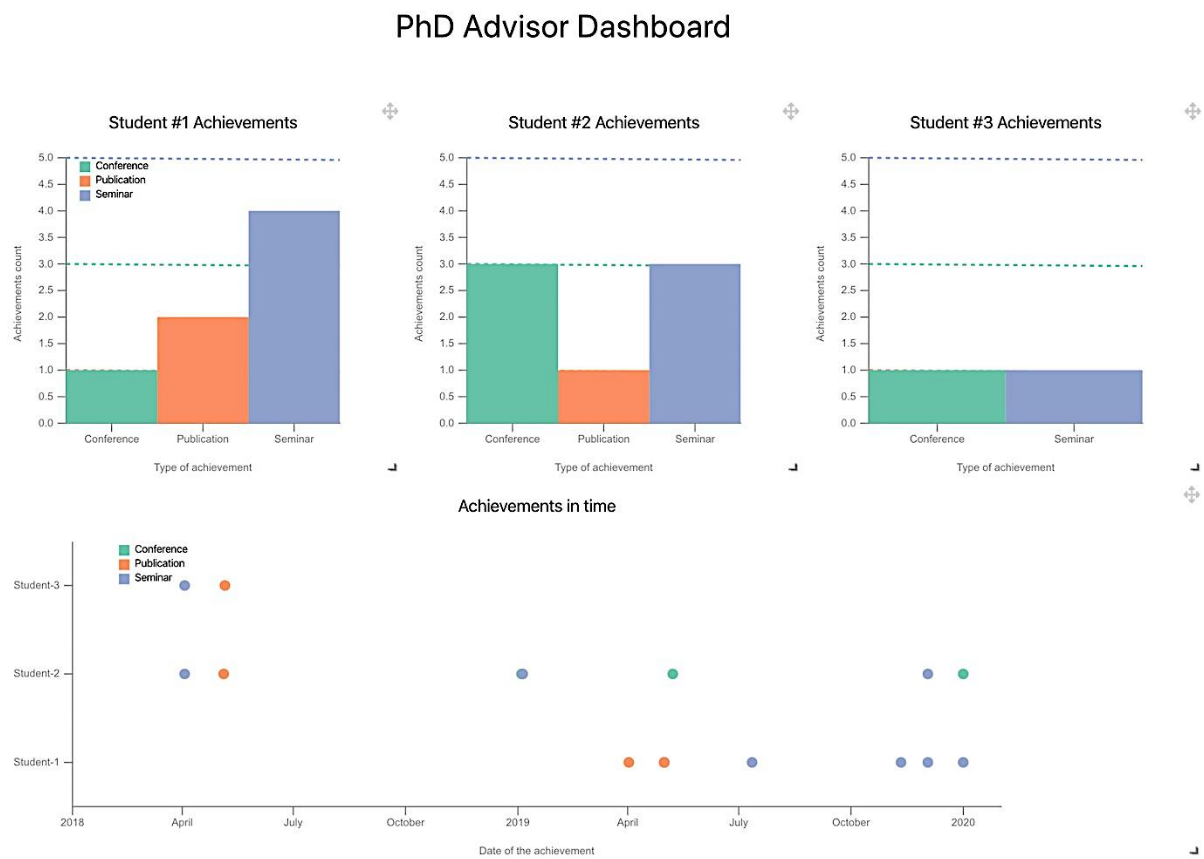


Figure 32. Ph.D. advisor dashboard proposal. Source: own elaboration, published in [66].

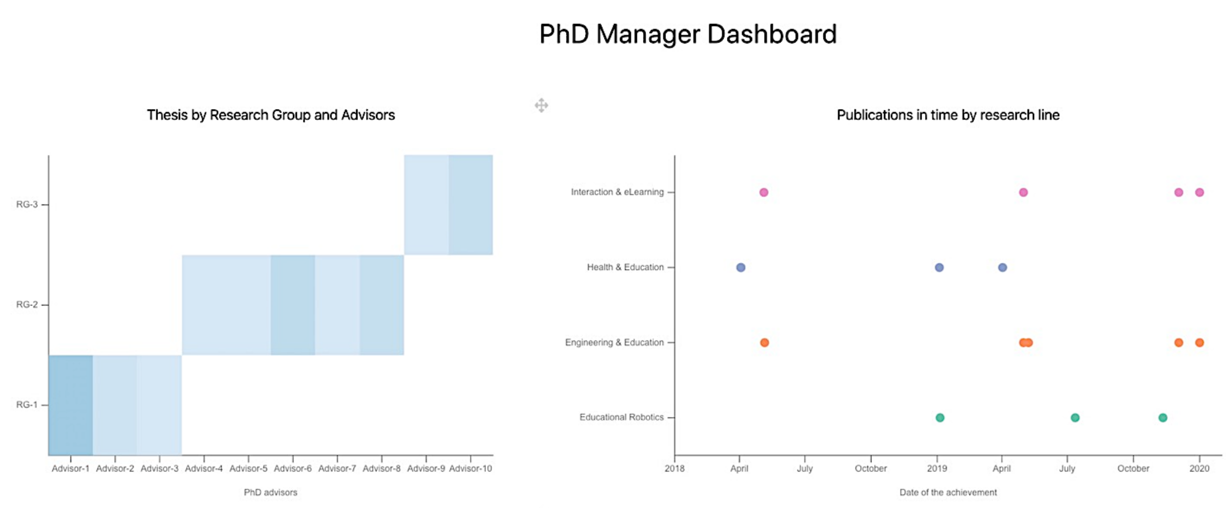


Figure 33. PhD manager dashboard proposal. Source: own elaboration, published in [66].

This study and the detailed requirement elicitation process can be consulted in **Appendix S**. Generating Dashboards Using Fine-Grained Components: A Case Study for a PhD Programme [66].

These proposals were finally implemented in the Ph.D. Programme of Education in the Knowledge Society at the University of Salamanca (Spain) [180]. An usability study was conducted to score the integration of data visualization into the Ph.D. portal using the System Usability Scale (SUS) questionnaire [181]. 35 participants (including Ph.D. students, advisors, and managers) answered the SUS questionnaire. The analysis of the answers yielded the following outcomes:

- The average perceived usability of the Ph.D portal visualizations is **75.36**, which can be considered as a good SUS score, as it is above the average SUS score (68.00) and falls around the 75th percentile (interpretation based on the studies done in [134, 182]).
- On the other hand, the perceived learnability is **74.29**, a slightly lower score than the usability (**75.63**), both being acceptable and good scores following the adjective scale of the SUS [182].

Figure 34 summarizes these results, also including the individual scores for every participant (represented by overlapping circles) across the three dimensions considered: total SUS score, Usability score and Learnability score.

The detailed results of this case study can be consulted in **Appendix X**. Following up the progress of doctoral students and advisors' workload through data visualizations: A case study in a PhD programme [67].

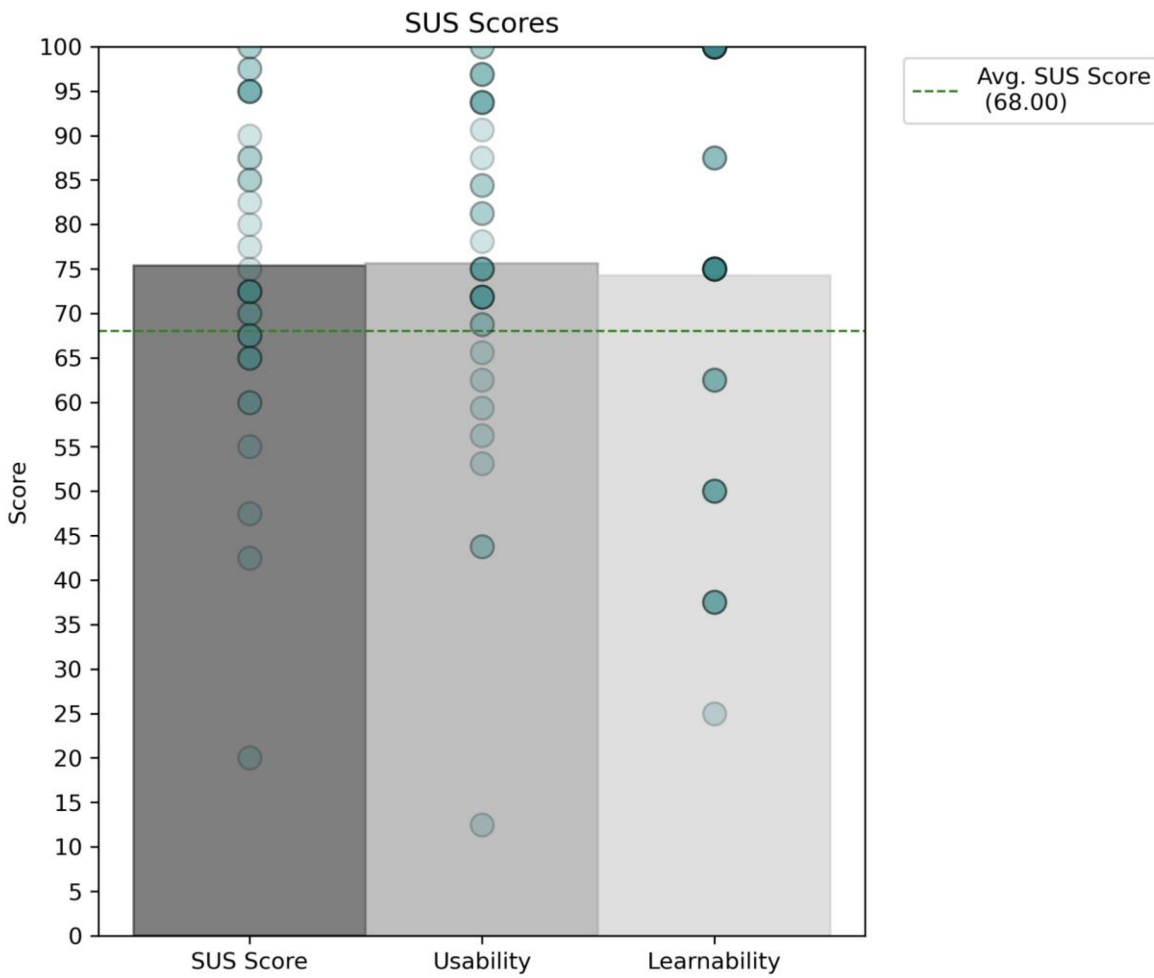


Figure 34. Visual representation of the SUS questionnaire results regarding the PhD portal data visualizations' usability and learnability scores. Source: own elaboration, published in [67].

4.3.2. M2-M1 transformations: Combining meta-modeling with AI

Although M1-M0 transformations can be effectively addressed through the generative pipeline and the SPL paradigm, automatizing M2-M1 model transformations remains a challenge. The main issue is that this transformation requires explicit guidelines and domain knowledge [144] to obtain effective M1 models of dashboards and visualizations.

Following this notion, the idea of integrating AI mechanisms into the generative pipeline was explored. AI would be in charge of learning good practices and guidelines in data visualization design and selecting the best dashboard configuration given the specific necessities of the target context.

In this context, a study to test the idea of using the generative pipeline to train ML models and identify potentially harmful configurations in data visualizations [68] was carried out. Different ways of obtaining visualization datasets were evaluated to use them as a training input for a ML algorithm. The first approach was to research different news portals in pursuit of misleading or confusing information visualizations, to subsequently structure their features taking as a reference the previously presented meta-model. However, this approach was time-consuming, and the number of tagged visualizations obtained through this method was not very significant given the time devoted.

That is why the final decision was to generate a dataset of different information visualizations by the developed generative pipeline. In addition, by using this approach, the configurations of the generated visualizations are already structured and prepared for their processing and use as an input, which also saved time and allowed to focus more thoroughly on the tagging process.

This approach was tested with a tri-variate dataset with two numeric columns and one nominal column, so the combinations of displayed variables are also constrained.

Finally, a basic tagging infrastructure was included into the generated source code to ease the tagging process through two buttons that automatically save the configuration and the given tag and a text field to store notes regarding the tagging process. **Figure 35** shows a screenshot of the generated tool.

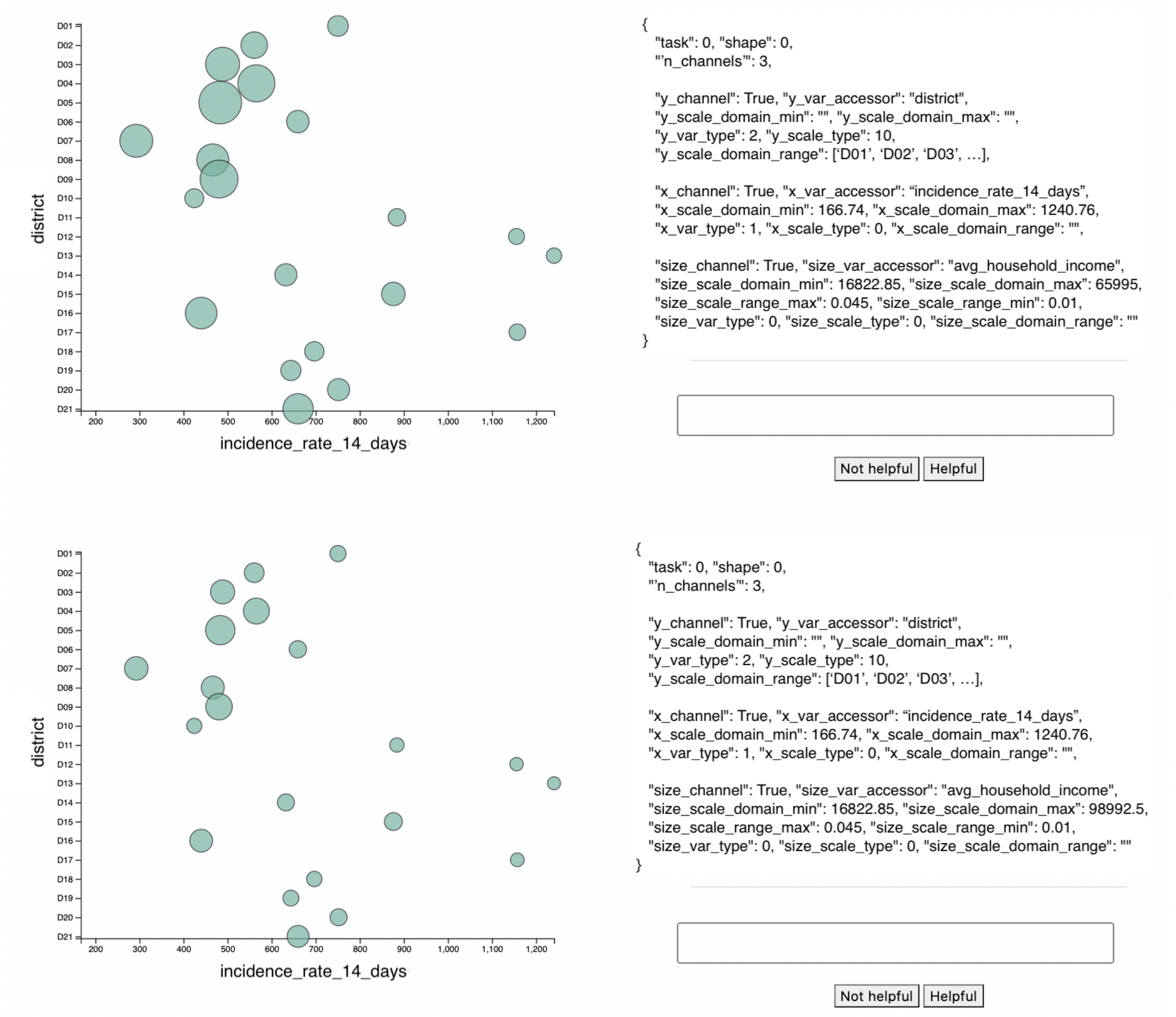


Figure 35. A demonstration of the generation process output, with the different visualizations (left side of the figure) followed by their configuration and the labeling functionalities (right side of the figure). Source: own elaboration, published in [68].

The tagging infrastructure was employed to create a training dataset of misleading visualization. The full process is divided into two phases (**Figure 36**), a first phase focused on generating the visualization dataset described above, and a second phase for the visualization tagging process. Both phases different experts in areas related to data management and visualization from the GRIAL research group [16, 183]. Each generated visualization was tagged as "helpful" or "not helpful" based on the experts' knowledge.

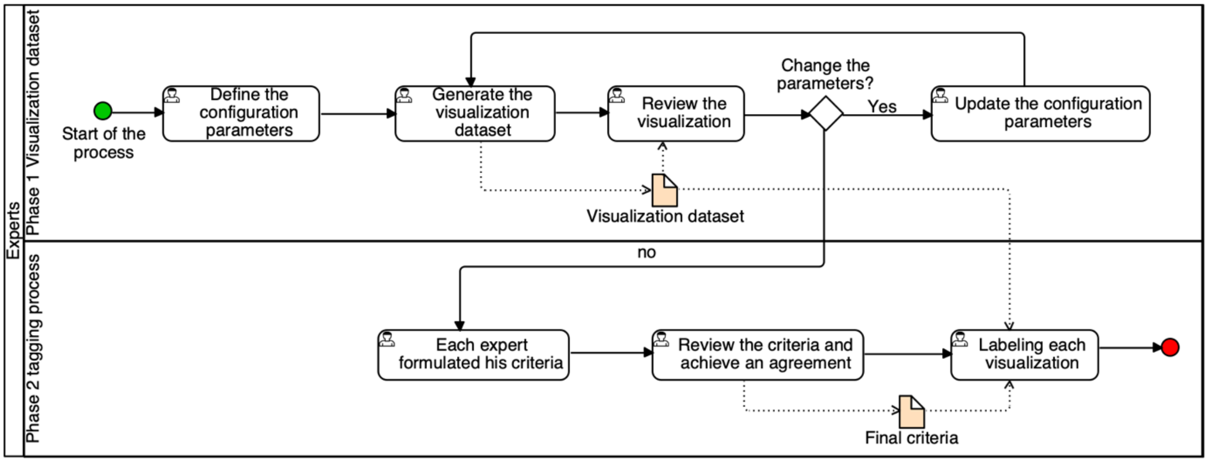


Figure 36. Methodology followed to create the training dataset. Source: own elaboration, published in [68].

Once the dataset was generated, different ML models were trained: Naïve Bayes, Support Vector Machine (SVM), Ada Boost and Random Forest (RF). The RF classifier obtained the best results in terms of accuracy, precision, recall and F1-score (**Table 12**). Given these results, the RF classifier was chosen to test the outcomes of individual predictions by introducing other visualizations' values manually.

Table 12. Classification report of the Random Forest classifier. Source: own elaboration, published in [68].

Class	Precision	Recall	F1-score
0	0.98	1.00	0.99
1	1.00	0.98	0.99
Accuracy: 0.989			

Although the accuracy of the RF model is very high, it needs to be thoroughly discussed. One of the reasons for the high accuracy is because the model is mimicking the criteria previously defined to classify each visualization. However, it is clear that the model has learned the most important features from the classification process. The top important features were the maximum and minimum values of the X, Y, and size's scales domain.

This result aligns with previous research found in the literature, in which the definition of a visualization scales' ranges is determining for its proper understanding [184]. The detailed research related to this work can be consulted in **Appendix Y**. Proof-of-concept of an information visualization classification approach based on their fine-grained features [68].

4.3.3. MetaViz: a graphical instantiation application

As introduced in the beginning of this subsection, the previous studies provided the context and mechanisms to integrate both transformations (M2-M1 and M1-M0) into a platform that enables users to instantiate models from the dashboard meta-model. This platform, named MetaViz, not only makes the meta-model more accessible for practical applications, but also implements the mechanisms to address complex model transformations visually.

The architecture of MetaViz relies on the previously described MDA paradigm's concepts. The MDA layers (specifically the M2, M1, and M0 layers) are materialized in the system through different software modules. The reason why the M3 layer is excluded from the final system's architecture is that the meta-model was built as a standalone artifact using Ecore (that is, it is an instance of the M3 layer as explained in the methodology section), but directly integrated as a set of rules and constraints into the codebase of MetaViz's interface.

Figure 37 outlines the architecture of MetaViz. The backend is developed using the Django framework (https://www.djangoproject.com/), and it is divided into four main modules:

1. **Workspace module.** This module acts as an API, and it oversees receiving the current dashboard configuration that is being crafted through the user interface to subsequently send the information to other modules.

2. **Recommendation module.** This module is still under development, but it will implement a rule engine based on the meta-model rules and heuristics to yield recommended dashboard configurations using the current and previous ones.
3. **Dashboard management module.** The dashboard management module has two main goals: to persistently store and retrieve the last dashboard configuration and to format it to feed the dashboard generator module.
4. **Dashboard generator module.** The last module implements a dashboard generator based on Jinja2 code templates [70].

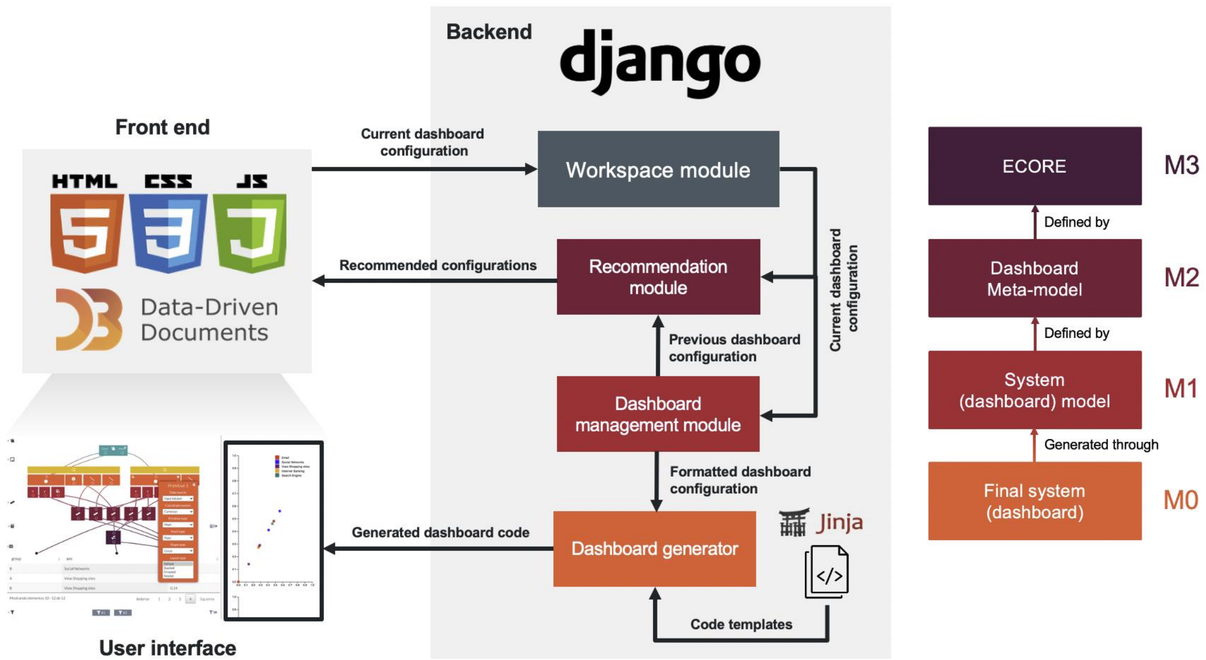


Figure 37. MetaViz architecture and connections among its different modules. The right section of the figure shows the MDA architecture layers adapted to this context. The color of each module matches the MDA layer that it addresses. Source: own elaboration.

Not only the software architecture is based on the MDA layers, but also the user interface. **Figure 38** shows the correspondence of the different layers of MDA and the interface components of MetaViz. The toolbar (M2) enables users to add new primitive elements into the workspace (M1). The configuration parameters of each element also rely on the meta-model attributes and relationships, as can be seen in the dialogue box opened in **Figure 38**.

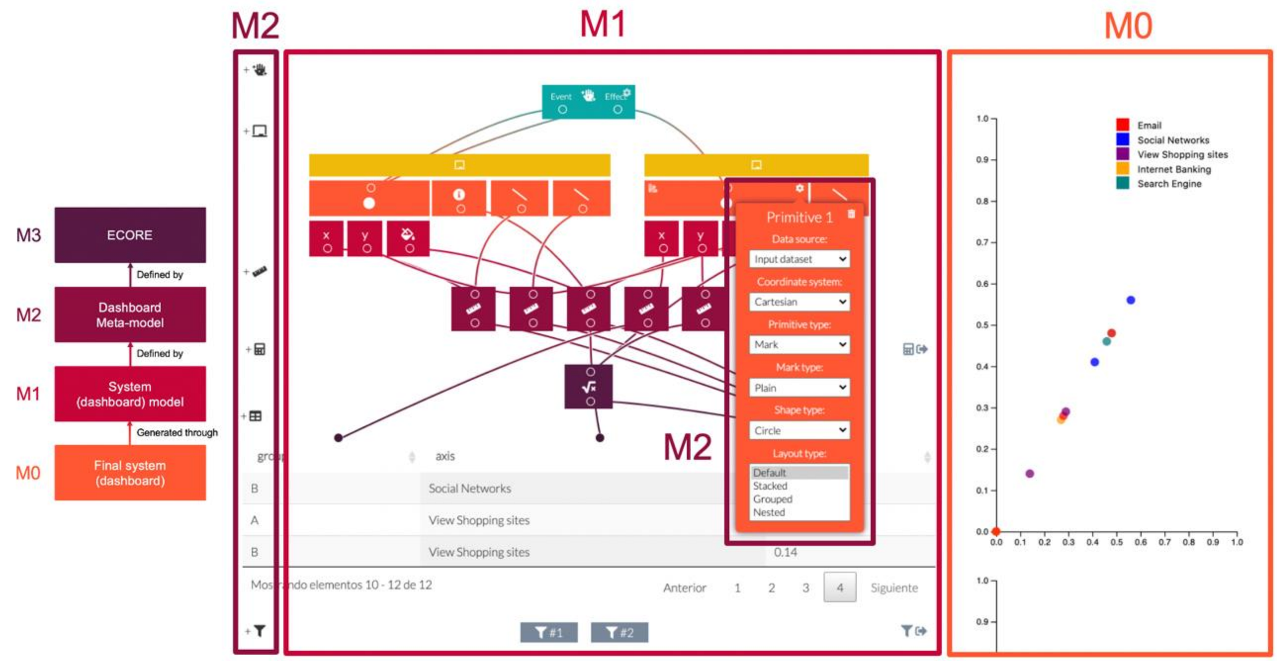


Figure 38. MetaViz interface. The left section of the figure shows the MDA architecture layers adapted to this context. The color of each container matches the MDA layer that it addresses. Source: own elaboration.

The workspace (M1) allows the instantiation of new information dashboards by connecting and configuring the different "meta-elements." Finally, the canvas (M0) shows in real-time the generated dashboard or visualizations. The elements that can be included in the workspace are listed and detailed below.

1. **Dataset.** It matches the *Dataset* and *Variable* elements from the meta-model. The interface provides a table to visualize the raw dataset.

   - **Filters.** A filter is an operation that can be applied to the dataset to yield a subset of rows and columns that match a certain condition. Although this element matches the *Operation* class of the meta-model, it was decided to separate it as an element by itself in the interface to make the active filters easier to spot and manipulate.

2. **Operations.** Operations are transformations that can be applied to the input dataset. In this case, it is possible to aggregate data (to obtain the mean, the count, the standard deviation, etc. of a column) or even to perform operations among columns or scalars (subtractions, additions, multiplications, divisions, etc.). This interface element matches the meta-model's *Operation* class, as its name implies.

3. **Scales.** Scales are the "transition" between the data space to the visual space. They allow the materialization of data into visual elements. They match the *Scale* element in the meta-model.

4. **Visualizations.** Visualizations are "containers" that can hold different primitives (as it can be seen in the *Visualization* meta-class of the meta-model).

   - **Primitives.** This concept is employed to identify the different elements that can be included in a visualization. In the meta-model, these elements are *Marks, Legends, Axes, Resources and Annotations.* At this moment, the system enables users to include the first three elements.

     - **(Visual) Channels or Encodings.** If a primitive is identified as a *Mark*, then it is possible to add different visual channels or encodings (*Channel* class in the meta-model), which are the elements that ultimately encode [10] the input data into visual marks' characteristics (position, size, color, stroke color, text, etc.).

5. **Interactive behavior.** The dashboard meta-model also captures interactive behavior. The support of user interactions can be crucial to make dashboards and visualizations more effective and engaging. This interface element represents the meta-class *Interaction*, and it is possible to configure the trigger event (such as hovering over a primitive) and the interaction's effect (such as highlighting the target primitive or showing a tooltip). Interactions, as it is specified in the meta-model, affect the primitives of a

visualization (and not the visualization as a whole), supporting a fine-grained specification of interactive behaviors [185].

Figure 39 provides the instantiation workflow, in which the "tangible" elements (the visual marks, axes or legends) are inspected first to see if they need to be rendered. If not, the generation process stops and waits for the user to add new elements to the model, or to change any other configuration.

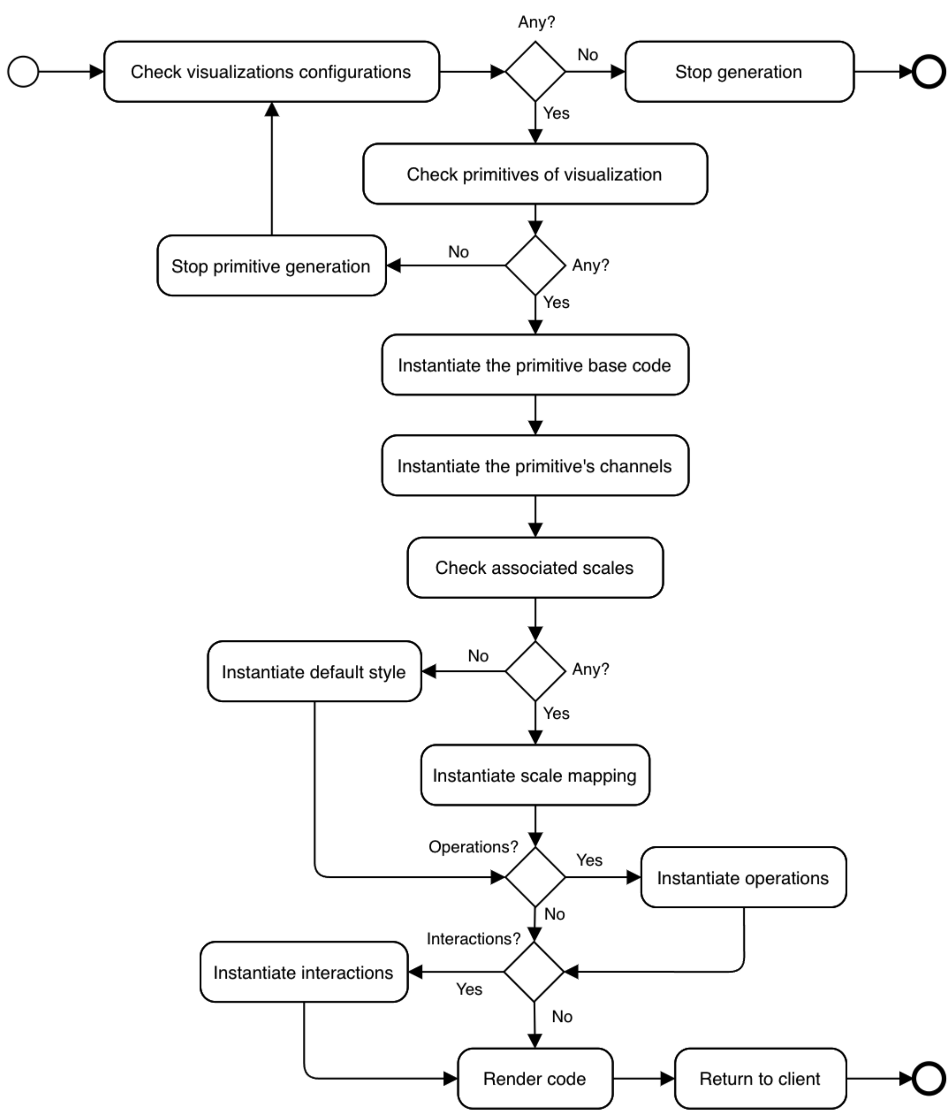


Figure 39. Business Process Model of the instantiation workflow. Source: own elaboration.

MetaViz allows the instantiation of different data visualizations. First, it is necessary to upload a dataset (for example, a CSV file). The system will show the uploaded dataset as a data table and users can start configuring their visualizations by adding and connecting elements, as seen in **Figure 40**.

Figure 40. Example instantiation of the dashboard meta-model and the generated visualizations using MetaViz. Source: own elaboration.

MetaViz is available at https://metaviz.grial.eu/. A basic overview of the system's functionalities and a tutorial can be consulted through the following link: https://www.youtube.com/watch?v=iCFEFW0zL5I. The detailed research related to this work can be consulted in **Appendix AJ.** MetaViz - A graphical meta-model instantiator for generating information dashboards and visualizations [69].

4.4 Integrations in real-world scenarios

The proposed approach to automatically generate information dashboards has been subject to continuous validations to improve the meta-model and the functionalities of the generative pipeline. These validations have been carried out by integrating the generative pipeline into different contexts.

The preliminary version of the meta-model and generative pipeline was tested in the employment and employability domain. After this integration, the meta-model was improved and the newer version, along with the ecosystem approach conceptualized in [64], were integrated in different data-driven health platforms. Finally, the learning dimension of the meta-model and the graphical instantiator —MetaViz— has also been tested in the educational context.

4.4.1. Employment and employability

The concept of employability has increasingly gained relevance over the last decades. There is a reason: knowing which factors increase the possibility to obtain a job or to perform better in current job positions could be decisive to improve individual and collective life quality.

However, generating knowledge in such a field is not a trivial task. There could be several variables involved in the research of students' employment and employability, so it is necessary to collect significant data volumes to be able to reach valuable insights. In addition to data collection, performing data analysis [186] is required to be able to reach useful insights.

Visual analytics tools like information dashboards, are crucial in these data-intensive contexts, as they can assist knowledge generation. But information dashboards not only need to be useful concerning functionality. They should be customizable, flexible, and scalable regarding its data sources and structures, making the development and maintenance of information dashboards even more complicated. Of course, these issues could be addressed by developing particular dashboards for each involved user to achieve every specific goal, but clearly, this solution would be time-consuming and require many resources during the development and maintenance phases.

In this context, the preliminary version of the meta-model and software product line were applied to tackle the introduced issues [49, 70]. Specifically, this approach was applied in The Spanish Observatory for University Employment and Employability [18, 24]. This network of researchers and technicians conduct studies about these fields in the academic context [20, 23, 24], through a data-driven approach to collect, analyze, visualize, and disseminate employment and employability data of graduates from Spanish universities. The research related to the data-driven ecosystem of the Observatory is published can be consulted in **Appendix A**. Scaffolding the OEEU's data-driven ecosystem to analyze the employability of Spanish graduates, **Appendix B**. Improving the OEEU's data-driven technological ecosystem's interoperability with GraphQL, **Appendix C**. How different versions of layout and complexity of web forms affect users after they start it? A pilot experience, and **Appendix D**. Enabling adaptability in web forms based on user characteristics detection through A/B testing and machine learning [18, 19, 21, 41].

For the Observatory's dashboards, three main configurable visual components (features) were defined: a scatter diagram, a chord diagram, and a heat map (**Figure 41**). These visualizations address the requirements of the Observatory's data but can be reused for other data domains.

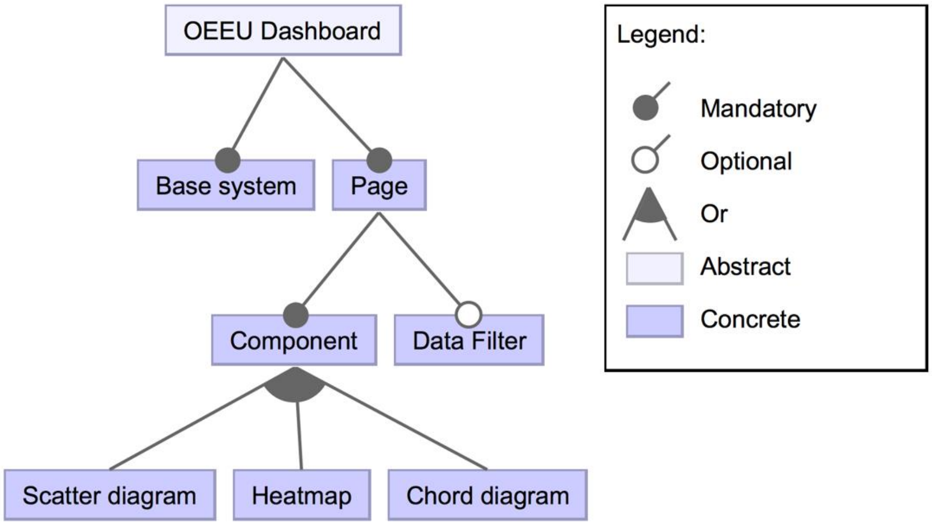


Figure 41. Feature model for the OEEU dashboards product line. Source: own elaboration, published in [70]

A detailed view of the scatter diagram feature can be seen in **Figure 42**. It has a set of subsequent features, either mandatory, optional, or alternative.

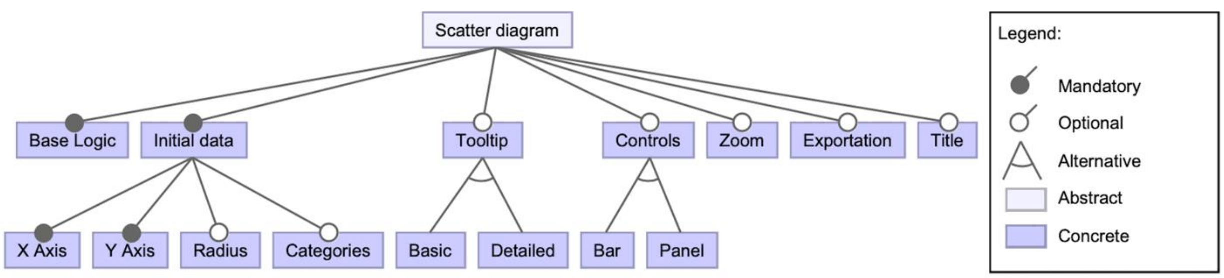


Figure 42. Feature model for the scatter chart component. Source: own elaboration, published in [70]

```xml
<xs:complexType name="LayoutType">
    <xs:choice>
        <xs:element name="RowGroup" type="LayoutType"/>
        <xs:element name="ColumnGroup" type="LayoutType"/>
        <xs:element name="Row" type="LayoutType" maxOccurs="unbounded"/>
        <xs:element name="Column" type="LayoutType" maxOccurs="unbounded"/>
        <xs:element name="Component">
            <xs:complexType>
                <xs:simpleContent>
                    <xs:extension base="xs:anySimpleType">
                        <xs:attribute name="ref" type="xs:string"/>
                    </xs:extension>
                </xs:simpleContent>
            </xs:complexType>
        </xs:element>
        <xs:element name="DataFilter">
            <xs:complexType>
                <xs:simpleContent>
                    <xs:extension base="xs:anySimpleType">
                        <xs:attribute name="ref" type="xs:string"/>
                    </xs:extension>
                </xs:simpleContent>
            </xs:complexType>
        </xs:element>
    </xs:choice>
    <xs:attribute name="width" type="xs:string" use="optional"/>
    <xs:attribute name="height" type="xs:string" use="optional"/>
</xs:complexType>
```

Figure 43. Excerpt of the specification of a dashboard using the DSL. Source: own elaboration, published in [70]

A DSL was designed to accomplish the connection between the meta-model and source code. The DSL was based on the identified domain's features of the SPL, by structuring them with XML technology [187] and by validating the model restrictions with an XML schema [188]. **Figure 43** shows an excerpt of the DSL regarding how the layout of the dashboard is specified in terms of rows, columns, and components.

The results show the success of the implementation of the SPL in this context. The previous version of the OEEU's system provided static dashboards with the same indicators for each user. With this approach, dashboards can be customized by instantiating the desired components and functionalities. **Figure 44** displays an example of a generated dashboard with the three available components in the SPL.

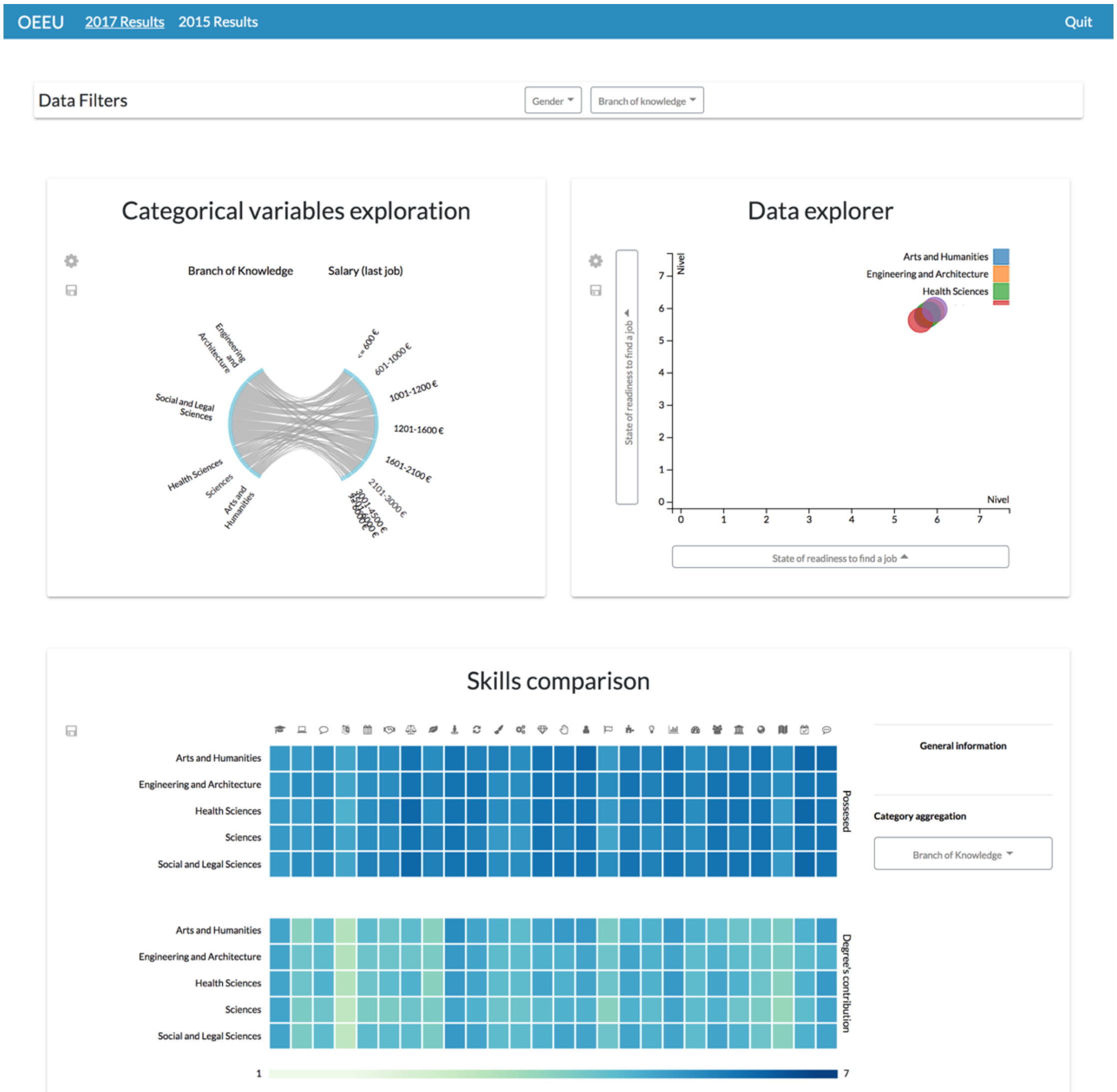


Figure 44. Example dashboard generated through the SPL. Source: own elaboration, published in [70].

The research associated with this work is published and can be consulted in **Appendix E**. Domain engineering for generating dashboards to analyze employment and employability in the academic context, and **Appendix L**. Taking advantage of the software product line paradigm to generate customized user interfaces for decision-making processes: a case study on university employability [49, 70].

4.4.2. Health

Data analysis mechanisms and interfaces are crucial in data-intensive domains. Scenarios in which data is being continuously generated represent a challenge in terms of analysis, storage, and exploitation.

One of these complex scenarios is the medical context. Not only several data sources can be involved, but also different data structures. This data heterogeneity is a challenge both for its management and exploitation.

In this subsection, different research works related to health platforms focused on providing interfaces to assist both decision-making processes and AI applications in this domain are presented.

CARTIER-IA

One of the main challenges of applying AI algorithms in real medical scenarios relies on the unification and accessibility of the generated data, which can involve structured data as well as imaging data. For this reason, information systems are required to gather, clean, organize and structure data to apply AI algorithms in a friendly, secure, and anonymized manner.

This section presents a platform for the management of structured data and imaging resources in the medical context with advanced features such as their visualization, edition and application of AI on the stored resources [36, 40].

The CARTIER-IA platform can be seen as a technological ecosystem that supports all data-management related tasks (including structured data and medical imaging collection) and also enables both healthcare professionals and data scientists to apply AI models to the stored images.

Deep learning and machine learning models can be stored by AI developers through Python scripts. Using this approach, scripts can be executed through the web interface by any user interested in analyzing the image, with the goal of providing the benefits of these scripts without requiring python-programming skills nor advanced knowledge regarding Artificial Intelligence algorithms.

This platform relies on three main modules: structured data management, DICOM [189] images management, and AI support. The architecture of CARTIER-IA is also based on customization, as the schema of the projects created in this platform can completely vary among them. However, the flexibility related to the system's structure allows the management and unification of different data formats.

This design has made possible the application of AI algorithms to imaging data without requiring programming skills. AI algorithms are available for any user to execute and retrieve its results. **Figure 45** shows an execution example of a segmentation AI algorithm for ultrasound images in CARTIER-IA.

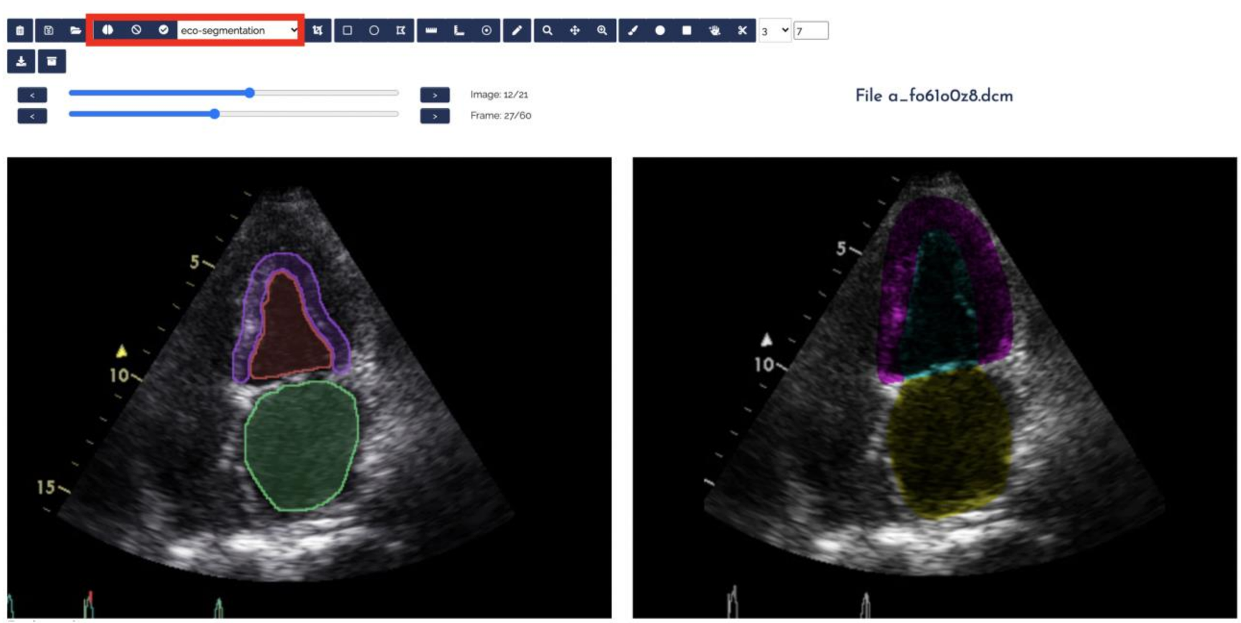


Figure 45. Screenshot of a manual segmentation (left) and the AI algorithm output (right). Source: own elaboration, published in [36].

This architecture also allows the integration of external data sources (such as REDCap [190]) and modules, such as the generative pipeline developed in this thesis. As of this date, the integration of automatically generated dashboards in the

CARTIER-IA platform to support the exploration of different structured data schemas is in progress.

The detailed specification of the CARTIER-IA platform and its associated evaluations can be consulted in **Appendix AA**. A platform for management and visualization of medical data and medical imaging, and **Appendix AC**. Application of Artificial Intelligence Algorithms Within the Medical Context for Non-Specialized Users: the CARTIER-IA Platform [36, 40].

KoopaML

ML has become a powerful approach to tackle complex tasks that involve analyzing significant amounts of data. Data-intensive contexts, such as the health domain, benefit directly from applying ML algorithms to their data, supporting tasks such as identifying patterns, clustering, classification, predictions, etc., that could become time- and resource-consuming if approached through manual paradigms. The application of ML to health data has proven its usefulness in specific challenges like diagnoses, disease detection, segmentation, assessment of organ functions, etc. [191-193].

However, applying ML approaches is not straightforward. More specifically, using them in sensitive domains (such as health) could be hazardous if practitioners do not fully understand the results derived from the models.

A graphical platform (KoopaML) has been developed to alleviate these issues and to offer intuitive and educational interfaces to build and run ML pipelines to tackle these challenges. The primary target audience of this platform is non-expert users interested in learning and applying ML models to their domain data. A user-centered design approach was followed to capture relevant requirements and necessities from potential user profiles involved in this context.

The architecture of KoopaML is based on different modules connected by information flows. One of the primary purposes of this design is to provide flexible pipelines with reusable components.

In this regard, a domain engineering approach was employed following the previously described requirements elicitation process with potential users and literature reviews. Following this approach, four general functional blocks were proposed: User, Heuristics, Pipelines and Task management modules. These blocks will interact and collaborate among them to provide support for the implementation of flexible ML workflows

The user management module provides the services related to authentication, sessions, and roles. The heuristics management module allows IA experts to modify the heuristics through a graphic interface. The pipelines management module provides a workspace to create ML pipelines using visual elements. Finally, the tasks management module defines the operations related to each ML pipeline potential stage.

Following the software product line architecture paradigm [87, 194-197], the ML pipelines were divided into fine-grained tasks with well-defined inputs and outputs. Through this approach, the tasks management module acts as a repository of loosely coupled ML-related tasks, in which algorithms and operations can be added and modified without impacting the features of the remaining modules/tasks.

Users can instantiate nodes from each category and connect them through their inputs and outputs. These inputs and outputs are also categorized to ensure that information flows are compatible among the instantiated nodes. **Figure 46** shows an example pipeline created with KoopaML.

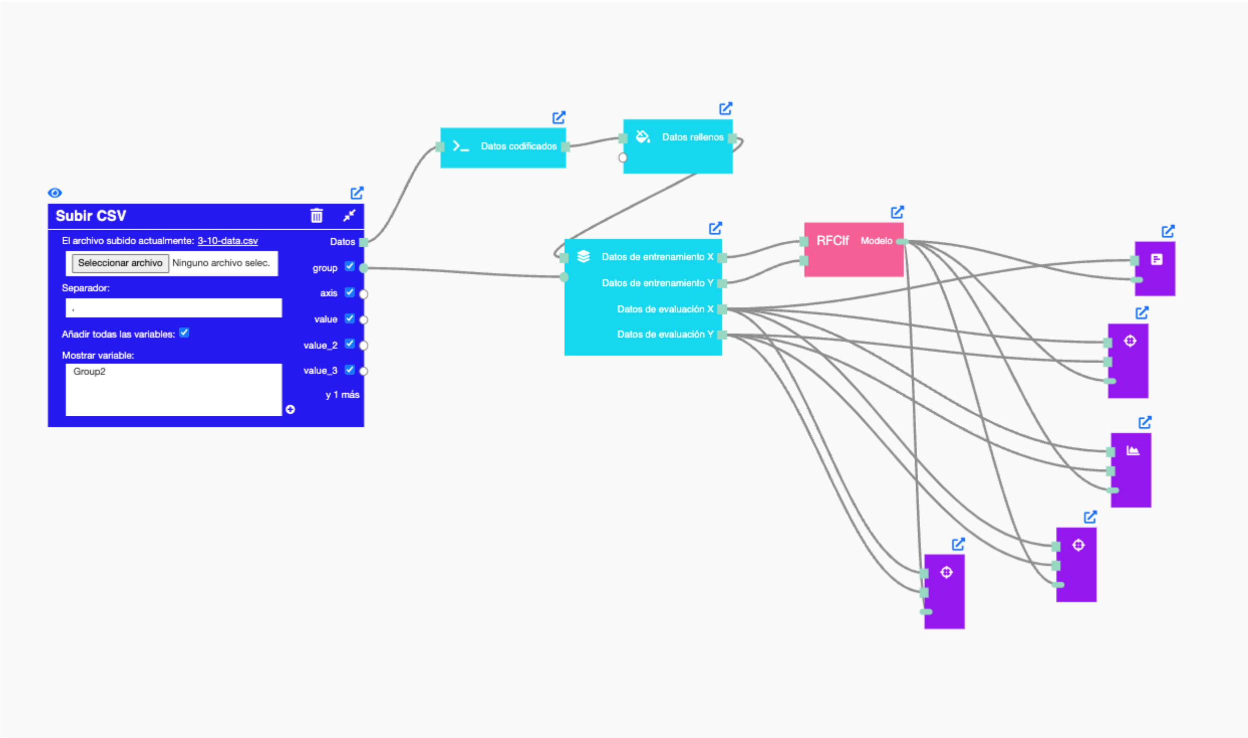


Figure 46. Example pipeline to train a Random Forest classifier. Source: own elaboration.

To support Exploratory Data Analysis (EDA) [198], which is a crucial part of ML applications, the generative dashboard pipeline was integrated into KoopaML using the approach previously detailed in [64].

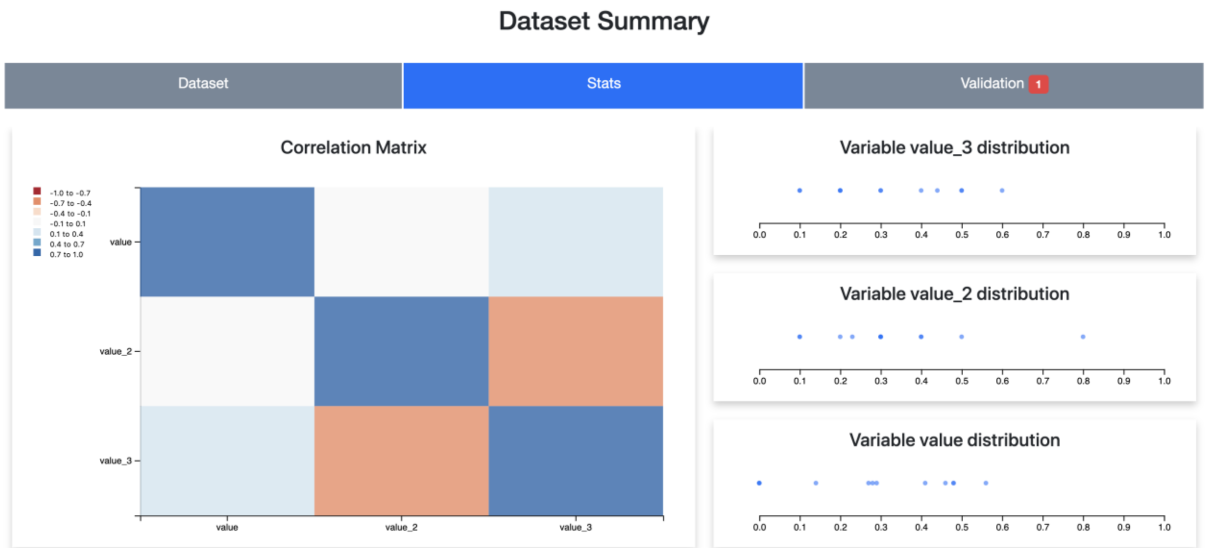


Figure 47. Generated dashboard to explore training datasets. Source: own elaboration, published in [35].

Figure 48. Generated dashboard to explore training results. Source: own elaboration.

This integration allowed the dynamic generation of dataset summaries depending on the input data (**Figure 47**), as well as reports of the results derived from the training process adapted to the metrics required by the user (**Figure 48**).

The complete description of this platform, including its preliminary usability evaluations can be found in **Appendix AD**. User-Centered Design Approach for a Machine Learning Platform for Medical Purposes, **Appendix AE**. Bringing machine learning closer to non-experts: proposal of a user-friendly machine learning tool in the healthcare domain, and **Appendix AI**. KoopaML: A graphical platform for building machine learning pipelines adapted to health professionals [35, 38, 39].

Visual SALMANTICOR

The SALMANTICOR study [199] is a population-based cross-sectional descriptive study on the prevalence of structural heart disease and its risk factors. A total of 2,400 individuals, stratified by age, sex and place of residence (rural and urban), were recruited in the province of Salamanca (Spain). The study took place in the period between 2015 and 2018.

The SALMANTICOR study was conceived to obtain data concerning the prevalence and incidence of structural heart disease and therefore to be able to find relevant patterns with which to build appropriate public policies and/or public health campaigns.

However, the number of total variables collected (more than 300) makes the analysis of these data complex for non-experienced users. It is therefore important to provide practitioners, who are the ones with the medical knowledge, with a suitable platform to visualize these data in a simple way and thus confirm or disprove their hypotheses, as well as to find interesting patterns of analysis. In addition, these premises can help experienced users in further analysis.

In this context, a visual analytics platform was developed. The platform is not only focused on providing basic visualizations to summarize the results, but also on offering a good user experience to reach insights regarding the study [37]. For these reasons, the architecture is composed of modules that oversee the retrieval, visualization, and analysis of each study section. Dividing the architecture into individual but related modules enables flexibility to extend the platform with more functionalities, analyses, and visualizations.

The platform is developed as a web service. The front-end provides a usable interface and data visualizations to navigate the data, and the backend offers data computation, storage, and retrieval functionalities through API calls. Moreover, data visualizations can also be generated on demand through drag and drop interactions involving the study variables by using the generative pipeline.

Due to the vast number of variables involved in the study, it is necessary to provide end-users with a helpful tool to explore the different facets of data instead of static summary visualizations. However, displaying variables in a map without further details (such as sociodemographic factors or sampling details) could lead users to a superficial view of the results.

For these reasons, the exploration view provides controls to select specific variables to be inspected and a Sankey visualization that includes sociodemographic variables to overview the participants' characteristics (**Figure 49**).

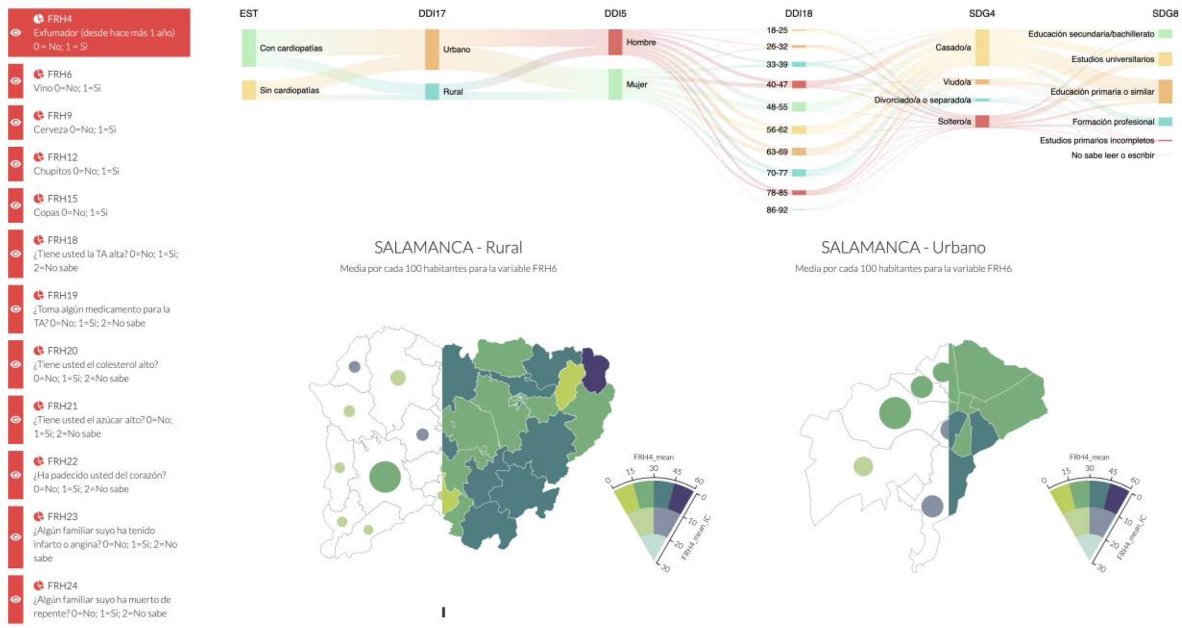


Figure 49. Risk factors exploration view. Source: own elaboration, published in [37].

The platform also provides an advanced workspace in which the summaries of every variable are displayed; numeric variables can be inspected through histograms, while categorical variables can be examined through bar charts with the count of every unique category involved (**Figure 50**).

Besides this individual summary of each variable, this feature allows users to drag and drop variables into a workspace and combine them to obtain different information visualizations. Visualizations are automatically generated through the generation service developed during this thesis.

Figure 50. Advanced exploration of variables. The leftmost section provides a summary of each variable and the possibility of dragging them into the middle section to craft a visualization. The rightmost section displays the generated visualizations. Source: own elaboration, published in [37].

This work is published and can be consulted in **Appendix AB**. A platform to support the visual analysis of the SALMANTICOR study outcomes: conveying cardiological data to lay users [37].

The applications of the meta-model in the health domain have set the foundations and been part of the **AVisSA project** (ref. PID2020-118345RB-I00). The main objective of AVisSA is focused on generating information dashboards automatically and adapt them to data analysis and knowledge management needs in heterogeneous contexts such as the health sector, and, consequently, to improve these processes within the health system, impacting decision-making processes. The description of this project can be consulted in **Appendix AF**. Fostering Decision-Making Processes in Health Ecosystems through Visual Analytics and Machine Learning [34].

4.4.3. Education

This subsection describes a work-in-progress to test the role of the dashboard meta-model and the graphical instantiator (MetaViz) as learning resources.

Several platforms have emerged to smooth the process of designing and implementing data visualizations.

Commercial visualization systems like Tableau (https://www.tableau.com/), Microsoft Excel (https://www.microsoft.com/microsoft-365/excel), Power BI (https://powerbi.microsoft.com/), etc., provide graphical interfaces that allow users with no experience in programming to create data visualizations and even assist them in the design process to choose the best encodings. However, it is crucial to understand and account for every element involved in data visualizations to deliver effective and non-confusing or misleading displays of information [8-10, 144].

A pilot study to measure the understandability of the elements involved in the design of data visualizations and information dashboards was developed to explore the potential educational role of the meta-model.

For this matter, a procedure and a questionnaire that aims at measuring two cognitive dimensions following Bloom's taxonomy of educational objectives [200-202] were designed:

- **Remember** – Identification of the elements that compose a data visualization at first sight, i.e., to test if the subject can recognize and recall basic elements of data visualizations.
- **Understand** – Understandability of the data visualization design process, i.e., to test if the student is aware of the dimensions and elements involved in the display.

The study procedure consists of two similar parts. The first part involves a widely used data visualization tool, in this case, Tableau. Users are asked to download a test dataset and to create a scatter plot that shows data values from two numerical variables and one categorical variable with Tableau. When finished, users need to close Tableau and answer the following questions regarding the scatter plot they just created:

1. Which variable was represented in the X-axis?
2. What was the maximum value of the Y-axis scale?
3. How many visual encodings were employed in your visualization?
4. How many scales were involved in your visualization?

This set of questions are focused on testing if users were aware of the design process of their own data visualizations and if they remember basic features of their charts. Textual and graphical indications regarding the meaning of the data visualization terms involved in the questions are provided to avoid confusion regarding these concepts. **Figure 51** and **Figure 52** shows explanations regarding the meaning of "scale" and "visual channel/encoding", respectively.

The remaining questions of this part test the ability of users to identify the elements that compose an already implemented data visualization. In this regard, different screenshots of data visualizations created in Tableau are displayed, following the next questions:

5. How many variables are involved in the (screenshot's) visualization?
6. How many scales are involved in the (screenshot's) visualization?
7. How many visual channels or encodings are involved in the (screenshot's) visualization?

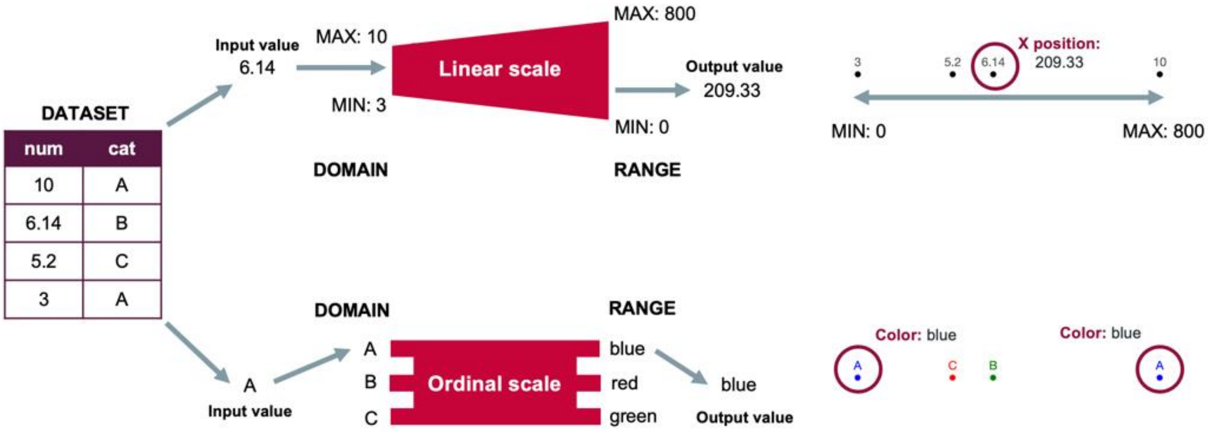


Figure 51. Indications regarding the scale concept in the data visualization domain. Source: own elaboration.

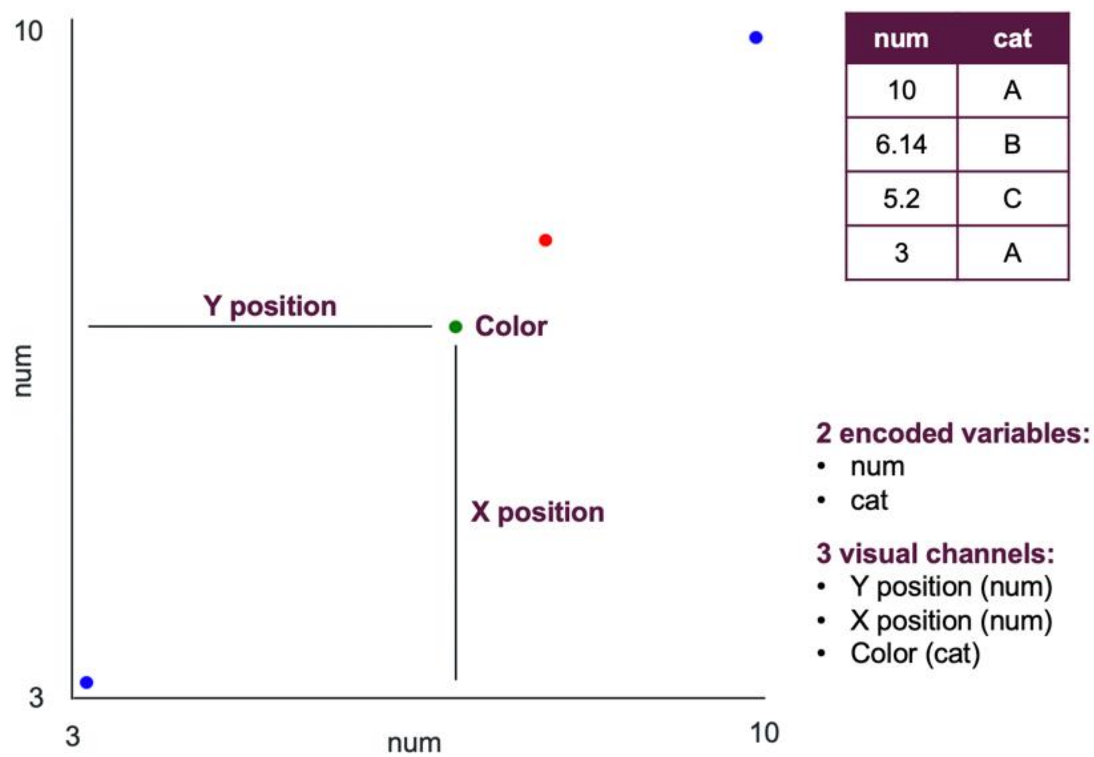


Figure 52. Indications regarding the channels/encodings concept in the data visualization domain. Source: own elaboration.

Figure 53 (a) shows an example of a question from this block. Once finished, the user is asked to create another scatter plot with MetaViz with the same dataset. A quick tutorial is also provided due to the novelty of this system. When the task has been completed, the previous questionnaire is presented in the context of the MetaViz system, that is, questions 1 to 4 referring to the scatter plot created by the user in MetaViz and questions 5 to 7 with screenshots of data visualizations created in this same system, as **Figure 53** (b) presents.

The responses to the questionnaire will be analyzed to get insights about the learning experience derived from the use of MetaViz, and, in turn, from the meta-model.

This work can be consulted in **Appendix AH**. A proposal to measure the understanding of data visualization elements in visual analytics applications [71].

Figure 53. Questions regarding visualizations created in Tableau (a) and MetaViz (b). Source: own elaboration.

4.5 Conclusions

This chapter presented the results derived from the application of the meta-model within different contexts, processes, and dimensions. The development of a dashboard meta-model has opened several research paths, both at theoretical and practical levels.

The versatility of this artifact and the associated generative pipeline has been validated through different case studies. First, a content expert validation of the meta-model was conducted to seek for conceptual issues. The results proved that the identified entities and relationships were relevant, coherent, and understandable. These features are crucial, especially in this context. Due to the inherent complexity of the data visualization and dashboards domain, it is necessary to convey the knowledge and entities related to this context in an understandable manner.

Conducting case studies in different domains has also improved the meta-model and the pipeline, as new entities and relationships arose from the integrations. The evolution of these resources can be observed through the presented studies, starting from a very basic, coarse-grained version of the meta-model integrated in the employment and employability domain [49, 70] and ending in a complete framework that has served to train ML models [68], conceptualize information dashboards at high level [61, 62, 65-67], support decision-making processes in health platforms [36-40], and even drive the architecture and interface of a whole platform (MetaViz).

All these studies have been part of the defined action-research cycles, which enabled the identification of new relationships, concepts, and nuances in the data visualization domain, resulting in a powerful and versatile meta-model to define information dashboards and visualizations.

The results described in this section have several theoretical and practical implications. The following chapter will discuss the outcomes of the dimensions explored during the development of this thesis.

5 Discussion

Several results and applications have been obtained during the development of the present thesis. Figure 54 summarizes the main outcomes of this work and their related publications.

First, the **systematic literature review results** were crucial to select the most suitable methodology to tackle the automatization of tailored information dashboards and data visualizations [47, 48]. These results indicated that several methodologies can be applied in the data visualization and information dashboards domains.

After the analysis, the selected methodologies to carry out the generative dashboard system were the **MDD and SPL paradigms**. On the one hand, the SLR contained 8 applications of these methodologies in this domain, proving their feasibility. On the other hand, the related benefits derived from applying these methodologies in other domains —flexibility, evolving capabilities, traceability of requirements, integration of external configuration algorithms— aligns with the requirements of the target domain.

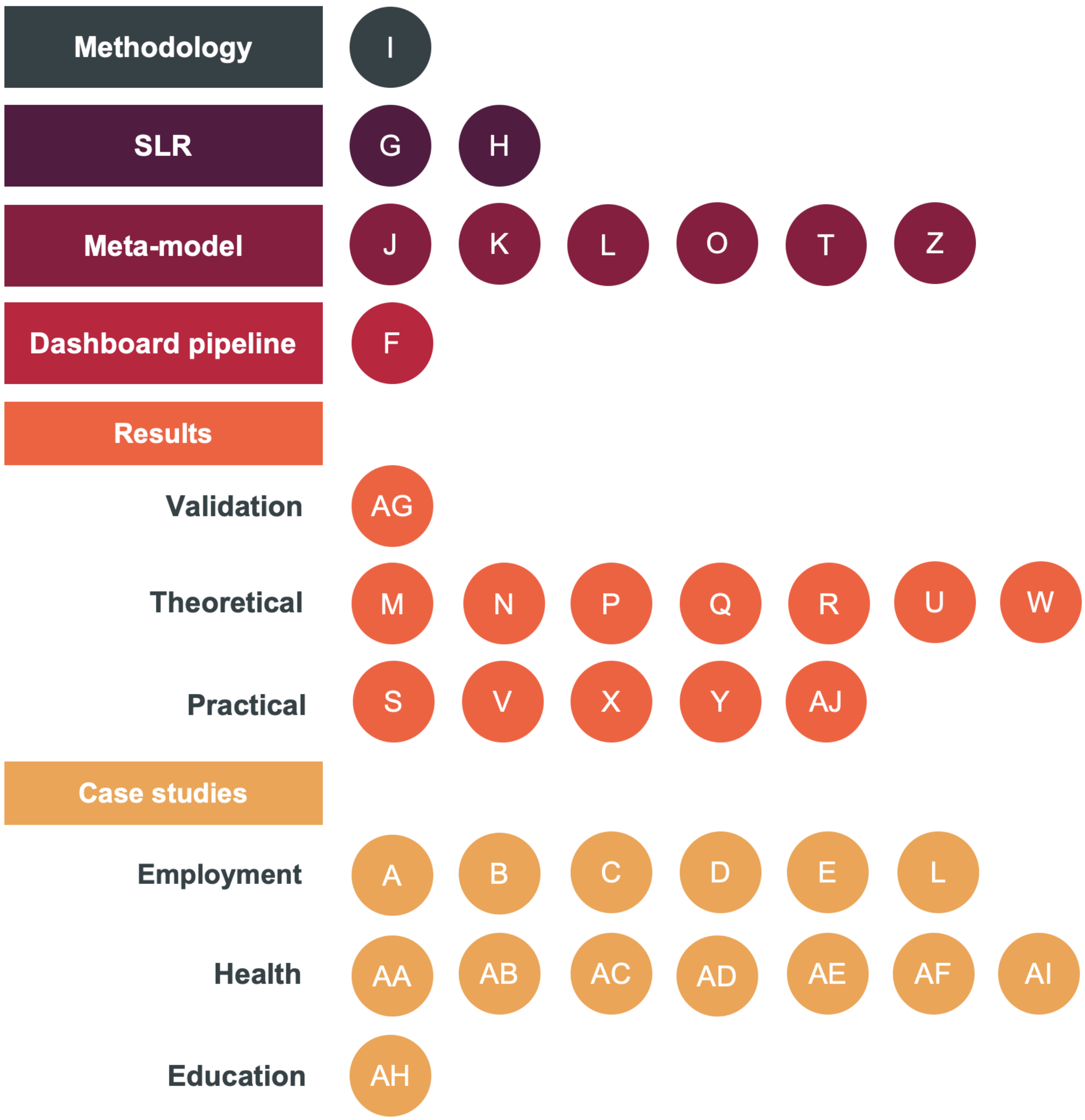


Figure 54. Summary of the outcomes and associated publications (with their related appendix number) of the present thesis.

The main challenge related to applying a combination of MDD and SPL were the **fine-grained nature of information dashboards and data visualization features**. These tools can be configured in several ways, from the input dataset to the visual properties of the displayed shapes, interactivity, layouts, etc.

The preliminary application of this approach in the context of employment and employability [49, 70] proved the **feasibility of the generative pipeline** in a real-world system. However, the first version of the meta-model and, in turn, of the SPL, only accounted for coarse-grained features in this domain, and only three types of data visualizations were available for instantiation.

In this context, a domain engineering approach [85, 86] along with an example-driven methodology [139] were applied to extract the main and common features among the elements of data visualizations and information dashboards and to arrange them into a **meta-model**.

Although other dashboard meta-models exist in the literature [101, 107, 203], they do not account for fine-grained features, which are crucial in this domain, as a slight modification in the design of a dashboard or data visualization could lead to significantly different insights [11, 184, 204].

The development of the dashboard meta-model has not been a linear process. Several improvements and modifications have been made before obtaining the current version of the meta-model. These modifications are the result of the studies that have branched out from the application of domain engineering in this context.

The first version of the meta-model lacked detail both in terms of the visual components and of the end-user. Including the user as an extremely significant element within the dashboards and data visualization domain is crucial. The development processes of a visualizations and dashboards start with the user (requirement elicitation) and end with the user (product refining) [93, 144, 205], so not only the technical features of a dashboard should be taken into account when meta-modeling these tools, as these features arises from the users' requirements and are influenced by them [92].

In fact, the results of the instantiation in [61] and [66] proved that **user information goals** were necessary to design dashboards that support different roles and necessities.

The same applies to the work derived from the integration of the generative pipeline in the health platforms described in [36-40]. The heterogeneous sources of data and variety of roles involved in the health domain ask for **highly customizable interfaces**. In this sense, providing dashboards as a service [64] enabled the possibility of generating dashboards transparently and build-on-demand at different stages of the data analysis of structured data, DICOM images, and ML results.

In addition, it was necessary to include entities and relationships related to data into the meta-model. The experiment carried out in [68] allowed to tag data visualizations using expert knowledge about the data context. However, in the second experiment [54] in which data was randomly generated (and therefore did not belong to any specific context), it was nearly impossible to decide if a visualization was potentially misleading because more **context** was required to perform the tagging. Thanks to this experience, a lack of the notion of data context and domain variables was identified in the meta-model and subsequently corrected [54].

The final version of the meta-model was validated through **expert judgement**. The obtained feedback provided more information to keep improving the meta-model, but as the results show, the meta-model, although complex (which is inherent to the domain), was rated as a **highly relevant, coherent, and clear resource**.

Regarding the technical details, the dashboard generator has been tested with different languages: VegaLite [138] in [63], React (https://es.reactjs.org/) in [53], and, finally, D3.js [166] in [37-39, 64-66, 68] and in the development of MetaViz.

The decision of using D3.js was driven by the **expressiveness** provided by this language, which aligns with the fine-grain structure of the meta-model. However, using code templates to materialize the SPL features into code [56] has enabled the possibility to develop the SPL's core assets through different technologies.

The meta-model was also an essential resource for exploring the application of AI to the data visualization domain [68]. First, following the model-driven development and the software product line engineering, it was possible to automatically generate the visualizations that were later labeled to **build a training**

set. Furthermore, second, the meta-model provided the fine-grained features and relationships to structure the training dataset schema, which was crucial before applying any ML algorithm.

The precision and accuracy metrics show that the resulting ML model learnt from the **implicit expertise and heuristics** that were used to manually label the training set. The criteria followed to train the models could be seen as obvious or very basic (such as not exaggerating the scale values or using certain encodings) [68], but this study is focused on developing an automatic detector of data visualization potential design flaws, so that novices or not skilled users are aware of latent misleading configurations they are unconsciously introducing in their designs.

Moreover, the meta-model provided the **backbone for the development of a complete platform** for generating information dashboards through a graphical interface [69]. The MetaViz platform (https://metaviz.grial.eu/) represents the **unification of all the knowledge** derived from the research performed during this thesis. Both the system architecture and the user interface are driven by the MDA architecture layers and by the entities and relationships of the meta-model. The implementation of MetaViz has served to materialize the meta-model into a **functional resource** that can be interactively explored, instantiated, and transparently transformed into real code through the generative pipeline.

Finally, the number of applications of the meta-model in **different dimensions** (theoretical and practical) and **domains** (employment and employability, health, and education) is also a result itself. Every outcome associated with this thesis is driven by the dashboard meta-model, which also proves its versatility and flexibility when it comes to conceptualize, generate, and capture knowledge related to dashboards and data visualizations.

6 Conclusions

This research explored different dimensions of the **automatization** of the development of **tailored** information dashboards and data visualizations.

On the one hand, **automatization** is linked with several benefits, mostly related to decreases in development times and, thus, decreases in time-to-market. In fact, decreasing time-to-market in the domain of data visualizations and dashboards is crucial nowadays since data drives several processes and activities in practically any context [206]. For this reason, having tools ready to exploit data promptly is truly valuable, as they can give users an advantage to reach insights and understand the information contained in these data more effectively [8-12, 94, 144, 207-209].

On the other hand, **tailored** products have proven to deliver better experiences, especially in fields related to Human-Computer Interaction. Adapting user interfaces [210] to specific domain or users' characteristics can improve ease of use, satisfaction, efficiency, and accessibility, among other benefits [211, 212]. However, tailoring interfaces at low-level using fine-grained components is a challenge both at theoretical and practical levels.

In this context, the present thesis hypothesizes the possibility of improving the benefits related to functional and non-functional features of **tailored** user interfaces for supporting decision-making processes through **automatization**.

To explore this hypothesis, the primary goal of this research was focused on designing and implementing a **generative framework** for the automatic and **systematic development of tailored information dashboards**.

Three main outcomes were obtained from the research related to this Ph.D. thesis. First, the **systematic literature review** and **mapping** of tailoring capabilities in the domain of information dashboards and data visualizations [47, 48]. This review enabled us to take an informed decision regarding the selection of methodologies to address the development of the generative framework.

Among the different kinds of tailoring approaches, MDD and SPL paradigms were selected due to their feasibility in this domain, but also because of their capabilities to support fine-grained features –useful to tackle the functional features of the generated dashboards– and their benefits related to the improvement of the development processes and knowledge management –useful to tackle the non-functional features of the generation processes–.

This research has been framed within the Action-Research methodology, which answers to each posed objective through several cycles driven by the SLR results. This approach has made it possible to determine shortcomings in the MDD and SPL proposals for developing the additional two outcomes of this thesis: a **dashboard meta-model** and the subsequent **generative pipeline of information dashboards**.

While meta-modeling and MDD methodologies provide a powerful approach to characterize, identify, and model the target domain, the SPL paradigm allows the materialization of these abstract features into real-world, highly customizable products.

As the results derived from this thesis have shown, the meta-model has been subject to several modifications, additions, and improvements driven by the outcomes of case studies in a variety of domains [50-52, 54, 70, 185]. These iterations have been crucial to obtain the current version of the meta-model, which captures both functional and non-functional features of information dashboards and data visualizations.

Using the meta-model as an input, it has been possible to develop a generative pipeline [37-39, 58-68, 70] that transforms high-level specifications of dashboards into source code, providing an improvement in the development process of these tools in terms of traceability of the requirements, time-to-market, and customization.

In this sense, it is clear to conclude that the MDD and SPL approaches are beneficial to improve the development processes of tailored information dashboards and data visualizations.

First, because they support the definition of fine-grained features, crucial in this context to address the functional features of these tools [11, 184, 204]. Second, because the successful integration of the generative pipeline in different domains [37-39, 49, 70] has proved its benefits decreasing the development time of these tools, as well as improving customization, and flexibility.

Another clear benefit of automatizing the development of tailored information dashboards through MDD and SPL has been the knowledge management of the domain expertise. Capturing the entities and relationships of this domain and arranging them into a meta-model has provided a valuable resource to drive data visualization-related research and even to create a platform (https://metaviz.grial.eu/) with the goal of offering a learning experience while designing and generating data visualizations.

Following this idea, another conclusion has been reached through the research line that started by exploring the integration of AI mechanisms in the generative pipeline [68]. While this integration has been highly beneficial to capture expert knowledge automatically and further research will explore automatic configurations for non-expert users, it has also a downside. As it is well known, training AI models can lead to biased results if there is a lack of domain expertise or if input data are not interpreted correctly [171, 172].

This work put the focus on the automatic generation of information dashboards. But generative pipelines can be dangerous if not controlled. As any AI-related application, it needs human knowledge. It needs human supervision. Software systems can assist the design of data visualizations, but the honest endeavor in this context is to train critical thinking through these tools and their design process. How can data be lost (purposely or not) while designing a means to convey them? It's paradoxical.

Knowing that data visualization can be affected by biases [13] it is important for the user to be aware of what they are doing and take action to convey or generate knowledge [213-215].

In this sense, this thesis has also served not only to analyze the automatization of the design and implementation of these tools, but to explore how to raise awareness on good practices while developing them.

The combination of the results and the previous conclusions have made it possible to affirm that the main objective and derived sub-objectives of this thesis have been achieved, and following its outcomes, that **the hypothesis posed at the beginning of this research project is valid**.

6.1 Future work

The development of the dashboard meta-model and its related generative pipeline has unlocked several research opportunities related to different areas.

First, the meta-model has proven to be a useful artifact to drive research related to **data visualization**. Several concepts captured in the meta-model can be further explored, specifically those related to the user characteristics and goals. Although modeling and detecting user bias and beliefs is a challenge, the benefits derived from retrieving this information would be highly valuable to fight fake news, polarization, and deceptive data visualizations.

In the area of **Software Engineering**, the meta-model can be improved with more specific OCL rules [216] that capture good practices and guidelines related to the data visualization realm. These enhancements can set the foundations of a rule-based recommendation system or knowledge base for data visualization design.

On the other hand, in the **Human-Computer Interaction** area, a future research line is to continue the user validation in terms of performance, usability, or satisfaction, among other metrics, of the generated dashboards, as well as the graphical instantiator platform.

The ideas presented in this thesis can also be applied in **educational** contexts, as the results derived from Section 4.4.3 hinted. The dashboard meta-model captures domain knowledge, providing a learning resource to teach the basic elements of data visualization. Further work could involve measuring the performance of the graphical instantiator in terms of its capability to provide a learning experience to their users.

Finally, there are also research opportunities in the **AI** field. The notion of identifying misleading visualization by training AI algorithms through the entities captured in the meta-model can be further explored to improve the model, and even develop an automated detector of deceptive data visualizations. Moreover, AI models could also be applied to assist the instantiation process through recommended configurations given the context of the dashboard or visualization and the user characteristics.

6.2 Ph.D. thesis' outcomes

Throughout the development of this doctoral thesis, a series of scientific publications have been carried out to validate the proposal. The publication process in different media has allowed obtaining feedback from experts in the field. Specifically, 12 articles have been published in indexed journals, and 23 papers in international conferences, in addition to a book chapter.

- **JCR-SCIE indexed journals (Q1: 5; Q2: 6):**

1. J. Cruz-Benito, A. Vázquez-Ingelmo, J. C. Sánchez-Prieto, R. Therón, F. J. García-Peñalvo, and M. Martín-González, "Enabling Adaptability in Web Forms Based on User Characteristics Detection Through A/B Testing and Machine Learning," *IEEE Access*, vol. 6, pp. 2251-2265, 2018, doi: 10.1109/ACCESS.2017.2782678. ISSN: 2169-3536. (JCR SCIE – COMPUTER SCIENCE, INFORMATION SYSTEMS – Q1 (23 of 155 – percentile 85.48); ENGINEERING, ELECTRICAL & ELECTRONIC – Q1 (52 of 265 – percentile 80.64); TELECOMMUNICATIONS – Q1 (19 of 88 – percentile 78.98) – JIF 4.098); (JCR JCI – COMPUTER SCIENCE, INFORMATION SYSTEMS – Q1 (31 of 220 – percentile 86.14); TELECOMMUNICATIONS – Q1 (16 of 104 – percentile 85.10); ENGINEERING, ELECTRICAL & ELECTRONIC – Q1 (51 of 312 – percentile 83.81) – JCI 1.40); (SJR 0.609 – COMPUTER SCIENCE (MISCELLANEOUS) – Q1; ENGINEERING (MISCELLANEOUS) – Q1; MATERIALS SCIENCE (MISCELLANEOUS) – Q2); (CiteScore 3.1; GENERAL ENGINEERING – Q1 (56 of 283 – percentile 80); GENERAL COMPUTER SCIENCE – Q2 (54 of 211 – percentile 74); GENERAL MATERIALS SCIENCE – Q2 (178 of 440 – percentile 59)) [41].

2. A. Vázquez-Ingelmo, F. J. García-Peñalvo, and R. Therón, "Information Dashboards and Tailoring Capabilities - A Systematic Literature Review," *IEEE Access,* vol. 7, pp. 109673-109688, 2019, doi: 10.1109/ACCESS.2019.2933472. ISSN: 2169-3536. (JCR SCIE – COMPUTER SCIENCE, INFORMATION SYSTEMS – Q1 (35 of 156 – percentile 77.88); ENGINEERING, ELECTRICAL & ELECTRONIC – Q1 (61 of 266 – percentile 77.26); TELECOMMUNICATIONS – Q2 (26 of 90 – percentile 71.67) – JIF 3.745); (JCR JCI – COMPUTER SCIENCE, INFORMATION SYSTEMS – Q1 (41 of 223 – percentile 81.84); TELECOMMUNICATIONS – Q1 (20 of 105 – percentile 81.43); ENGINEERING, ELECTRICAL & ELECTRONIC – Q1 (61 of 318 –

percentile 80.97) – JCI 1.22); (SJR 0.775 – COMPUTER SCIENCE (MISCELLANEOUS) – Q1; ENGINEERING (MISCELLANEOUS) – Q1; MATERIALS SCIENCE (MISCELLANEOUS) – Q2); (CiteScore 3.9; GENERAL ENGINEERING – Q1 (47 of 299 – percentile 84); GENERAL COMPUTER SCIENCE – Q1 (45 of 221 – percentile 79); GENERAL MATERIALS SCIENCE – Q2 (155 of 460 – percentile 66)) [48].

3. A. Vázquez-Ingelmo, F. J. García-Peñalvo, and R. Therón, "Taking advantage of the software product line paradigm to generate customized user interfaces for decision-making processes: a case study on university employability," *PeerJ Computer Science*, vol. 5, e203, 2019, doi: 10.7717/peerj-cs.203. ISSN: 2376-5992. (JCR SCIE – COMPUTER SCIENCE, ARTIFICIAL INTELLIGENCE – Q2 (50 of 137 – percentile 63.87); COMPUTER SCIENCE, INFORMATION SYSTEMS – Q2 (53 of 156 – percentile 66.35); COMPUTER SCIENCE, THEORY & METHODS – Q1 (24 of 108 – percentile 78.24) – JIF 3.091); (JCR JCI – COMPUTER SCIENCE, INFORMATION SYSTEMS – Q1 (45 of 223 – percentile 80.04); COMPUTER SCIENCE, THEORY & METHODS – Q1 (20 of 137 – percentile 85.77); COMPUTER SCIENCE, ARTIFICIAL INTELLIGENCE – Q1 (34 of 174 – percentile 80.75) – JCI 1.18); (SJR 1.601 – COMPUTER SCIENCE (MISCELLANEOUS) – Q1); (CiteScore 6.7; General Computer Science – Q1 (21 of 221 – percentile 90)) [70].

4. A. Vázquez-Ingelmo, F. J. García-Peñalvo, R. Therón, and M. Á. Conde, "Representing Data Visualization Goals and Tasks through Meta-Modeling to Tailor Information Dashboards," *Applied Sciences*, vol. 10, no. 7, 2306, 2020. [Online]. Available: https://www.mdpi.com/2076-3417/10/7/2306. ISSN: 2076-3417. (JCR SCIE – PHYSICS, APPLIED – Q2 (73 of 160 – percentile 54.69); CHEMESTRY, MULTIDISCIPLINARY – Q3 (101 of 178 – percentile 43.54); ENGINEERING, MULTIDISCIPLINARY – Q2 (38 of 90 – percentile 58.33); MATERIALS SCIENCE, MULTIDISCIPLINARY – Q3 (201 of 334 – percentile 39.97) –

JIF 2.679); (JCR JCI – PHYSICS, APPLIED – Q2 (63 of 171 – percentile 63.45); ENGINEERING, MULTIDISCIPLINARY – Q2 (55 of 170 – percentile 67.94); MATERIALS SCIENCE, MULTIDISCIPLINARY – Q2 (173 of 384 – percentile 55.08); CHEMESTRY, MULTIDISCIPLINARY – Q2 (84 of 219 – percentile 61.87) – JCI 0.61); (SJR 0.435 – COMPUTER SCIENCE APPLICATIONS – Q2; ENGINEERING (MISCELLANEOUS) – Q2; FLUID FLOW AND TRANSFER PROCESSES – Q2; INSTRUMENTATION – Q2; MATERIALS SCIENCE (MISCELLANEOUS) – Q2; PROCESS CHEMESTRY AND TECHNOLOGY – Q2); (CiteScore 3.0; FLUID FLOW AND TRANSFER PROCESSES – Q2 (34 of 83 – percentile 59); GENERAL ENGINEERING – Q2 (85 of 297 – percentile 71); INSTRUMENTATION – Q2 (49 of 128 percentile 62); COMPUTER SCIENCE APPLICATIONS – Q2 (300 of 693 – percentile 56); GENERAL MATERIALS SCIENCE – Q2 (222 of 455 – percentile 51); PROCESS CHEMESTRY AND TECHNOLOGY – Q2 (30 of 59 – percentile 50)) [52].

5. A. Vázquez-Ingelmo, A. García-Holgado, F. J. García-Peñalvo, and R. Therón, "A Meta-Model Integration for Supporting Knowledge Discovery in Specific Domains: A Case Study in Healthcare," *Sensors*, vol. 20, no. 15, 4072, 2020. [Online]. Available: https://www.mdpi.com/1424-8220/20/15/4072. ISSN: 1424-8220. (JCR SCIE – CHEMESTRY, ANALYTICAL – Q2 (26 of 83 – percentile 69.28); ENGINEERING, ELECTRICAL & ELECTRONIC – Q2 (82 of 273 – percentile 70.15); INSTRUMENTS & INSTRUMENTATION – Q1 (14 of 64 – percentile 78.91) – JIF 3.576); (JCR JCI – INSTRUMENTS & INSTRUMENTATION – Q1 (14 of 72 – percentile 81.25); ENGINEERING, ELECTRICAL & ELECTRONIC – Q2 (100 of 319 – percentile 68.81); CHEMISTRY, ANALYTICAL – Q2 (25 of 98 – percentile 75) – JCI 0.89); (SJR 0.636 – ANALYTICAL CHEMESTRY – Q2; ATOMIC AND MOLECULAR PHYSICS, AND OPTICS – Q2;

BIOCHEMESTRY – Q3; ELECTRICAL AND ELECTRONIC ENGINEERING – Q2; INFORMATION SYSTEMS – Q2; INSTRUMENTATION – Q2; MEDICINE (MISCELLANEOUS) – Q2); (CiteScore 5.8; INSTRUMENTATION – Q1 (13 of 128 – percentile 90); ATOMIC AND MOLECULAR PHYSICS, AND OPTICS – Q1 (42 of 192 – percentile 78); ELECTRICAL AND ELECTRONIC ENGINEERING – Q1 (135 of 693 – percentile 80); INFORMATION SYSTEMS – Q1 (69 of 329 – percentile 79); ANALYTICAL CHEMESTRY – Q1 (29 of 122 – percentile 76); BIOCHEMESTRY – Q2 (133 of 415 – percentile 67)) [60].

6. A. Vázquez-Ingelmo, F. J. García-Peñalvo, R. Therón, D. Amo Filvà, and D. Fonseca Escudero, "Connecting domain-specific features to source code: towards the automatization of dashboard generation," *Cluster Computing*, vol. 23, no. 3, pp. 1803-1816, 2020, doi: 10.1007/s10586-019-03012-1. ISSN: 1386-7857. (JCR SCIE – COMPUTER SCIENCE, INFORMATION SYSTEMS – Q4 (123 of 162 – percentile 24.38); COMPUTER SCIENCE, THEORY & METHODS – Q2 (50 of 110 – percentile 55.00) – JIF 1.809); (JCR JCI – COMPUTER SCIENCE, INFORMATION SYSTEMS – Q3 (112 of 223 – percentile 50.00); COMPUTER SCIENCE, THEORY & METHODS – Q2 (53 of 137 – percentile 61.68) – JCI 0.65); (SJR 0.335 – COMPUTER NETWORKS AND COMMUNICATIONS – Q3; SOFTWARE – Q3); (CiteScore 3.1; COMPUTER NETWORKS AND COMMUNICATIONS – Q2 (144 of 334 – percentile 57); SOFTWARE Q3 (213 of 389 – percentile 45)) [63].

7. A. Vázquez-Ingelmo, F. J. García-Peñalvo, and R. Therón, "Towards a Technological Ecosystem to Provide Information Dashboards as a Service: A Dynamic Proposal for Supplying Dashboards Adapted to Specific Scenarios," *Applied Sciences*, vol. 11, no. 7, 3249, 2021, doi: 10.3390/app11073249. ISSN: 2076-3417. (JCR SCIE – PHYSICS, APPLIED – Q2 (73 of 160 – percentile 54.69); CHEMESTRY, MULTIDISCIPLINARY – Q3 (101 of 178 – percentile 43.54);

ENGINEERING, MULTIDISCIPLINARY – Q2 (38 of 91 – percentile 58.79); MATERIALS SCIENCE, MULTIDISCIPLINARY – Q3 (201 of 333 – percentile 39.79) – JIF 2.679); (JCR JCI – PHYSICS, APPLIED – Q2 (63 of 171 – percentile 63.45); ENGINEERING, MULTIDISCIPLINARY – Q2 (55 of 170 – percentile 67.94); MATERIALS SCIENCE, MULTIDISCIPLINARY – Q2 (173 of 384 – percentile 55.08); CHEMESTRY, MULTIDISCIPLINARY – Q2 (84 de 219 – percentile 61.87) – JCI 0.61); (SJR 0.435 – COMPUTER SCIENCE APPLICATIONS – Q2; ENGINEERING (MISCELLANEOUS) – Q2; FLUID FLOW AND TRANSFER PROCESSES – Q2; INSTRUMENTATION – Q2; MATERIALS SCIENCE (MISCELLANEOUS) – Q2; PROCESS CHEMESTRY AND TECHNOLOGY – Q2); (CiteScore 3.0; FLUID FLOW AND TRANSFER PROCESSES – Q2 (34 of 83 – percentile 59); GENERAL ENGINEERING – Q2 (85 of 297 – percentile 71); INSTRUMENTATION – Q2 (49 of 128 percentile 62); COMPUTER SCIENCE APPLICATIONS – Q2 (300 of 693 – percentile 56); GENERAL MATERIALS SCIENCE – Q2 (222 of 455 – percentile 51); PROCESS CHEMESTRY AND TECHNOLOGY – Q2 (30 of 59 – percentile 50)) [64].

8. A. Vázquez-Ingelmo, A. García-Holgado, F. J. García-Peñalvo, and R. Therón, "Proof-of-concept of an information visualization classification approach based on their fine-grained features," *Expert Systems*, e12872, 2022, doi: 10.1111/exsy.12872. ISSN: 0266-4720. (JCR SCIE – COMPUTER SCIENCE, ARTIFICIAL INTELLIGENCE – Q3 (73 of 139 – percentile 47.84); COMPUTER SCIENCE, THEORY & METHODS – Q2 (31 of 110 – percentile 72.27) – JIF 2.587); (JCR JCI – COMPUTER SCIENCE, THEORY & METHODS – Q2 (58 of 137 – percentile 58.03); COMPUTER SCIENCE, ARTIFICIAL INTELLIGENCE – Q2 (80 of 175 – percentile 54.57) – JCI 0.60); (SJR 0.365 – ARTIFICIAL INTELLIGENCE – Q3; COMPUTATIONAL THEORY AND MATHEMATICS – Q3; CONTROL AND SYSTEMS ENGINEERING – Q2; THEORETICAL

COMPUTER SCIENCE – Q3); (CiteScore 3.4; THEORETICAL COMPUTER SCIENCE – Q2 (44 of 120 – percentile 63); COMPUTATIONAL THEORY AND MATHEMATICS – Q2 (44 of 133 – percentile 67); CONTROL AND SYSTEMS ENGINEERING – Q2 (95 of 260 – percentile 63); ARTIFICIAL INTELLIGENCE – Q2 (110 of 227 – percentile 51)) [68].

9. F. J. García-Peñalvo, A. Vázquez-Ingelmo, A. García-Holgado, J. Sampedro-Gómez, A. Sánchez-Puente, V. Vicente-Palacios, P. I. Dorado-Díaz, and P. L. Sánchez, "Application of Artificial Intelligence Algorithms Within the Medical Context for Non-Specialized Users: the CARTIER-IA Platform," *International Journal of Interactive Multimedia and Artificial Intelligence,* vol. 6, no. 6, pp. 46-53, 2021, doi: 10.9781/ijimai.2021.05.005. ISSN: 1989-1660. (JCR SCIE – COMPUTER SCIENCE, ARTIFICIAL INTELLIGENCE – Q2 (58 of 139 – percentile 58.63); COMPUTER SCIENCE, INTERDISCIPLINARY APPLICATIONS – Q2 (55 of 112 – percentile 51.34) – JIF 3.137); (JCR JCI – COMPUTER SCIENCE, INTERDISCIPLINARY APPLICATIONS – Q2 (65 of 142 – percentile 54.58); COMPUTER SCIENCE, ARTIFICIAL INTELLIGENCE – Q2 (68 of 175 – percentile 61.43) – JCI 0.76) [36].

10. F. J. García-Peñalvo, A. Vázquez-Ingelmo, A. García-Holgado, J. Sampedro-Gómez, A. Sánchez-Puente, V. Vicente-Palacios, P. I. Dorado-Díaz, and P. L. Sánchez, "KoopaML: A graphical platform for building machine learning pipelines adapted to health professionals," *International Journal of Interactive Multimedia and Artificial Intelligence,* In Press. ISSN: 1989-1660. (JCR SCIE – COMPUTER SCIENCE, ARTIFICIAL INTELLIGENCE – Q2 (58 of 139 – percentile 58.63); COMPUTER SCIENCE, INTERDISCIPLINARY APPLICATIONS – Q2 (55 of 112 – percentile 51.34) – JIF 3.137); (JCR JCI – COMPUTER SCIENCE, INTERDISCIPLINARY APPLICATIONS – Q2 (65 of 142 –

percentile 54.58); COMPUTER SCIENCE, ARTIFICIAL INTELLIGENCE – Q2 (68 of 175 – percentile 61.43) – JCI 0.76) [35].

11. A. Vázquez-Ingelmo, F. J. García-Peñalvo, R. Therón, " MetaViz - A graphical meta-model instantiator for generating information dashboards and visualizations," *Journal of King Saud University-Computer and Information Sciences,* In Press. ISSN: 1319-1578. (JCR SCIE – COMPUTER SCIENCE, INFORMATION SYSTEMS – Q1 (12 of 164 – percentile 92.99) – JIF 8.839); (JCR JCI – COMPUTER SCIENCE, INFORMATION SYSTEMS – Q1 (30 of 246 – percentile 88.01) – JCI 1.50) [69].

- **ESCI indexed journals:**

1. A. Vázquez-Ingelmo and R. Therón, "Beneficios de la aplicación del paradigma de líneas de productos software para generar dashboards en contextos educativos," RIED. Revista Iberoamericana de Educación a Distancia, vol. 23, no. 2, pp. 169-185, 07/01 2020, doi: 10.5944/ried.23.2.26389. ISSN: 1138-2783. (JCR JCI – EDUCATION & EDUCATIONAL RESEARCH – Q1 (97 of 724 – percentile 86.67) – JCI 1.69). (ESCI) [65].

- **Book chapters**

1. A. Vázquez-Ingelmo, J. Cruz-Benito, F. J. García-Peñalvo, and M. Martín-González, "Scaffolding the OEEU's Data-Driven Ecosystem to Analyze the Employability of Spanish Graduates," in *Global Implications of Emerging Technology Trends*, F. J. García-Peñalvo Ed. Hershey, PA, USA: IGI Global, 2018, pp. 236-255 [18].

- **Proceedings of international conferences:**

1. A. Vázquez-Ingelmo, J. Cruz-Benito, and F. J. García-Peñalvo, "Improving the OEEU's data-driven technological ecosystem's interoperability with GraphQL," in *Proceedings of the 5th International*

Conference on Technological Ecosystems for Enhancing Multiculturality, Cádiz, Spain, 2017, New York, NY, USA: Association for Computing Machinery, p. Article 89, doi: 10.1145/3144826.3145437 [21].

2. J. Cruz-Benito, J. C. Sánchez-Prieto, A. Vázquez-Ingelmo, R. Therón, F. J. García-Peñalvo, and M. Martín-González, "How Different Versions of Layout and Complexity of Web Forms Affect Users After They Start It? A Pilot Experience," Cham, 2018: Springer International Publishing, in Trends and Advances in Information Systems and Technologies, pp. 971-979 [19].
3. A. Vázquez-Ingelmo, F. J. García-Peñalvo, and R. Therón, "Domain engineering for generating dashboards to analyze employment and employability in the academic context," presented at the Proceedings of the Sixth International Conference on Technological Ecosystems for Enhancing Multiculturality, Salamanca, Spain, 2018. [Online]. Available: https://doi.org/10.1145/3284179.3284329 [49].
4. A. Vázquez-Ingelmo, F. J. García-Peñalvo, and R. Therón, "Capturing high-level requirements of information dashboards' components through meta-modeling," presented at the Proceedings of the Seventh International Conference on Technological Ecosystems for Enhancing Multiculturality, León, Spain, 2019. [Online]. Available: https://doi.org/10.1145/3362789.3362837 [50].
5. A. Vázquez-Ingelmo, F. J. García-Peñalvo, and R. Therón, "Automatic generation of software interfaces for supporting decision-making processes. An application of domain engineering and machine learning," presented at the Proceedings of the Seventh International Conference on Technological Ecosystems for Enhancing Multiculturality, León, Spain, 2019. [Online]. Available: https://doi.org/10.1145/3362789.3362923 [55].

6. A. Vázquez-Ingelmo, F. J. García-Peñalvo, and R. Therón, "Addressing Fine-Grained Variability in User-Centered Software Product Lines: A Case Study on Dashboards," Cham, 2019: Springer International Publishing, in New Knowledge in Information Systems and Technologies, pp. 855-864, doi: 10.1007/978-3-030-16181-1_80 [56].

7. A. Vázquez Ingelmo, F. J. García-Peñalvo, R. Therón Sánchez, and M. Á. Conde González, "Extending a dashboard meta-model to account for users' characteristics and goals for enhancing personalization," *Proceedings of LASI-SPAIN 2019. Learning Analytics Summer Institute Spain 2019: Learning Analytics in Higher Education (Vigo, Spain, June 27-28, 2019). CEUR Workshop Proceedings Series,* 2019. [Online]. Available: http://hdl.handle.net/10366/139803 [51].

8. A. Vázquez-Ingelmo, A. García-Holgado, F. J. García-Peñalvo, and R. Therón, "Dashboard Meta-Model for Knowledge Management in Technological Ecosystem: A Case Study in Healthcare," *Proceedings,* vol. 31, no. 1, 2019, doi: 10.3390/proceedings2019031044 [61].

9. A. Vázquez-Ingelmo, F. J. García-Peñalvo, and R. Therón, "Tailored information dashboards: A systematic mapping of the literature," presented at the Proceedings of the XX International Conference on Human Computer Interaction, Donostia, Gipuzkoa, Spain, 2019. [Online]. Available: https://doi.org/10.1145/3335595.3335628 [47].

10. A. Vázquez-Ingelmo, A. García-Holgado, F. J. García-Peñalvo, and R. Therón, "A meta-model to develop learning ecosystems with support for knowledge discovery and decision-making processes," in 2020 15th Iberian Conference on Information Systems and Technologies (CISTI), 24-27 June 2020 2020, pp. 1-6, doi: 10.23919/CISTI49556.2020.9140986 [58].

11. A. Vázquez-Ingelmo, F. J. García-Peñalvo, R. Therón, and A. García-Holgado, "Specifying information dashboards' interactive features

through meta-model instantiation," in Proceedings of LASI-SPAIN 2020. Learning Analytics Summer Institute Spain 2020: Learning Analytics. Time for Adoption? (Valladolid, Spain, June 15-16, 2020), A. Martínez-Monés, A. Álvarez, M. Caeiro-Rodríguez, and Y. Dimitriadis Eds., (CEUR Workshop Proceedings Series, no. 2671). Aachen, Germany: CEUR-WS.org, 2020, pp. 47-59 [53].

12. A. Vázquez-Ingelmo, A. García-Holgado, F. J. García-Peñalvo, and R. Therón, "A Dashboard to Support Decision-Making Processes in Learning Ecosystems: A Metamodel Integration," presented at the Proceedings of the 2020 European Symposium on Software Engineering, Rome, Italy, 2020. [Online]. Available: https://doi.org/10.1145/3393822.3432326 [59].

13. A. Vázquez Ingelmo, F. J. García-Peñalvo, and R. Therón Sánchez, "Aggregation Bias: A Proposal to Raise Awareness Regarding Inclusion in Visual Analytics," (in eng), Trends and Innovations in Information Systems and Technologies, WorldCIST 2020 (Vol. 3, pp. 409-417). Springer Nature., vol. 3, 2020, doi: https://doi.org/10.1007/978-3-030-45697-9_40 [62].

14. A. Vázquez-Ingelmo, F. J. García-Peñalvo, and R. Therón, "Generating Dashboards Using Fine-Grained Components: A Case Study for a PhD Programme," in Learning and Collaboration Technologies. Designing, Developing and Deploying Learning Experiences, Cham, P. Zaphiris and A. Ioannou, Eds., 2020// 2020: Springer International Publishing, pp. 303-314, doi: 10.1007/978-3-030-50513-4_23 [66].

15. A. Vázquez Ingelmo, A. García-Holgado, H. Hernández-Payo, F. J. García-Peñalvo, and R. Therón Sánchez, "Following up the progress of doctoral students and advisors' workload through data visualizations: a case study in a PhD program," (in eng), *Proceedings of LASI-SPAIN 2021. Learning Analytics Summer Institute Spain 2021: Learning Analytics in times*

of COVID-19: Opportunity from crisis (Barcelona, Spain, July 7-9, 2021). CEUR Workshop Proceedings Series, 2021. [Online]. Available: http://ceur-ws.org/Vol-3029/paper06.pdf [67].

16. A. Vázquez-Ingelmo et al., "A platform for management and visualization of medical data and medical imaging," presented at the Eighth International Conference on Technological Ecosystems for Enhancing Multiculturality, Salamanca, Spain, 2020. [Online]. Available: https://doi.org/10.1145/3434780.3436652 [40].
17. A. Vázquez Ingelmo, A. García-Holgado, F. J. García-Peñalvo, and R. Therón Sánchez, "A Meta-modeling Approach to Take into Account Data Domain Characteristics and Relationships in Information Visualizations," in *9th World Conference on Information Systems and Technologies*, Azores, Portugal, Á. Rocha, H. Adeli, G. Dzemyda, F. Moreira, and A. M. R. Correia, Eds., 2021, vol. 2: Springer Nature, in Trends and Innovations in Information Systems and Technologies, WorldCIST [54].
18. A. García-Holgado *et al.*, "User-Centered Design Approach for a Machine Learning Platform for Medical Purpose," in *HCI-COLLAB 2021*, Sao Paulo, Brazil, 8-10, September 2021, Cham, Switzerland: Springer International Publishing, in Human-Computer Interaction, pp. 237-249, doi: 10.1007/978-3-030-92325-9_18 [38].
19. A. Vázquez-Ingelmo *et al.*, "Bringing machine learning closer to non-experts: proposal of a user-friendly machine learning tool in the healthcare domain," presented at the Ninth International Conference on Technological Ecosystems for Enhancing Multiculturality (TEEM'21), Barcelona, Spain, 2021. [Online]. Available: https://doi.org/10.1145/3486011.3486469 [39].
20. A. Vázquez-Ingelmo et al., "A platform to support the visual analysis of the SALMANTICOR study outcomes: conveying cardiological data to

lay users," presented at the Ninth International Conference on Technological Ecosystems for Enhancing Multiculturality (TEEM'21), Barcelona, Spain, 2021. [Online]. Available: https://doi.org/10.1145/3486011.3486471 [37].

21. F. J. García-Peñalvo, A. Vázquez-Ingelmo, and A. García-Holgado, "Fostering Decision-Making Processes in Health Ecosystems through Visual Analytics and Machine Learning," presented at the 9th International Conference on Learning and Collaboration Technologies, Virtual, June 28, 2022 [34].
22. A. Vázquez-Ingelmo, A. García-Holgado, F. J. García-Peñalvo, R. Therón, and R. Colomo-Palacios, "Content-validation questionnaire of a meta-model to ease the learning of data visualization concepts," presented at the Learning Analytics Summer Institute Spain 2022 (LASI Spain 22), Salamanca, Spain, 20-21 June, 2022 [57].
23. A. Vázquez-Ingelmo, F. J. García-Peñalvo, R. Therón, V. Byrd, and J. D. Camba, "A proposal to measure the understanding of data visualization elements in visual analytics applications," presented at the Learning Analytics Summer Institute Spain 2022 (LASI Spain 22), Salamanca, Spain, 20-21 June, 2022 [71].

- **Software released:**
  - A. Vázquez-Ingelmo. "Code repository that supports the research presented in the paper 'Taking advantage of the software product line paradigm to generate customized user interfaces for decision-making processes: a case study on university employability'." https://github.com/AndVazquez/dashboard-spl-assets (accessed 24-05-2022) [217].
  - A. Vázquez-Ingelmo. "Code repository that supports the research presented in the paper 'Information dashboards and tailoring capabilities

- A systematic literature review'." https://github.com/AndVazquez/slr-tailored-dashboards (accessed 24-05-2022) [218].

  - A. Vázquez-Ingelmo. "Code repository that supports the research presented in the paper 'Towards a technological ecosystem to provide information dashboards as a service: A dynamic proposal for supplying dashboards adapted to specific scenarios'." https://github.com/AndVazquez/generation-workflow-example (accessed 24-05-2022) [219].
  - A. Vázquez-Ingelmo. "Code repository that supports the research presented in the paper 'Following up the progress of PhD students and advisors' workload through data visualizations: a case study in a PhD program'." https://github.com/AndVazquez/phd-visualizations-sus (accessed 24-05-2022) [220].

- **Awards:**
  - Best paper award in the track International Workshop on Software Engineering for E-Learning (ISELEAR'17) within the International Conference on Technological Ecosystems for Enhancing Multiculturality (TEEM) 2017 held in Cádiz, Spain between October 18-20, 2017. Award granted for the paper "Improving the OEEU's data-driven technological ecosystem's interoperability with GraphQL" developed jointly to J. Cruz-Benito and F. J. García-Peñalvo.
  - Best paper award in the track International Workshop on Software Engineering for E-Learning (ISELEAR'18) within the International Conference on Technological Ecosystems for Enhancing Multiculturality (TEEM) 2018 held in Salamanca, Spain between October 24-26, 2018. Award granted for the paper "Domain engineering for generating dashboards to analyze employment and employability in the academic context" developed jointly to F. J. García-Peñalvo and R. Therón.

- Best paper award in the track International Workshop on Software Engineering for E-Learning (ISELEAR'19) within the International Conference on Technological Ecosystems for Enhancing Multiculturality (TEEM) 2019 held in León, Spain between October 16-18, 2019. Award granted for the paper "Capturing high-level requirements of information dashboards' components through meta-modeling" developed jointly to F. J. García-Peñalvo and R. Therón.
- Conference best paper of the Learning Analytics Summer Institute (LASI Spain) Salamanca, Spain between June 20-21, 2022. Award granted for the paper "A proposal to measure the understanding of data visualization elements in visual analytics applications" developed jointly to F. J. García-Peñalvo, R. Therón, V. Byrd, and J. D. Camba.

It is worth mentioning the completion of two doctoral internships. The first, an online internship from July 1, 2021, to October 10, 2021, at Østfold University College, Computer Science Department (Halden, Norway). This research stay was focused on validating the meta-model and the results derived from the case study are presented in Section 4.1.

The second research stay was carried out from January 10, 2022, to April 14, 2022, at the Department of Computer Graphics Technology of Purdue University (West Lafayette, Indiana, United States of America). The research was related to data visualization applications and the results can be consulted in the last case study of this Ph.D. thesis.

Finally, this doctoral thesis has received funding from the Spanish *Ministry of Education and Vocational Training* under an FPU fellowship (FPU17/03276).

References

[1] A. Abran, J. W. Moore, P. Bourque, and R. Dupuis, "Guide to the Software Engineering Body of Knowledge. 2004 Version." Los Alamitos, CA, USA: IEEE Computer Society, 2004.

[2] P. Cooper, "Data, information, knowledge and wisdom," *Anaesthesia & Intensive Care Medicine,* vol. 18, no. 1, pp. 55-56, 2017/01/01/ 2017.

[3] J. Rowley, "The wisdom hierarchy: representations of the DIKW hierarchy," *Journal of Information Science,* vol. 33, no. 2, pp. 163-180, 2007/04/01 2007.

[4] C. Zins, "Conceptual approaches for defining data, information, and knowledge," *Journal of the American Society for Information Science and Technology,* vol. 58, no. 4, pp. 479-493, 2007.

[5] D. Bawden and L. Robinson, "Information overload: An overview," *Oxford encyclopedia of political decision making,* 2020.

[6] D. Bawden, C. Holtham, and N. Courtney, "Perspectives on information overload," in *Aslib proceedings*, 1999: MCB UP Ltd.

[7] A. Cairo, *How charts lie: Getting smarter about visual information*. New York, NY, USA: WW Norton & Company, 2019.

[8] A. Cairo, *The truthful art: Data, charts, and maps for communication*. San Francisco, CA, USA: New Riders, 2016.

[9] A. Cairo, *The Functional Art: An introduction to information graphics and visualization*. San Francisco, CA, USA: New Riders, 2012.

[10] C. O. Wilke, *Fundamentals of data visualization: a primer on making informative and compelling figures*. Sebastopol, CA, USA: O'Reilly Media, 2019.

[11] S. L. Franconeri, L. M. Padilla, P. Shah, J. M. Zacks, and J. Hullman, "The Science of Visual Data Communication: What Works," *Psychological Science in the Public Interest,* vol. 22, no. 3, pp. 110-161, 2021.

[12] A. Álvarez-Arana, M. Villamañe-Gironés, and M. Larrañaga-Olagaray, "Improving Assessment Using Visual Learning Analytics," *Education in the Knowledge Society,* vol. 21, no. 9, pp. 1-9, 2020.

[13] E. Dimara, S. Franconeri, C. Plaisant, A. Bezerianos, and P. Dragicevic, "A task-based taxonomy of cognitive biases for information visualization," *IEEE transactions on visualization and computer graphics,* vol. 26, no. 2, pp. 1413-1432, 2018.

[14] D. A. Szafir, "The good, the bad, and the biased: Five ways visualizations can mislead (and how to fix them)," *Interactions,* vol. 25, no. 4, pp. 26-33, 2018.

[15] V. Dibia and D. Ç, "Data2Vis: Automatic Generation of Data Visualizations Using Sequence-to-Sequence Recurrent Neural Networks," *IEEE Computer Graphics and Applications,* vol. 39, no. 5, pp. 33-46, 2019.

[16] F. J. García-Peñalvo, M. J. Rodríguez-Conde, A. M. Seoane-Pardo, M. A. Code-González, V. Zangrando, and A. García-Holgado, "GRIAL (GRupo de investigación en InterAcción y eLearning), USAL," *IE Comunicaciones: Revista Iberoamericana de Informática Educativa,* no. 15, pp. 85-94, 2012.
[17] Grupo GRIAL, "Producción Científica del Grupo GRIAL de 2011 a 2019," Grupo GRIAL, University of Salamanca, Salamanca, Spain2019, Available: https://bit.ly/30l9mLh, Accessed on: 24-05-2022.
[18] A. Vázquez-Ingelmo, J. Cruz-Benito, F. J. García-Peñalvo, and M. Martín-González, "Scaffolding the OEEU's Data-Driven Ecosystem to Analyze the Employability of Spanish Graduates," in *Global Implications of Emerging Technology Trends*, F. J. García-Peñalvo, Ed. Hershey, PA, USA: IGI Global, 2018, pp. 236-255.
[19] J. Cruz-Benito, J. C. Sánchez-Prieto, A. Vázquez-Ingelmo, R. Therón, F. J. García-Peñalvo, and M. Martín-González, "How Different Versions of Layout and Complexity of Web Forms Affect Users After They Start It? A Pilot Experience," Cham, 2018, pp. 971-979: Springer International Publishing.
[20] F. Michavila, J. M. Martínez, M. Martín-González, F. J. García-Peñalvo, and J. Cruz Benito, "Empleabilidad de los titulados universitarios en España. Proyecto OEEU," *Education in the Knowledge Society,* vol. 19, no. 1, pp. 21-39, 2018.
[21] A. Vázquez-Ingelmo, J. Cruz-Benito, and F. J. García-Peñalvo, "Improving the OEEU's data-driven technological ecosystem's interoperability with GraphQL," in *Proceedings of the 5th International Conference on Technological Ecosystems for Enhancing Multiculturality*, Cádiz, Spain, 2017, p. Article 89, New York, NY, USA: Association for Computing Machinery.
[22] F. Michavila, M. Martín-González, J. M. Martínez, F. J. García-Peñalvo, and J. Cruz-Benito, "Analyzing the employability and employment factors of graduate students in Spain: the OEEU information system," presented at the Proceedings of the 3rd International Conference on Technological Ecosystems for Enhancing Multiculturality, Porto, Portugal, 2015. Available: https://doi.org/10.1145/2808580.2808622
[23] F. Michavila, J. M. Martínez, M. Martín-González, F. J. García-Peñalvo, and J. Cruz-Benito, *Barómetro de empleabilidad y empleo de los universitarios en España, 2015 (Primer informe de resultados)*. Madrid: Observatorio de Empleabilidad y Empleo Universitarios, 2016.
[24] F. Michavila, J. M. Martínez, M. Martín-González, F. J. García-Peñalvo, J. Cruz-Benito, and A. Vázquez-Ingelmo, *Barómetro de empleabilidad y empleo universitarios. Edición Máster 2017*. Madrid, España: Observatorio de Empleabilidad y Empleo Universitarios, 2018.
[25] J. M. Toribio-Guzmán, A. García-Holgado, F. Soto Pérez, F. J. García-Peñalvo, and M. Franco Martín, "Usability Evaluation of a Private Social Network on Mental Health for Relatives," *J Med Syst,* vol. 41, no. 9, p. 137, Sep 2017.
[26] F. J. García-Peñalvo, M. F. Martín, A. García-Holgado, J. M. Guzmán, J. L. Antón, and M. C. Sánchez-Gómez, "Psychiatric Patients Tracking Through a Private Social Network for Relatives: Development and Pilot Study," *J Med Syst,* vol. 40, no. 7, p. 172, Jul 2016.
[27] A. García-Holgado and F. J. García-Peñalvo, "A Metamodel Proposal for Developing Learning Ecosystems," in *Learning and Collaboration Technologies. Novel Learning Ecosystems. 4th International Conference, LCT 2017. Held as Part of HCI International 2017*, Vancouver, BC, Canada, 2017, no. 10295, pp. 100-109, Switzerland: Springer International Publishing, 2017.
[28] A. García-Holgado and F. J. García-Peñalvo, "Definición de ecosistemas de aprendizaje independientes de plataforma," in *IV Congreso Internacional sobre Aprendizaje, Innovación y Competitividad (CINAIC)*, Zaragoza, Spain, 2017, pp. 668-673: Servicio de Publicaciones Universidad de Zaragoza.
[29] A. García-Holgado and F. J. García-Peñalvo, "Preliminary validation of the metamodel for developing learning ecosystems," presented at the Proceedings of the 5th International

Conference on Technological Ecosystems for Enhancing Multiculturality, Cádiz, Spain, 2017. Available: https://doi.org/10.1145/3144826.3145439
[30] F. J. García-Peñalvo, "En clave de innovación educativa. Construyendo el nuevo ecosistema de aprendizaje.," presented at the I Congreso Internacional de Tendencias en Innovación Educativa, CITIE 2016, Arequipa, Peru, 2016. Available: https://goo.gl/8HQovc
[31] F. J. García-Peñalvo, "La evolución de los sistemas software educativos: Los ecosistemas tecnológicos de aprendizaje," presented at the Programa de Doctorado en Ingeniería Informática Universidad Complutense de Madrid, Madrid, Spain, 2017. Available: https://goo.gl/Ykwrtv
[32] A. García-Holgado and F. J. García-Peñalvo, "Gestión del conocimiento abierto mediante ecosistemas tecnológicos basados en soluciones Open Source," in *Ecosistemas del Conocimiento Abierto*, J. A. Merlo Vega, Ed. Salamanca, Spain: Ediciones Universidad de Salamanca, 2018.
[33] S. Marcos-Pablos and F. J. García-Peñalvo, "Emotional Intelligence in Robotics: A Scoping Review," Cham, 2022, pp. 66-75: Springer International Publishing.
[34] F. J. García-Peñalvo, A. Vázquez-Ingelmo, and A. García-Holgado, "Fostering Decision-Making Processes in Health Ecosystems through Visual Analytics and Machine Learning," presented at the 9th International Conference on Learning and Collaboration Technologies, Virtual, June 28, 2022.
[35] F. J. García-Peñalvo *et al.*, "KoopaML: A graphical platform for building machine learning pipelines adapted to health professionals," *International Journal of Interactive Multimedia and Artificial Intelligence,* In Press.
[36] F. J. García-Peñalvo *et al.*, "Application of Artificial Intelligence Algorithms Within the Medical Context for Non-Specialized Users: the CARTIER-IA Platform," *International Journal of Interactive Multimedia and Artificial Intelligence,* vol. 6, no. 6, pp. 46-53, 2021.
[37] A. Vázquez-Ingelmo *et al.*, "A platform to support the visual analysis of the SALMANTICOR study outcomes: conveying cardiological data to lay users," in *Proceedings TEEM'21. Ninth International Conference on Technological Ecosystems for Enhancing Multiculturality (Barcelona, Spain, October 27th – 29th, 2021)*, pp. 335–341, New York, NY, USA: Association for Computing Machinery, 2021.
[38] A. García-Holgado *et al.*, "User-Centered Design Approach for a Machine Learning Platform for Medical Purpose," in *HCI-COLLAB 2021*, Sao Paulo, Brazil, 2021, pp. 237-249, Cham, Switzerland: Springer International Publishing.
[39] A. Vázquez-Ingelmo *et al.*, "Bringing machine learning closer to non-experts: proposal of a user-friendly machine learning tool in the healthcare domain," in *Proceedings TEEM'21. Ninth International Conference on Technological Ecosystems for Enhancing Multiculturality (Barcelona, Spain, October 27th – 29th, 2021)*, pp. 324–329, New York, NY, USA: Association for Computing Machinery, 2021.
[40] A. Vázquez-Ingelmo *et al.*, "A platform for management and visualization of medical data and medical imaging," in *Proceedings TEEM'20. Eighth International Conference on Technological Ecosystems for Enhancing Multiculturality*, Salamanca, Spain, pp. 518–522, New York, NY, USA: Association for Computing Machinery, 2020.
[41] J. Cruz-Benito, A. Vázquez-Ingelmo, J. C. Sánchez-Prieto, R. Therón, F. J. García-Peñalvo, and M. Martín-González, "Enabling Adaptability in Web Forms Based on User Characteristics Detection Through A/B Testing and Machine Learning," *IEEE Access,* vol. 6, pp. 2251-2265, 2018.
[42] J. Cruz-Benito, R. Therón, F. J. García-Peñalvo, and E. Pizarro Lucas, "Discovering usage behaviors and engagement in an Educational Virtual World," *Computers in Human Behavior,* vol. 47, pp. 18-25, 2015/06/01/ 2015.

[43] A. García-Holgado and F. J. García-Peñalvo, "Validation of the learning ecosystem metamodel using transformation rules," *Future Generation Computer Systems,* vol. 91, pp. 300-310, 2019/02/01/ 2019.
[44] A. García-Holgado and F. J. García-Peñalvo, "Architectural pattern to improve the definition and implementation of eLearning ecosystems," *Science of Computer Programming,* vol. 129, pp. 20-34, 2016/11/01/ 2016.
[45] A. García-Holgado, F. J. García-Peñalvo, and M. J. Rodríguez-Conde, "Definition of a technological ecosystem for scientific knowledge management in a PhD programme," presented at the Proceedings of the 3rd International Conference on Technological Ecosystems for Enhancing Multiculturality, Porto, Portugal, 2015. Available: https://doi.org/10.1145/2808580.2808686
[46] A. García-Holgado and F. J. Garcia-Peñalvo, "Architectural pattern for the definition of eLearning ecosystems based on Open Source developments," in *2014 International Symposium on Computers in Education (SIIE)*, 2014, pp. 93-98.
[47] A. Vázquez-Ingelmo, F. J. García-Peñalvo, and R. Therón, "Tailored information dashboards: A systematic mapping of the literature," in *Proceedings of the XX International Conference on Human Computer Interaction*, Donostia, Gipuzkoa, Spain, 2019, p. Article 26, New York, NY, USA: Association for Computing Machinery, 2019.
[48] A. Vázquez-Ingelmo, F. J. García-Peñalvo, and R. Therón, "Information Dashboards and Tailoring Capabilities - A Systematic Literature Review," *IEEE Access,* vol. 7, pp. 109673-109688, 2019.
[49] A. Vázquez-Ingelmo, F. J. García-Peñalvo, and R. Therón, "Domain engineering for generating dashboards to analyze employment and employability in the academic context," in *Proceedings of the Sixth International Conference on Technological Ecosystems for Enhancing Multiculturality*, Salamanca, Spain, 2018, pp. 896–901, New York, NY, USA: Association for Computing Machinery, 2018.
[50] A. Vázquez-Ingelmo, F. J. García-Peñalvo, and R. Therón, "Capturing high-level requirements of information dashboards' components through meta-modeling," in *TEEM'19 Proceedings of the Seventh International Conference on Technological Ecosystems for Enhancing Multiculturality*, León, Spain, 2019, pp. 815–821, New York, NY, USA: Association for Computing Machinery, 2019.
[51] A. Vázquez Ingelmo, F. J. García-Peñalvo, R. Therón Sánchez, and M. Á. Conde González, "Extending a dashboard meta-model to account for users' characteristics and goals for enhancing personalization," *Proceedings of LASI-SPAIN 2019. Learning Analytics Summer Institute Spain 2019: Learning Analytics in Higher Education (Vigo, Spain, June 27-28, 2019). CEUR Workshop Proceedings Series,* 2019.
[52] A. Vázquez-Ingelmo, F. J. García-Peñalvo, R. Therón, and M. Á. Conde, "Representing Data Visualization Goals and Tasks through Meta-Modeling to Tailor Information Dashboards," *Applied Sciences,* vol. 10, no. 7, p. 2306, 2020.
[53] A. Vázquez-Ingelmo, F. J. García-Peñalvo, R. Therón, and A. García-Holgado, "Specifying information dashboards' interactive features through meta-model instantiation," in *LASI-SPAIN 2020. Learning Analytics Summer Institute Spain 2020: Learning Analytics. Time for Adoption?*, Valladolid, Spain, 2020, pp. 47-59, Aachen, Germany: CEUR-WS.org.
[54] A. Vázquez Ingelmo, A. García-Holgado, F. J. García-Peñalvo, and R. Therón Sánchez, "A Meta-modeling Approach to Take into Account Data Domain Characteristics and Relationships in Information Visualizations," in *Trends and Innovations in Information Systems and Technologies, WorldCIST 2021*, Azores, Portugal, 2021, vol. 2, pp. 570-580, Cham, Switzerland: Springer Nature, 2021.
[55] A. Vázquez-Ingelmo, F. J. García-Peñalvo, and R. Therón, "Automatic generation of software interfaces for supporting decision-making processes. An application of domain engineering and machine learning," in *EEM'19 Proceedings of the Seventh International*

Conference on Technological Ecosystems for Enhancing Multiculturality, León, Spain, 2019, pp. 1007–1011, New York, NY, USA: Association for Computing Machinery, 2019.
[56] A. Vázquez-Ingelmo, F. J. García-Peñalvo, and R. Therón, "Addressing Fine-Grained Variability in User-Centered Software Product Lines: A Case Study on Dashboards," in *Knowledge in Information Systems and Technologies*, 2019, vol. 1, pp. 855-864, Switzerland: Springer Nature, 2019.
[57] A. Vázquez-Ingelmo, A. García-Holgado, F. J. García-Peñalvo, R. Therón, and R. Colomo-Palacios, "Content-validation questionnaire of a meta-model to ease the learning of data visualization concepts," in *Learning Analytics Summer Institute Spain 2022 (LASI Spain 22)*, Salamanca, Spain, 2022, vol. 3238, pp. 6-11: CEUR-WS.org.
[58] A. Vázquez-Ingelmo, A. García-Holgado, F. J. García-Peñalvo, and R. Therón, "A meta-model to develop learning ecosystems with support for knowledge discovery and decision-making processes," in *2020 15th Iberian Conference on Information Systems and Technologies (CISTI)*, 2020, pp. 1-6.
[59] A. Vázquez-Ingelmo, A. García-Holgado, F. J. García-Peñalvo, and R. Therón, "A Dashboard to Support Decision-Making Processes in Learning Ecosystems: A Metamodel Integration," in *Proceedings of the 2020 European Symposium on Software Engineering - ESSE 2020*, Rome, Italy, 2020, pp. 80–87, New York, NY, USA: Association for Computing Machinery, 2020.
[60] A. Vázquez-Ingelmo, A. García-Holgado, F. J. García-Peñalvo, and R. Therón, "A Meta-Model Integration for Supporting Knowledge Discovery in Specific Domains: A Case Study in Healthcare," *Sensors,* vol. 20, no. 15, p. 4072, 2020.
[61] A. Vázquez-Ingelmo, A. García-Holgado, F. J. García-Peñalvo, and R. Therón, "Dashboard Meta-Model for Knowledge Management in Technological Ecosystem: A Case Study in Healthcare," *Proceedings,* vol. 31, no. 1, p. 44, 2019.
[62] A. Vázquez Ingelmo, F. J. García-Peñalvo, and R. Therón Sánchez, "Aggregation Bias: A Proposal to Raise Awareness Regarding Inclusion in Visual Analytics," in *Trends and Innovations in Information Systems and Technologies, WorldCIST 2020*, 2020, vol. 3, Cham, Switzerland: Springer Nature.
[63] A. Vázquez-Ingelmo, F. J. García-Peñalvo, R. Therón, D. Amo Filvà, and D. Fonseca Escudero, "Connecting domain-specific features to source code: towards the automatization of dashboard generation," *Cluster Computing,* vol. 23, no. 3, pp. 1803-1816, 2020/09/01 2020.
[64] A. Vázquez-Ingelmo, F. J. García-Peñalvo, and R. Therón, "Towards a Technological Ecosystem to Provide Information Dashboards as a Service: A Dynamic Proposal for Supplying Dashboards Adapted to Specific Scenarios," *Applied Sciences,* vol. 11, no. 7, art. 3249, 2021.
[65] A. Vázquez-Ingelmo and R. Therón, "Beneficios de la aplicación del paradigma de líneas de productos software para generar dashboards en contextos educativos," *RIED. Revista Iberoamericana de Educación a Distancia,* vol. 23, no. 2, pp. 169-185, 07/01 2020.
[66] A. Vázquez-Ingelmo, F. J. García-Peñalvo, and R. Therón, "Generating Dashboards Using Fine-Grained Components: A Case Study for a PhD Programme," in *Learning and Collaboration Technologies. Design, Experiences. 7th International Conference, LCT 2020, Held as Part of the 22nd HCI International Conference, HCII 2020*, Copenhagen, Denmark, 2020, no. 12205, pp. 303-314, Cham, Switzerland: Springer Nature, 2020.
[67] A. Vázquez Ingelmo, A. García-Holgado, H. Hernández-Payo, F. J. García-Peñalvo, and R. Therón Sánchez, "Following up the progress of doctoral students and advisors' workload through data visualizations: a case study in a PhD program," *Proceedings of LASI-SPAIN 2021. Learning Analytics Summer Institute Spain 2021: Learning Analytics in times of COVID-19: Opportunity from crisis (Barcelona, Spain, July 7-9, 2021). CEUR Workshop Proceedings Series,* 2021.

[68] A. Vázquez-Ingelmo, A. García-Holgado, F. J. García-Peñalvo, and R. Therón, "Proof-of-concept of an information visualization classification approach based on their fine-grained features," *Expert Systems, e12872*, 2022.
[69] A. Vázquez Ingelmo, F. J. García-Peñalvo, and R. Therón, "MetaViz - A graphical meta-model instantiator for generating information dashboards and visualizations," *Journal of King Saud University - Computer and Information Science,* In Press.
[70] A. Vázquez-Ingelmo, F. J. García-Peñalvo, and R. Therón, "Taking advantage of the software product line paradigm to generate customized user interfaces for decision-making processes: a case study on university employability," *PeerJ Computer Science,* vol. 5, 2019.
[71] A. Vázquez-Ingelmo, F. J. García-Peñalvo, R. Therón, V. Byrd, and J. D. Camba, "A proposal to measure the understanding of data visualization elements in visual analytics applications," in *Learning Analytics Summer Institute Spain 2022 (LASI Spain 22)*, Salamanca, Spain, 2022, vol. 3238, pp. 70-76: CEUR-WS.org.
[72] K. Lewin, "Action research and minority problems," *Journal of social issues,* vol. 2, no. 4, pp. 34-46, 1946.
[73] S. Kemmis, "Point-by-point guide to action research," *Victoria: Deakin University*, 1984.
[74] R. McTaggart and S. Kemmis, *The action research planner*. Deakin university, 1988.
[75] K. Schwaber, "Scrum development process," in *Business object design and implementation*: Springer, 1997, pp. 117-134.
[76] K. Schwaber and M. Beedle, *Agile software development with Scrum*. Prentice Hall Upper Saddle River, 2002.
[77] J. W. Creswell and J. D. Creswell, *Research design: Qualitative, quantitative, and mixed methods approaches*. Sage publications, 2017.
[78] F. J. García-Peñalvo, "Developing robust state-of-the-art reports: Systematic Literature Reviews," *Education in the Knowledge Society,* vol. 23, p. e28600, 2022.
[79] B. Kitchenham and S. Charters, "Guidelines for performing Systematic Literature Reviews in Software Engineering. Version 2.3," School of Computer Science and Mathematics, Keele University and Department of Computer Science, University of Durham, Technical Report EBSE-2007-01, 2007, Available: https://goo.gl/L1VHcw.
[80] B. Kitchenham, "Procedures for performing systematic reviews," Keele University Technical Report TR/SE-0401 and NICTA Technical Report 0400011T.12004.
[81] M. Harsu, *A survey on domain engineering*. Citeseer, 2002.
[82] I. Reinhartz-Berger, S. Cohen, J. Bettin, T. Clark, and A. Sturm, *Domain engineering*. Heidelberg, Germany: Springer, 2013.
[83] M. Voelter and E. Visser, "Product line engineering using domain-specific languages," in *Software Product Line Conference (SPLC), 2011 15th International*, 2011, pp. 70-79: IEEE.
[84] D. Bjørner, "Domain Engineering," in *Formal Methods: State of the Art and New Directions*, P. Boca, J. P. Bowen, and J. Siddiqi, Eds. London: Springer London, 2010, pp. 1-41.
[85] K. C. Kang, S. G. Cohen, J. A. Hess, W. E. Novak, and A. S. Peterson, "Feature-oriented domain analysis (FODA) feasibility study," Carnegie-Mellon University, Software Engineering Institute, Pittsburgh, PA, USA1990.
[86] G. F. Arango, "Domain engineering for software reuse," 1988.
[87] P. Clements and L. Northrop, *Software product lines*. Boston, MA, USA: Addison-Wesley, 2002.
[88] N. Anquetil *et al.*, "Traceability for model driven, software product line engineering," in *ECMDA Traceability Workshop Proceedings*, 2008, vol. 12, pp. 77-86: SINTEF.
[89] A. G. Kleppe, J. Warmer, and W. Bast, "MDA Explained. The Model Driven Architecture: Practice and Promise," ed: Addison-Wesley Longman Publishing Co., Inc., Boston, MA, 2003.

[90] S. Trujillo, D. Batory, and O. Diaz, "Feature oriented model driven development: A case study for portlets," in *Proceedings of the 29th international conference on Software Engineering*, 2007, pp. 44-53: IEEE Computer Society.
[91] S. J. Mellor, K. Scott, A. Uhl, and D. Weise, "Model-Driven Architecture," in *Advances in Object-Oriented Information Systems: OOIS 2002 Workshops Montpellier, France, September 2, 2002 Proceedings*, J.-M. Bruel and Z. Bellahsene, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2002, pp. 290-297.
[92] A. Sarikaya, M. Correll, L. Bartram, M. Tory, and D. Fisher, "What Do We Talk About When We Talk About Dashboards?," *IEEE Transactions on Visualization and Computer Graphics,* vol. 25, no. 1, pp. 682-692, 2019.
[93] S. Few, *Information dashboard design*. Sebastopol, CA, USA: O'Reilly Media, 2006.
[94] S. Wexler, J. Shaffer, and A. Cotgreave, *The big book of dashboards: visualizing your data using real-world business scenarios*. Hoboken, NJ, USA: John Wiley & Sons, 2017.
[95] B. Kitchenham, D. Budgen, and O. P. Brereton, "Using mapping studies as the basis for further research–a participant-observer case study," *Information Software Technology,* vol. 53, no. 6, pp. 638-651, 2011.
[96] M. Petticrew and H. Roberts, *Systematic reviews in the social sciences: A practical guide*. Hoboken, NJ, USA: John Wiley & Sons, 2008, p. 354.
[97] O. Pastushenko, J. Hynek, and T. Hruška, "Generation of Test Samples for Construction of Dashboard Design Guidelines: Impact of Color on Layout Balance," in *World Conference on Information Systems and Technologies 2018 (WorldCIST'18)*, Naples, Italy, 2018, pp. 980-990: Springer International Publishing.
[98] A. Liberati *et al.*, "The PRISMA statement for reporting systematic reviews and meta-analyses of studies that evaluate healthcare interventions: explanation and elaboration," *PLOS Medicine,* vol. 6, no. 7, art. e1000100, 2009.
[99] M. J. Page *et al.*, "PRISMA 2020 explanation and elaboration: updated guidance and exemplars for reporting systematic reviews," *BMJ,* vol. 372, p. n160, 2021.
[100] M. J. Page *et al.*, "The PRISMA 2020 statement: an updated guideline for reporting systematic reviews," *BMJ,* vol. 372, p. n71, 2021.
[101] T. Palpanas, P. Chowdhary, G. Mihaila, and F. Pinel, "Integrated model-driven dashboard development," *Information Systems Frontiers,* vol. 9, no. 2-3, pp. 195-208, 2007.
[102] G. L. Miotto, L. Magnoni, and J. E. Sloper, "The TDAQ Analytics Dashboard: a real-time web application for the ATLAS TDAQ control infrastructure," in *18th International Conference on Computing in High Energy and Nuclear Physics (CHEP 2010)*, Taipei, Taiwan, 2011, vol. 331, no. 2, p. 22019: IOP Publishing.
[103] M. Elias and A. Bezerianos, "Exploration views: understanding dashboard creation and customization for visualization novices," in *IFIP Conference on Human-Computer Interaction*, Lisbon, Portugal, 2011, pp. 274-291, Berlin, Heidelberg: Springer, 2011.
[104] M. Kintz, "A semantic dashboard description language for a process-oriented dashboard design methodology," in *2nd International Workshop on Model-based Interactive Ubiquitous Systems (MODIQUITOUS 2012)*, Copenhagen, Denmark, 2012, vol. 947, pp. 31–36: CEUR Workshop Proceedings.
[105] D. Filonik, R. Medland, M. Foth, and M. Rittenbruch, "A Customisable Dashboard Display for Environmental Performance Visualisations," in *Persuasive Technology. PERSUASIVE 2013*, Sydney, Australia, 2013, pp. 51-62, Berlin Heidelberg: Springer.
[106] O. Belo, P. Rodrigues, R. Barros, and H. Correia, "Restructuring Dynamically Analytical Dashboards Based on Usage Profiles," in *21st International Symposium on Methodologies for Intelligent Systems (ISMIS'2014)*, Roskilde, Denmark, 2014, pp. 445-455: Springer International Publishing.
[107] I. Logre, S. Mosser, P. Collet, and M. Riveill, "Sensor data visualisation: a composition-based approach to support domain variability," in *European Conference on Modelling Foundations and Applications*, York, United Kingdom, 2014, vol. 8569, pp. 101-116: Springer.

[108] S. Radovanović, B. Majstorović, S. Kukolj, and M. Z. Bjelica, "Device Cloud platform with customizable Remote User Interfaces," in *2014 IEEE Fourth International Conference on Consumer Electronics Berlin (ICCE-Berlin)*, Berlin, Germany, 2014, pp. 202-204: IEEE.

[109] B. S. Nascimento, A. S. Vivacqua, and M. R. Borges, "A flexible architecture for selection and visualization of information in emergency situations," in *2016 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, Budapest, Hungary, 2016, pp. 003317-003322: IEEE.

[110] S. Van Hoecke, C. Huys, O. Janssens, R. Verborgh, and R. Van de Walle, "Dynamic Monitoring Dashboards Through Composition of Web and Visualization Services," in *International Internet of Things Summit*, Rome, Italy, 2015, vol. 170, pp. 465-474: Springer.

[111] B. Mayer and R. Weinreich, "A dashboard for microservice monitoring and management," in *2017 IEEE International Conference on Software Architecture Workshops (ICSAW)*, Gothenburg, Sweden, 2017, pp. 66-69: IEEE.

[112] K. Kumar, J. Bose, and S. K. Soni, "A Generic Visualization Framework based on a Data Driven Approach for the Analytics data," in *2017 14th IEEE India Council International Conference (INDICON)*, Roorkee, India, 2017, pp. 1-6: IEEE.

[113] M. Kintz, M. Kochanowski, and F. Koetter, "Creating User-specific Business Process Monitoring Dashboards with a Model-driven Approach," in *MODELSWARD 2017*, Porto, Portugal, 2017, pp. 353-361: ACM.

[114] C. Michel, E. Lavoué, S. George, and M. Ji, "Supporting awareness and self-regulation in project-based learning through personalized dashboards," *International Journal of Technology Enhanced Learning,* vol. 9, no. 2/3, pp. 204-226, 2017.

[115] H. Santos, V. Dantas, V. Furtado, P. Pinheiro, and D. L. McGuinness, "From data to city indicators: a knowledge graph for supporting automatic generation of dashboards," in *European Semantic Web Conference (ESWC 2017)*, Portorož, Slovenia, 2017, vol. 10250, pp. 94-108: Springer.

[116] I. Dabbebi, S. Iksal, J.-M. Gilliot, M. May, and S. Garlatti, "Towards Adaptive Dashboards for Learning Analytic: An Approach for Conceptual Design and implementation," in *9th International Conference on Computer Supported Education (CSEDU 2017)*, Porto, Portugal, 2017, vol. 1, pp. 120-131: SCITEPRESS.

[117] W. Noonpakdee, T. Khunkornsiri, A. Phothichai, and K. Danaisawat, "A framework for analyzing and developing dashboard templates for small and medium enterprises," in *2018 5th International Conference on Industrial Engineering and Applications (ICIEA)*, Singapore, Singapore, 2018, pp. 479-483: IEEE.

[118] A. Cardoso, C. J. Vieira Teixeira, and J. Sousa Pinto, "Architecture for Highly Configurable Dashboards for Operations Monitoring and Support," *Studies in Informatics Control,* vol. 27, no. 3, pp. 319-330, 2018.

[119] M. A. Yalçın, N. Elmqvist, and B. B. Bederson, "Keshif: Rapid and expressive tabular data exploration for novices," *IEEE Transactions on Visualization and Computer Graphics,* vol. 24, no. 8, pp. 2339-2352, 2018.

[120] S. Arjun, "Personalizing data visualization and interaction," in *Adjunct Publication of the 26th Conference on User Modeling, Adaptation and Personalization*, Singapore, Singapore, 2018, pp. 199-202: ACM.

[121] G. Petasis, A. Triantafillou, and E. Karstens, "YourDataStories: Transparency and Corruption Fighting Through Data Interlinking and Visual Exploration," in *International Conference on Internet Science (INSCI 2017)*, Thessaloniki, Greece, 2017, vol. 10750, pp. 95-108: Springer.

[122] G. G. Chua *et al.*, "A Flexible Qualitative Data Analytics Dashboard," in *TENCON 2018 - 2018 IEEE Region 10 Conference*, Jeju, Korea, 2018, pp. 1865-1869: IEEE.

[123] S. V. Hautte *et al.*, "A Dynamic Dashboarding Application for Fleet Monitoring Using Semantic Web of Things Technologies," *Sensors,* vol. 20, no. 4, p. 1152, 2020.

[124] E. Rojas, V. Bastidas, and C. Cabrera, "Cities-Board: A Framework to Automate the Development of Smart Cities Dashboards," *IEEE Internet of Things Journal,* vol. 7, no. 10, pp. 10128-10136, 2020.
[125] A. Tundo, C. Castelnovo, M. Mobilio, O. Riganelli, and L. Mariani, "Declarative Dashboard Generation," in *2020 IEEE International Symposium on Software Reliability Engineering Workshops (ISSREW)*, Coimbra, Portugal, 2020, pp. 215-218: IEEE.
[126] D. Orlovskyi and A. Kopp, "A Business Intelligence Dashboard Design Approach to Improve Data Analytics and Decision Making," presented at the 7th International Conference "Information Technology and Interactions" (IT&I-2020), Kyiv, Ukraine, 2020.
[127] S. D. Col, R. Ciucanu, M. Soare, N. Bouarour, and S. Amer-Yahia, "DashBot: An ML-Guided Dashboard Generation System," presented at the 30th ACM International Conference on Information and Knowledge Management, Queensland, Australia, 2021. Available: https://doi.org/10.1145/3459637.3481968
[128] O. Pastushenko, J. Hynek, and T. Hruška, "Evaluation of user interface design metrics by generating realistic-looking dashboard samples," *Expert Systems,* vol. 38, no. 5, p. e12434, 2021.
[129] R. Agrawal and R. Srikant, "Fast algorithms for mining association rules," in *Proc. 20th int. conf. very large data bases, VLDB*, Santiago, Chile, 1994, vol. 1215, pp. 487-499.
[130] O. Belo, H. Correia, P. Rodrigues, and R. Barros, "A personalization system for data visualization platforms," in *2016 Sixth International Conference on Innovative Computing Technology (INTECH)*, Dublin, Ireland, 2016, pp. 162-167: IEEE.
[131] K. Reinecke *et al.*, "Predicting users' first impressions of website aesthetics with a quantification of perceived visual complexity and colorfulness," in *CHI '13: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, Paris, France, 2013, pp. 2049-2058: ACM.
[132] M. Dyczkowski, J. Korczak, and H. Dudycz, "Multi-criteria evaluation of the intelligent dashboard for SME managers based on scorecard framework," in *2014 Federated Conference on Computer Science and Information Systems*, Warsaw, Poland, 2014, pp. 1147-1155: IEEE.
[133] P. Saraiya, C. North, and K. Duca, "An insight-based methodology for evaluating bioinformatics visualizations," *IEEE Transactions on Visualization and Computer Graphics,* vol. 11, no. 4, pp. 443-456, 2005.
[134] J. Sauro, *A practical guide to the system usability scale: Background, benchmarks & best practices*. Scotts Valley, CA, US: Createspace Independent Pub, 2011.
[135] L. Wilkinson, "The grammar of graphics," in *Handbook of computational statistics*: Springer, 2012, pp. 375-414.
[136] B. Selic, "The pragmatics of model-driven development," *IEEE Software,* vol. 20, no. 5, pp. 19-25, 2003.
[137] J. M. Álvarez, A. Evans, and P. Sammut, "Mapping between Levels in the Metamodel Architecture," in *«UML» 2001 — The Unified Modeling Language. Modeling Languages, Concepts, and Tools. UML 2001. Lecture Notes in Computer Science*, vol. 2185, M. Gogolla and C. Kobryn, Eds. Berlin, Heidelberg: Springer, 2001, pp. 34-46.
[138] A. Satyanarayan, D. Moritz, K. Wongsuphasawat, and J. Heer, "Vega-Lite: A Grammar of Interactive Graphics," *IEEE Transactions on Visualization and Computer Graphics,* vol. 23, no. 1, pp. 341-350, 2017.
[139] J. J. López-Fernández, J. S. Cuadrado, E. Guerra, and J. De Lara, "Example-driven meta-model development," *Software & Systems Modeling,* vol. 14, no. 4, pp. 1323-1347, 2015.
[140] J. Mackinlay, P. Hanrahan, and C. Stolte, "Show me: Automatic presentation for visual analysis," *IEEE transactions on visualization computer graphics,* vol. 13, no. 6, 2007.
[141] J. Mackinlay, "Automating the design of graphical presentations of relational information," *Acm Transactions On Graphics (Tog),* vol. 5, no. 2, pp. 110-141, 1986.
[142] N. Iliinsky and J. Steele, *Designing data visualizations: Representing informational Relationships*. Sebastopol, CA, USA: O'Reilly Media, Inc., 2011.

[143] T. Munzner, *Visualization analysis and design*. Boca Raton, FL, USA: AK Peters/CRC Press, 2014.
[144] S. Berinato, *Good charts: The HBR guide to making smarter, more persuasive data visualizations*. Brighton, MA, USA: Harvard Business Review Press, 2016.
[145] A. V. Maltese, J. A. Harsh, and D. Svetina, "Data visualization literacy: Investigating data interpretation along the novice—expert continuum," *Journal of College Science Teaching*, vol. 45, no. 1, pp. 84-90, 2015.
[146] B. Shneiderman, "The eyes have it: a task by data type taxonomy for information visualizations," in *The craft of information visualization*Burlington, MA, USA: Morgan Kaufmann, 2003, pp. 364-371.
[147] H. Lam, M. Tory, and T. Munzner, "Bridging from goals to tasks with design study analysis reports," *IEEE transactions on visualization and computer graphics*, vol. 24, no. 1, pp. 435-445, 2017.
[148] R. Amar, J. Eagan, and J. Stasko, "Low-level components of analytic activity in information visualization," in *IEEE Symposium on Information Visualization, 2005. INFOVIS 2005.*, 2005, pp. 111-117: IEEE.
[149] M. Tory and T. Moller, "Rethinking Visualization: A High-Level Taxonomy," in *IEEE Symposium on Information Visualization*, 2004, pp. 151-158.
[150] H. d. Boer. (2019, 19-05-2022)). *The lie-factor / baseline paradox*. Available: https://nightingaledvs.com/the-lie-factor-and-the-baseline-paradox/
[151] E. Tufte and P. Graves-Morris, "The visual display of quantitative information.; 1983," ed. Cheshire, CT, USA: Graphics Press, 2014.
[152] D. F. Williamson, R. A. Parker, and J. S. Kendrick, "The box plot: a simple visual method to interpret data," *Annals of internal medicine*, vol. 110, no. 11, pp. 916-921, 1989.
[153] W. H. Finch and M. E. Hernández Finch, "Poverty and Covid-19: rates of incidence and deaths in the United States during the first 10 weeks of the pandemic," *Frontiers in Sociology*, vol. 5, p. 47, 2020.
[154] A. Vázquez Ingelmo, F. J. García-Peñalvo, and R. Therón. (2021). *Ecore version of the meta-model for information dashboards (v3)*. Available: https://doi.org/10.5281/zenodo.5788981
[155] C. Gacek and M. Anastasopoules, "Implementing product line variabilities," in *ACM SIGSOFT Software Engineering Notes*, 2001, vol. 26, no. 3, pp. 109-117: ACM.
[156] J. Liebig, S. Apel, C. Lengauer, C. Kästner, and M. Schulze, "An analysis of the variability in forty preprocessor-based software product lines," in *Proceedings of the 32nd ACM/IEEE International Conference on Software Engineering-Volume 1*, 2010, pp. 105-114: ACM.
[157] J.-M. Favre, "Preprocessors from an abstract point of view," in *Reverse Engineering, 1996., Proceedings of the Third Working Conference on*, 1996, pp. 287-296: IEEE.
[158] H. Spencer and G. Collyer, "#ifdef considered harmful, or portability experience with C News," 1992.
[159] S. Jarzabek, P. Bassett, H. Zhang, and W. Zhang, "XVCL: XML-based variant configuration language," in *Proceedings of the 25th International Conference on Software Engineering*, 2003, pp. 810-811: IEEE Computer Society.
[160] H. Zhang, S. Jarzabek, and S. M. Swe, "XVCL approach to separating concerns in product family assets," in *International Symposium on Generative and Component-Based Software Engineering*, 2001, pp. 36-47: Springer.
[161] G. Kiczales *et al.*, "Aspect-oriented programming," in *European conference on object-oriented programming*, 1997, pp. 220-242: Springer.
[162] G. M. Waku, C. M. Rubira, and L. P. Tizzei, "A case study using aop and components to build software product lines in android platform," in *Software Engineering and Advanced Applications (SEAA), 2015 41st Euromicro Conference on*, 2015, pp. 418-421: IEEE.
[163] S.-h. Heo and E. M. Choi, "Representation of variability in software product line using aspect-oriented programming," in *Software Engineering Research, Management and Applications, 2006. Fourth International Conference on*, 2006, pp. 66-73: IEEE.

[164] M. Voelter and I. Groher, "Product line implementation using aspect-oriented and model-driven software development," in *Software Product Line Conference, 2007. SPLC 2007. 11th International*, 2007, pp. 233-242: IEEE.
[165] S. Clark. (2018). *Render your first network configuration template using Python and Jinja2*. Available: https://blogs.cisco.com/developer/network-configuration-template
[166] M. Bostock, V. Ogievetsky, and J. Heer, "D³ Data-Driven Documents," *IEEE Transactions on Visualization and Computer Graphics,* vol. 17, no. 12, pp. 2301-2309, 2011.
[167] J. J. López-Fernández, E. Guerra, and J. De Lara, "Assessing the Quality of Meta-models," in *MoDeVVa 2014. Model-Driven Engineering, Verification and Validation. Proceedings of the 11th Workshop on Model-Driven Engineering, Verification and Validation co-located with 17th International Conference on Model Driven Engineering Languages and Systems (MODELS 2014) (Valencia, Spain, September 30, 2014)*, no. 1235, pp. 3-12, Aachen, Germany: CEUR-WS.org, 2014.
[168] R. Skjong and B. H. Wentworth, "Expert judgment and risk perception," in *Proceedings of the Eleventh (2001) International Offshore and Polar Engineering Conference (Stavanger, Norway, June 17-22, 2001)*: International Society of Offshore and Polar Engineers, 2001.
[169] J. Escobar-Pérez and A. Martínez, "Validez de contenido y juicio de expertos: Una aproximación a su utilización," *Avances en Medición,* vol. 6, pp. 27-36, 01/01 2008.
[170] L. Sweeney, "Discrimination in Online Ad Delivery: Google ads, black names and white names, racial discrimination, and click advertising," *Queue,* vol. 11, no. 3, pp. 10-29, 2013.
[171] M. Garcia, "Racist in the machine: The disturbing implications of algorithmic bias," *World Policy Journal,* vol. 33, no. 4, pp. 111-117, 2016.
[172] S. Hajian, F. Bonchi, and C. Castillo, "Algorithmic bias: From discrimination discovery to fairness-aware data mining," in *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, 2016, pp. 2125-2126: ACM.
[173] L. Harrison, F. Yang, S. Franconeri, and R. Chang, "Ranking visualizations of correlation using weber's law," *IEEE transactions on visualization and computer graphics,* vol. 20, no. 12, pp. 1943-1952, 2014.
[174] C. O'Neil, *On being a data skeptic*. " O'Reilly Media, Inc.", 2013.
[175] S. Ramanana-Rahary, M. Zitt, and R. Rousseau, "Aggregation properties of relative impact and other classical indicators: Convexity issues and the Yule-Simpson paradox," *Scientometrics,* vol. 79, no. 2, pp. 311-327, 2009.
[176] E. H. Simpson, "The interpretation of interaction in contingency tables," *Journal of the Royal Statistical Society: Series B (Methodological),* vol. 13, no. 2, pp. 238-241, 1951.
[177] Y. Guo, C. Binnig, and T. Kraska, "What you see is not what you get!: Detecting Simpson's Paradoxes during Data Exploration," in *Proceedings of the 2nd Workshop on Human-In-the-Loop Data Analytics*, 2017, p. 2: ACM.
[178] K. Wongsuphasawat, D. Moritz, A. Anand, J. Mackinlay, B. Howe, and J. Heer, "Towards a general-purpose query language for visualization recommendation," presented at the Proceedings of the Workshop on Human-In-the-Loop Data Analytics, San Francisco, California, 2016. Available: https://doi.org/10.1145/2939502.2939506
[179] J. Kuzilek, M. Hlosta, and Z. Zdrahal, "Open university learning analytics dataset," *Scientific data,* vol. 4, p. 170171, 2017.
[180] F. J. García-Peñalvo, "Education in knowledge society: A new PhD programme approach," in *Proceedings of the First International Conference on Technological Ecosystems for Enhancing Multiculturality (TEEM'13) (Salamanca, Spain, November 14-15, 2013)*, pp. 575-577, New York, NY, USA: ACM, 2013.
[181] J. Brooke, "SUS-A quick and dirty usability scale," *Usability evaluation in industry,* vol. 189, no. 194, pp. 4-7, 1996.
[182] A. Bangor, P. Kortum, and J. Miller, "Determining what individual SUS scores mean: Adding an adjective rating scale," *Journal of usability studies,* vol. 4, no. 3, pp. 114-123, 2009.

[183] GRIAL Group, "GRIAL Research Group Scientific Production Report (2011-2017) Version 2.0," GRIAL Research Group, University of Salamanca, Salamanca, Spain2018, Available: https://goo.gl/kiUFn9, Accessed on: 13-05-2022.
[184] A. V. Pandey, A. Manivannan, O. Nov, M. Satterthwaite, and E. Bertini, "The persuasive power of data visualization," *IEEE transactions on visualization and computer graphics,* vol. 20, no. 12, pp. 2211-2220, 2014.
[185] A. Vázquez-Ingelmo, F. García-Peñalvo, R. Therón, and A. García-Holgado, *Specifying information dashboards' interactive features through meta-model instantiation* (Proceedings of LASI-SPAIN 2020. Learning Analytics Summer Institute Spain 2020: Learning Analytics. Time for Adoption? Valladolid, Spain, June 15-16, 2020). Aachen, Germany: CEUR-WS.org, 2020.
[186] S. C. Albright, W. Winston, and C. Zappe, *Data analysis and decision making*. Cengage Learning, 2010.
[187] T. Bray, J. Paoli, C. M. Sperberg-McQueen, E. Maler, and F. Yergeau, "Extensible markup language (XML)," *World Wide Web Journal,* vol. 2, no. 4, pp. 27-66, 1997.
[188] D. C. Fallside, "XML schema part 0: Primer," *W3C, April 2000,* 2000.
[189] P. Mildenberger, M. Eichelberg, and E. Martin, "Introduction to the DICOM standard," *European Radiology,* vol. 12, no. 4, pp. 920-927, 2002/04/01 2002.
[190] P. A. Harris, R. Taylor, R. Thielke, J. Payne, N. Gonzalez, and J. G. Conde, "Research electronic data capture (REDCap)—A metadata-driven methodology and workflow process for providing translational research informatics support," *Journal of Biomedical Informatics,* vol. 42, no. 2, pp. 377-381, 2009/04/01/ 2009.
[191] G. Litjens *et al.*, "A survey on deep learning in medical image analysis," *Medical Image Analysis,* vol. 42, pp. 60-88, Dec 2017.
[192] S. González Izard, R. Sánchez Torres, Ó. Alonso Plaza, J. A. Juanes Méndez, and F. J. García-Peñalvo, "Nextmed: Automatic Imaging Segmentation, 3D Reconstruction, and 3D Model Visualization Platform Using Augmented and Virtual Reality," *Sensors (Basel, Switzerland),* vol. 20, no. 10, p. 2962, 2020.
[193] S. G. Izard, J. A. Juanes, F. J. García Peñalvo, J. M. G. Estella, M. J. S. Ledesma, and P. Ruisoto, "Virtual Reality as an Educational and Training Tool for Medicine," *Journal of Medical Systems,* vol. 42, no. 3, p. 50, 2018/02/01 2018.
[194] J. Bosch, "From software product lines to software ecosystems," in *SPLC*, 2009, vol. 9, pp. 111-119.
[195] L. Chen, M. Ali Babar, and N. Ali, "Variability management in software product lines: a systematic review," 2009.
[196] C. Kästner, S. Apel, and M. Kuhlemann, "Granularity in software product lines," in *2008 ACM/IEEE 30th International Conference on Software Engineering*, 2008, pp. 311-320: IEEE.
[197] J. Van Gurp, J. Bosch, and M. Svahnberg, "On the notion of variability in software product lines," in *Proceedings Working IEEE/IFIP Conference on Software Architecture*, 2001, pp. 45-54: IEEE.
[198] J. W. Tukey, *Exploratory data analysis*. Reading, MA, 1977.
[199] J. I. Melero-Alegria *et al.*, "SALMANTICOR study. Rationale and design of a population-based study to identify structural heart disease abnormalities: a spatial and machine learning analysis," *BMJ Open,* vol. 9, no. 2, p. e024605, 2019.
[200] M. Keck, E. Stoll, and D. Kammer, "A Didactic Framework for Analyzing Learning Activities to Design InfoVis Courses," *IEEE Computer Graphics and Applications,* vol. 41, no. 6, pp. 80-90, 2021.
[201] D. R. Krathwohl, "A Revision of Bloom's Taxonomy: An Overview," *Theory Into Practice,* vol. 41, no. 4, pp. 212-218, 2002/11/01 2002.
[202] B. Bloom, M. Engelhart, E. Furst, W. Hill, and D. Krathwohl, *Taxonomy of educational objectives: The classification of educational goals: Handbook I: Cognitive Domain*. New York, NY, USA: D. Mckay, 1956.

[203] R. Morgan, G. Grossmann, M. Schrefl, M. Stumptner, and T. Payne, "VizDSL: A Visual DSL for Interactive Information Visualization," Cham, 2018, pp. 440-455: Springer International Publishing.
[204] M. Correll, E. Bertini, and S. Franconeri, "Truncating the y-axis: Threat or menace?," in *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, Honolulu, HI, USA, 2020, pp. 1-12: ACM.
[205] C. Plaisant, "The challenge of information visualization evaluation," in *Proceedings of the working conference on Advanced visual interfaces*, 2004, pp. 109-116: ACM.
[206] D. Patil and H. Mason, *Data Driven*. Sebastopol, CA, USA: O'Reilly Media, Inc., 2015.
[207] E. R. Tufte and D. Robins, *Visual explanations*. Graphics Cheshire, CT, 1997.
[208] E. R. Tufte, N. H. Goeler, and R. Benson, *Envisioning information*. Graphics press Cheshire, CT, 1990.
[209] E. R. Tufte, "The visual display of quantitative information," *The Journal for Healthcare Quality (JHQ),* vol. 7, no. 3, p. 15, 1985.
[210] M. H. Miraz, M. Ali, and P. S. Excell, "Adaptive user interfaces and universal usability through plasticity of user interface design," *Computer Science Review,* vol. 40, p. 100363, 2021/05/01/ 2021.
[211] K. Z. Gajos, M. Czerwinski, D. S. Tan, and D. S. Weld, "Exploring the design space for adaptive graphical user interfaces," presented at the Proceedings of the working conference on Advanced visual interfaces, Venezia, Italy, 2006. Available: https://doi.org/10.1145/1133265.1133306
[212] A. Bunt, C. Conati, and J. McGrenere, "What role can adaptive support play in an adaptable system?," presented at the Proceedings of the 9th international conference on Intelligent user interfaces, Funchal, Madeira, Portugal, 2004. Available: https://doi.org/10.1145/964442.964465
[213] D. Sacha, H. Senaratne, B. C. Kwon, G. Ellis, and D. A. Keim, "The Role of Uncertainty, Awareness, and Trust in Visual Analytics," *IEEE Transactions on Visualization and Computer Graphics,* vol. 22, no. 1, pp. 240-249, 2016.
[214] T. M. Green, W. Ribarsky, and B. Fisher, "Building and Applying a Human Cognition Model for Visual Analytics," *Information Visualization,* vol. 8, no. 1, pp. 1-13, 2009.
[215] B. C. Kwon, B. Fisher, and J. S. Yi, "Visual analytic roadblocks for novice investigators," in *2011 IEEE Conference on Visual Analytics Science and Technology (VAST)*, 2011, pp. 3-11.
[216] J. B. Warmer and A. G. Kleppe, *The object constraint language: getting your models ready for MDA*. Boston, MA, USA: Addison-Wesley Professional, 2003.
[217] A. Vázquez-Ingelmo. (2018, 24-05-2022)). *Code repository that supports the research presented in the paper 'Taking advantage of the software product line paradigm to generate customized user interfaces for decision-making processes: a case study on university employability'*. Available: https://github.com/AndVazquez/dashboard-spl-assets
[218] A. Vázquez-Ingelmo. (2019, 24-05-2022)). *Code repository that supports the research presented in the paper 'Information dashboards and tailoring capabilities - A systematic literature review'*. Available: https://github.com/AndVazquez/slr-tailored-dashboards
[219] A. Vázquez-Ingelmo. (2021, 24-05-2022)). *Code repository that supports the research presented in the paper 'Towards a technological ecosystem to provide information dashboards as a service: A dynamic proposal for supplying dashboards adapted to specific scenarios'*. Available: https://github.com/AndVazquez/generation-workflow-example
[220] A. Vázquez-Ingelmo. (2021, 24-05-2022)). *Code repository that supports the research presented in the paper 'Following up the progress of PhD students and advisors' workload through data visualizations: a case study in a PhD program'*. Available: https://github.com/AndVazquez/phd-visualizations-sus

7 Appendixes

The following appendixes contains the different papers related to the research scenarios carried out during the development of this thesis.

7.1 Appendix A. Scaffolding the OEEU's data-driven ecosystem to analyze the employability of Spanish graduates

Chapter 13
Scaffolding the OEEU's Data-Driven Ecosystem to Analyze the Employability of Spanish Graduates

Andrea Vázquez-Ingelmo
University of Salamanca, Spain

Juan Cruz-Benito
University of Salamanca, Spain

Francisco J. García-Peñalvo
University of Salamanca, Spain

Martín Martín-González
Technical University of Madrid, Spain

ABSTRACT

This chapter outlines the technological evolution experimented by the Observatory for University Employability and Employment's information system to become a data-driven technological ecosystem. This observatory collects data from more than 50 Spanish universities and their graduate students (bachelor's degree, master's degree) with the goal of measuring the factors that lead to students' employability and employment. The goals pursued by the observatory need a strong technological support to gather, process, and disseminate the related data. The system that supports these tasks has evolved from a standard (traditional) information system to a data-driven ecosystem, which provides remarkable benefits covering the observatory's requirements. The benefits, the foundations, and the way the data-driven ecosystem is built will be described throughout the chapter, as well as how the information obtained is exploited in order to provide insights about the employment and employability variables.

DOI: 10.4018/978-1-5225-4944-4.ch013

1. INTRODUCTION

The Observatory of University Employability and Employment (also known as *OEEU* using the Spanish initials for *Observatorio de Empleabilidad and Empleo Universitarios*) http://www.oeeu.org/, is an organization composed by researchers and technicians who work together from different parts of Spain with a unified methodology. The purpose of the Observatory is to produce, analyze and spread information and insights regarding the graduates' employability and employment in Spain. The Observatory is under the direction of the UNESCO Chair in University Management and Policy (based in the Universidad Politécnica de Madrid, Spain) and it relies on the mentoring of an Expert Council composed of national and international academic and university experts. This project is also developed in collaboration with the "La Caixa" Foundation, the Conference of Rectors of Spanish Universities (CRUE) and the GRIAL Research Group of the University of Salamanca (Peñalvo et al., 2012).

The Observatory's vision is to become the information reference for understanding and exploiting knowledge about variables related to employability and university employment and its behavior. To reach this vision, the Observatory has the following goals (Michavila, Martín-González, Martínez, García-Peñalvo, & Cruz-Benito, 2015; Michavila, Martínez, Martín-González, García-Peñalvo, & Cruz-Benito, 2016):

- To understand the evolution of the employability and employment, and its characteristics related to university graduates.
- To develop a system and a uniform methodology for measuring indicators about employability and employment of graduates.
- To generate information on the employability and employment of university comparable between regions, branches of study (knowledge areas) and professional profiles, among others.
- To support the development of strategies and employment policies for universities, basing it on well-founded studies and information.
- To understand the mechanisms and actions that use the Spanish universities to promote employment and employability of their graduates.
- To provide information to individual universities to adjust their academic supply and training demands to the labor market based on reliable data.

These objectives seek to resolve the lack of public information (and its analysis) regarding employment and university employability. To achieve them, the Observatory is developing, implementing and exploiting a series of data-driven products (Patil, 2012).

This data-driven approach helps the Observatory gain knowledge and wisdom from the gathered data. The data gathering procedure of the Observatory and its storage are useless if there are no other procedures to generate knowledge from raw data. Large volumes of data do not provide knowledge and wisdom by itself, but data is the base of the taxonomy of knowledge (Zeleny, 1987), and that is why it is also the base of the Observatory's system. In order to act wisely, it is necessary to have knowledge, information and data about the tasks to be solved (Zeleny, 1987).

The Observatory's studies are supported by an information system implemented to accomplish the organization's main goals. As referenced before, the Observatory products, as well as the organization itself are data-driven (Patil & Mason, 2015). This means the information system built for the Observatory has a series of characteristics and components that facilitate the data collection, analysis, presentation and exploitation, as it will be explained below.

The first Observatory's study was released in 2015 and its goal was to collect and analyze information about degree students who graduated during the 2009-2010 course. Currently in 2017, the study is now in its second edition. The target of this second study are the university master's degree students (graduated in the 2013-2014 academic year).

However, the magnitude (a growing volume of data) and the continuity of these studies, as well as the broad vision of the Observatory, makes the traditional information systems not enough. As it will be detailed throughout this chapter, the challenges of the Observatory need the support of a more powerful system. That "powerful system" could be consider as a *data-driven* or *information-driven technological ecosystem*, because this kind of collaborative environment potentially fits the Observatory requirements.

A technological ecosystem can be seen as a set of different components connected through information flows in a physical environment that supports such flows (García-Holgado, García-Peñalvo, Hernández-García, & Llorens-Largo, 2015). Ecosystems make possible the provision of new and improved services that isolated tools or systems cannot be able to provide in order to solved knowledge and information management challenges inside any kind of institution or company (Garcia-Peñalvo & Garcia-Holgado, 2016). Moreover, technological ecosystems provide better support to the management of information and knowledge in heterogeneous environments (García-Holgado, Cruz-Benito, & García-Peñalvo, 2015; Garcia-Peñalvo & Garcia-Holgado, 2016).

In the case of the Observatory, the implementation of a technological ecosystem should be made considering a data-driven approach, resulting into a "data-driven ecosystem". This approach implies the incorporation of the characteristics of a technological ecosystem in addition to the data-driven philosophy. The purpose of a data-driven ecosystem is to create a collaborative environment to gather, process, analyze and disseminate data, in order to make evidence-based decisions by enabling the horizontal interaction of different users, stakeholders, systems and tools. These, among others, are the characteristics that make a data-driven ecosystem suitable for the Observatory.

In addition to the above, the data-driven ecosystem provides the adequate features to support the Observatory's data-as-a-service (DaaS) approach. This means that the components within the ecosystem themselves, as well as other users, can consume data from other components on-demand and also produce and communicating data to other components (by encouraging the independence between components, their decoupling and enabling interoperability (Terzo, Ruiu, Bucci, & Xhafa, 2013). According to this philosophy and working procedure, the data-driven ecosystem stablishes a data-as-a-service approach that enables the consumption of the Observatory's raw or processed data, independently of the platform or location of the consumers and only depending on the consumer's goals and the permissions available in the ecosystem.

In 2015, the first implementation of the Observatory's information system faced a series of technical issues with keeping in mind the meeting of requirements at that moment. Later, considering the vision of becoming a reference information source, was required to make important technological changes to accomplish it. As the OEEU evolves into a fully data-driven organization, it deals with a lot of difficulties regarding the collection and sharing of heterogeneous data from different information sources, as well as it experiences the challenges associated to knowledge management and scalability. These difficulties could be addressed through a collaborative environment like an information ecosystem that could allow the management of heterogeneous data and knowledge (Cruz-Benito, García-Peñalvo, & Therón, 2014;

García-Holgado, Cruz-Benito, et al., 2015) and could report higher levels of scalability because of the independence and low decoupling of the components involved in the ecosystem.

The purpose of this chapter is to outline the current Observatory's technological ecosystem, to describe the evolution process from a monolithic information system to a data-driven ecosystem and to present the benefits and results derived from this transition.

The content is organized following the next structure: section 2 provides a description of the different technical and organizational challenges associated to the OEEU mission and their system's goals; the third section presents the system's transition to a new version that improves previously identified issues and weakness, as well as the data structuration principles, the users involved and the technologies used; section 4 outlines the results of the Observatory and the outcomes of the designed data-driven ecosystem. Finally, the fifth section discusses the problems solved, the issues to be solved and the next challenges in the short to medium term, followed by the sixth and last section, where the conclusions of this work are presented.

2. THE PROBLEM

The Observatory of Employability and Employment faces a series of challenges and considerations to perform its work and carry out its mission.

The first challenge faced to become an information source reference are the design of data collection and data processing procedures. All the organization's purposes and products are backed by data, so data can be seen as the backbone of the Observatory. However, the possession and storage of large volumes of data are not enough to provide important outcomes. Data Analytics techniques are necessary to make these datasets valuable and to reach insights (LaValle, Lesser, Shockley, Hopkins, & Kruschwitz, 2011) regarding the employment landscape in Spain (in this case). In brief, the Observatory needs to be able to properly support the data collection and its analysis, i.e. having efficient and automatic gathering and computation methods that enable people to make evidence-based decisions regarding employability and employment.

Nevertheless, having large volumes of data brings great responsibilities, even more if the data is considered as sensitive. The main sources of information are Spanish universities and their students; and universities, as data providers and customers of the insights provided by the Observatory, rely on the organization to keep their (and their students') data safe and anonymized. If the Observatory does not meet these conditions, universities will not trust in its projects and they will not want to be involved in them (potentially leading to the project's death by *data-inanition*).

Considering these conditions, data should be collected and stored complying the Spanish Personal Data Protection Act – generally referred to as LOPD (Ley Orgánica 15/1999, 1999). Regarding the anonymization and the Spanish law, the Observatory cannot allow the identification of individuals and universities behind the results of the study, as they will be exposed to the general audience. Anonymization and encryption techniques are needed to stick to these requirements.

In addition to the foregoing, the amount of data managed by the Observatory follows a growth trend. Nowadays, it handles more than 700 variables related more that 180000 students (these amounts are an addition of the variables and students involved in the first and second study performed by the Observatory). This is a challenge to consider, because the success of the project and the organization itself mainly remains in the capacity of handling this information in a proper way. The Observatory should keep developing its work without became flooded by the amount of data held, and this require high levels of scalability. Scalability could be a difficult issue to manage in any organization or (eco)system, but the benefits in the medium-long term are worth it.

Finally, the Observatory needs to spread the results once completed the different studies carried out. Given the importance of the employment and employability fields for the society, it is necessary to show and exploit the knowledge produced by the organization's activities in a proper way to reach all kind of audiences and stakeholders. This knowledge must be disseminated to anyone that aims to understand the Observatory's mission as well as its studies' results. Visualization and interaction methods can help solving this challenge, as it will be shown below.

Moreover, not only the society, but another similar organizations could take advantage of the Observatory results and knowledge bank (the knowledge and extra information that support the Observatory's activities). In this case, the communication methods for this kind of outcomes are also essential. These issues will be addressed in a deeper way in the following sections.

3. TRANSITION TO A DATA-DRIVEN TECHNOLOGICAL ECOSYSTEM

3.1 The Transition Itself

The Observatory owns a significant amount of information needed to achieve the goals of analyzing and understanding the factors involved in to became more employable (employability) or in the fact of get a job (employment). Regarding that, it is necessary to build the Observatory's system focusing on the effective use of large volumes of data. Not only that, but also being capable of spread and disseminate the collected information and knowledge among the system's users as well as to other external systems, so they can interact with the information collected and processed by the Observatory to reach deeper insights about employability and employment. It is important to consider that these users will have different requirements and the Observatory needs to support all of them.

After the first implementation of the Observatory's information system in 2015, its main features and functionality were the following (Michavila et al., 2015):

- **Data Collector:** This feature allowed the data gathering from different sources. The two main information sources are described below:
  - **Universities:** The participant universities provided administrative data about their students. To accomplish the storage of the administrative data, the system allowed the universities to upload CSV files containing the information required. When uploaded, the files were validated -in a semi-automatic way- by the Observatory's staff before its definitive storage, fixing the possible errors incorporated in the initial files.
  - **Students:** The system included custom questionnaires based on the administrative data collected from the universities. The questionnaires were the method used to collect employability and employment information about the graduates involved in the study. The answers given by the students were stored in the system and completed the universities' administrative data previously described, as well as the Observatory's knowledge bank, obtaining a larger and more interesting dataset to analyze.

- **Data Analyzer:** The main goal of this functionality was to help the analysis processes. It provided automated methods to obtain results from the gathered data. The data analyzer retrieved records from the persistent storage and processed (in a batch way) it to obtain simple statistics, derived variables and other results.
- **Interoperability:** The first implementation of the system was not prepared to support high interoperability levels between its components. On the other hand, there were neither methods to exchange information between the Observatory's system and other information systems, as it was not a priority at that time.
- **Information Dashboards:** Other main feature of the Observatory's system, supported by the data collector and the data analyzer, was the implementation of information dashboards. These dashboards allow the different users of the system to have a broad view of the results of the study through information visualizations. To make this possible, the dashboards were implemented following some basic Visual Analytics and Data Visualization principles, concepts and techniques (Keim et al., 2008; Keim, Kohlhammer, Ellis, & Mansmann, 2010). On the other hand, universities also had access to a basic dashboard with information tables to keep track of their own student's participation and other basic results.

These features satisfied the requirements at that moment (2015), giving the system the capacities to support the first edition of the Observatory's study with a simple administrative data collector and customized questionnaires for the students, as well as dashboards to show the general results of the study.

One of the principal issues were the performance of the data analysis, which made the user experience less satisfactory. Although pre-calculated statistics were used to reduce the response times in the public portal and dashboards, this solution was not flexible and, even more, it was not scalable.

As explained before, the public portal held a significant amount of visualizations representing the general results of the study, which provided general employment and employability information to anyone who accessed the results' section. However, universities were not able to have these visualizations and individual results applied to their own administrative and graduates' data. They only had the possibility of downloading the data from the system and analyze it on their own. This was an important issue and a valuable feature to be added in the future, but the performance of the data analyzer and the low levels of interoperability between the system's components made it not an easily addressable challenge.

The absence of high levels of interoperability was also a problem due to the organization's mission of analyze the data gathered and share the produced knowledge to external stakeholders. Increasing these interoperability levels could allow other information systems to take advantage of the Observatory's data, creating a collaborative knowledge environment and reporting more benefits to the project and society.

After analyzing all the issues found behind the first implementation of the Observatory's system, it was clear that the system must be upgraded to prepare the environment for future studies and functionalities. This was when the transition to a data-driven ecosystem began, considering the similarities between the Observatory's requirements and the problems addressed by this approach.

It is important to consider the challenges and priorities behind the construction of a digital ecosystem focused in knowledge discovery through data analysis and exploitation in order to obtain valuable insights from the domain data (Chajri & Fakir, 2014; Touya & Fakir, 2014). These priorities especially involve the interoperability between the series of components within the ecosystem, as well as the flexibility to be used by different stakeholders, and the scalability regarding to data management and data analysis (Garcia-Peñalvo & Garcia-Holgado, 2016). As referenced before, if high levels of interoperability are achieved, the components could able to evolve individually and collectively, and they even could add or change their own functionalities without affecting the rest of the components. Moreover, additional components, both internal or external, could be connected to build up a broader technological ecosystem without major difficulties in a transparent way (García-Holgado & García-Peñalvo, 2014a, 2014b).

Those characteristics met the Observatory's philosophy; employment and university employability fields are constantly evolving, and the Observatory needs to adapt their studies to get the maximum amount of information and analyze it in the best possible way. The Observatory needs to collect, process, analyze and present data, and in order to support those activities, as presented above, a data collector, data analyzer and data dashboards functionalities have been implemented through a series of software components. These components consume information from the persistent storage of the Observatory and also consume data generated by each one of them. It is clear that every component should be independent and need to have well-defined tasks, but also they must collaborate to achieve the goals of the whole system. Information flows between the components are important in order to reach the collaboration of all them, and promoting high interoperability levels is a common solution to decouple the components while they continue collaborating in a transparent way.

For all these reasons, it has been concluded that the data-driven ecosystem approach would report very important benefits to the Observatory system purposes. By implementing a data-driven ecosystem, the Observatory's technological environment can evolve following new requirements and providing new functionalities or services by the addition or modification of components, thanks to the high cohesion of the components but also to their loose coupling.

Taking all these points into account, the system has been upgraded in order to begin a transition to a data-driven ecosystem and accomplish the Observatory's new requirements due next studies' editions, as well as to fix other issues identified after the 2015 study and with the authors' previous work in software architectures that gather and manage data in different contexts (Cruz-Benito, Borrás-Gené, García-Peñalvo, Fidalgo Blanco, & Therón, 2015; Cruz-Benito, García-Peñalvo, et al., 2014; J. Cruz-Benito et al., 2016; Cruz-Benito, Therón, et al., 2014; Cruz-Benito, Therón, & García-Peñalvo, 2016; Cruz-Benito, Therón, García-Peñalvo, & Pizarro Lucas, 2015; García-Peñalvo, Cruz-Benito, Maderuelo, Pérez-Blanco, & Martín-Suárez, 2014; García-Sánchez, Cruz-Benito, Therón, & Gómez-Isla, 2015).

This upgrade took place during the development and beginning of the second study conducted by the Observatory (2016-2017). Among the different feature additions and modifications, the following could be highlighted:

- **Data Collector:** The component regarding the data collection has been upgraded to solve the issues encountered after the first implementation of the system. Despite the technical changes, the information sources have not varied:
    - **Universities:** As in the 2015 edition, the data gathering starts with universities sending its administrative information. One of the issues of that first edition was the need of manual validations by the Observatory's staff before considering the data as correct. This manual validation procedure slowed down the gathering process, since it was not automatized and did not give guarantees of the proper correction of all the errors found. For that reason, the administrative data collector has been upgraded to include a fully-automated data validation stage before the information is persistently stored into the system. The validation stage avoids manual checking procedures by analyzing the CSV files just after the universities have uploaded it, and presenting them a detailed report containing the exceptions and its

explanations. Universities are responsible of fixing the errors found and upload their new and corrected files afterwards. Once the system receives a file without any errors, the data passes another stage of cleaning and structuring to keep the Observatory's bank of information organized, as it will be briefly explained in the next subsection.
    - **Students:** The student's data collector has also experienced an upgrade. Rather than fix issues from the first edition, some changes had been added in order to make the current component more valuable. The data is collected (again) through questionnaires, but one of the main upgrades includes the improvement of the questionnaires themselves by creating different versions that will be displayed depending on the student that fills the questionnaire (some kind of adaptivity), in pursuance of an increase of the completion rate. Another upgrade involves the collection of variables generated from the student's interaction with the questionnaires (technically referred to as *paradata* (Stieger & Reips, 2010). These adaptive questionnaires themselves could lead to future research about this topic (research using A/B methodologies, etc.) (Cruz-Benito et al., 2017). As outlined in the previous bullet, the students' answers are also structured and cleaned before saving it permanently.
- **Data Analyzer:** The data analyzer has been modified to increase its performance and support more complex operations. The data retrieval process is made through the same methods as in the old version, against the Observatory's database, excepting the data calculations are performed on-demand (and not in a batch way) due to the increased efficiency through an in-memory analysis approach. Once the data is retrieved, it is loaded into memory, where the processing and analysis take place. In-memory analysis allows the datasets to be processed more efficiently, reporting significant increases of performance (Zhang, Chen, Ooi, Tan, & Zhang, 2015). This enhancement opens the capacities of the whole system, as most components could consume information generated by the data analyzer.
- **Interoperability:** A data-driven ecosystem should be able to exchange and exploit data along the components that form part of it. Interoperability reports several benefits to the Observatory's system and improve scalability, communication and collaboration between its elements. To achieve these goals, an interoperability component has been introduced to the system. This component was implemented by the incorporation of a REST (Fielding & Taylor, 2002) API (Application Programming Interface). Different data flows can be found within (and against) the system, so the API is designed to contemplate and give support to all of them. The design of this API brings to the Observatory's database isolation properties, as transactions won't interfere with each other considering the secure behavior of the API requests. This component allows the data to be available for users and other components through network, making a data-as-a-service approach possible (Terzo et al., 2013). It also allows the creation of methods for the users to download their entire datasets in different open format files (like CSV, etc.) on-demand.
- **Information Dashboards:** In general, the system's information dashboards and visualizations are the main beneficiaries of the transition to a data-driven ecosystem. The poor performance level of the older version of the system led to difficulties regarding the provision of individual dashboards to each participant university. The enhancement of the communication between components and the improvement of the data analyzer made possible the sending of real-time data to the information dashboards (as the data is retrieved and processed on demand). In addition, the good performance of this procedure makes the user experience more satisfying.

The backbone of the transition from a monolithic system to a data-driven ecosystem is the interoperability component, not only because it can be seen as a nexus between several information (eco)systems, but also because it made possible the scalability of the information dashboards and other components of the system. This interoperability component enables the information dashboards to consume processed data from the data analyzer component, decoupling them from other system's components and supporting the data-as-a-service approach. Having high decoupling levels facilitates the evolution of every component individually, without interfering with each other, but also collectively, meeting the characteristics of a technological ecosystem (García-Holgado & García-Peñalvo, 2014a, 2014b; García-Holgado & García-Peñalvo, 2014a, 2014b).

All the changes and functionalities introduced in this transition are resumed in *Table 1*, where the 2015 information system components and characteristics are compared with the 2017 data-driven technological ecosystem.

Finally, Figure 1 shows the conceptual view of the Observatory's data-driven ecosystem.

To complete the transition overview, the next subsections briefly describe the data structuration referenced before, the system's users, and the different technologies involved in the new ecosystem.

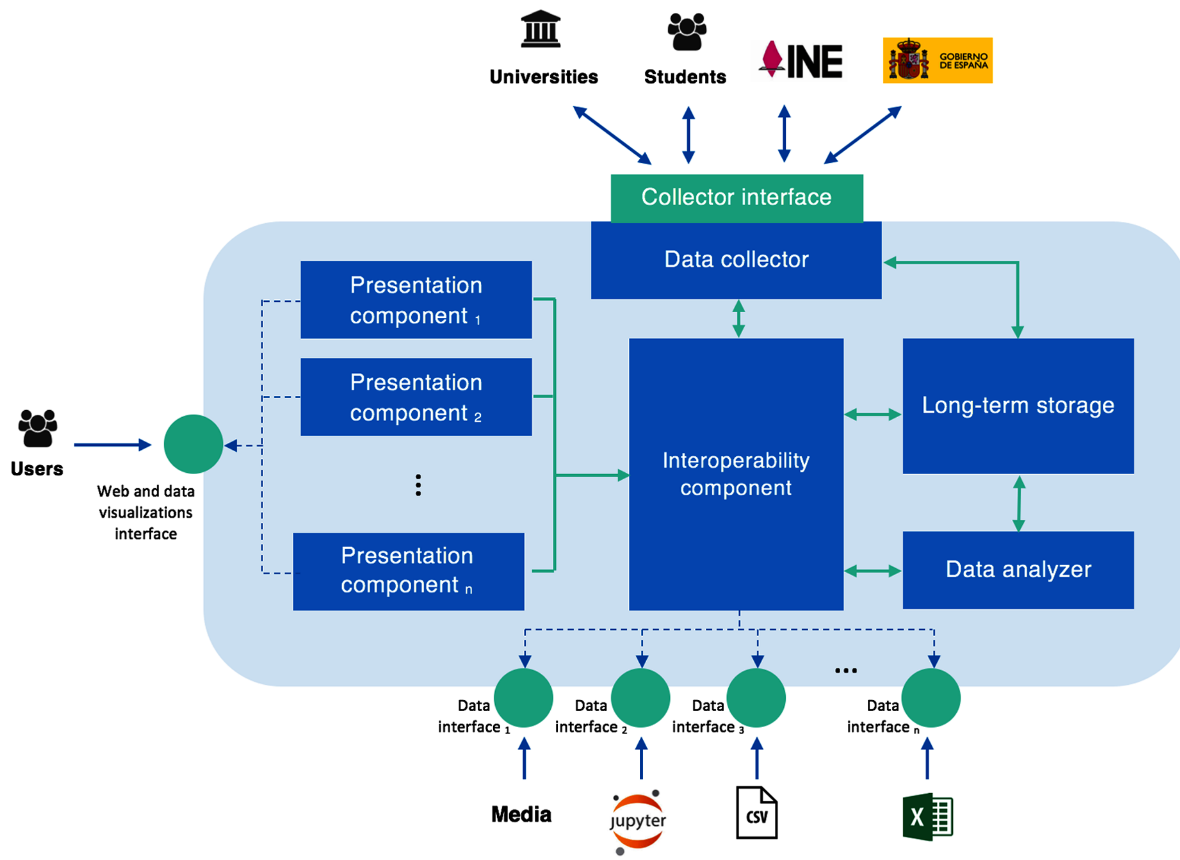
3.2 Data Structuration

Considering that the data is the driver element of the system, it is required to organize and structure it to perform its processing and analysis properly, reporting benefits for all the Observatory consumers. This structuration procedure is transparent for the users and it takes place at the system's backend.

Table 1. Summary of changes in the transition from a standard information system to a data-driven technological ecosystem

	OEEU's 2015 Information System	OEEU's 2017 Data-Driven Ecosystem
Data collector	1. Procedures to collect administrative data from universities. 2. Subsystem to generate questionnaires and collect data from the students' answers.	1. New stages in the gathering procedures: on-demand data validation and data cleaning. 2. Subsystem to generate questionnaires and different versions of them that could be displayed depending on the student profile that participates (supporting A/B testing). 3. Collection of *paradata* from the students involved in the study.
Data analyzer	1. Low levels of performance 2. Simple analysis	1. Increase of performance due to the in-memory data computation 2. Support of more complex analysis
Interoperability	Not contemplated in requirements.	Addition of an interoperability component to communicate the different components within the ecosystem and external systems or users through data flows, following the data-as-a-service approach.
Information dashboards	1. Visualizations for general results in the public website and personalized data tables for each universities' results in the system intranet. 2. Pre-calculated results. 3. Coupling between presentation and analysis components.	1. Adaptation of the visualizations and general dashboards to show each universities' results in their private dashboards. 2. Results calculated on-demand based on the request. 3. Higher levels of decoupling due to the interoperability component developed.

Figure 1. OEEU's ecosystem's components and relations



The principles followed to accomplish the data organization and structuration are technically named *tidy data* principles (Wickham, 2014). The Observatory's bank of information is organized in such a manner that, as Wickham points out, each variable forms a column, each observation forms a row and each type of observational unit forms a table. The raw data gathered from the different information sources is cleaned before its storage into the system's persistence layer sticking to these rules.

The principal benefit is the avoidance of messy datasets and the need of additional computations each time information is requested. For that last reason, tidy datasets also report performance gains as these unnecessary pre-processing operations are omitted.

3.3 System Users

There are a series of users of the Observatory's system, categorized by their needs. These users can be common users of the Observatory's public website (general audience), the graduates involved in the Observatory's studies, the Spanish universities that participate in the data gathering and consuming the knowledge and information generated, the data analysts (both external or internal to the organization) and other information systems that take advantage of the Observatory's components and information.

3.4 Technologies Used

3.4.1 Core System

The system has been implemented on Django, a Python web framework (Django Software Foundation, 2015; Holovaty & Kaplan-Moss, 2009). The election of Django has been backed up by the simplicity, flexibility and the community behind this framework.

On the other hand, the API has been built with the toolkit Django REST Framework (http://www.django-rest-framework.org/) and one of its variants: Django REST Pandas (https://github.com/wq/django-rest-pandas), which allows the API to serve Pandas dataframes (McKinney, 2011) to the client side. These frameworks meet the requirements of the project as they keep simplicity and good performance levels and provide a series of authorization and authentication methods for limiting and securing the access to the API, which is composed by different layers to keep the data model, the security and the logic of the system separate from each other.

3.4.2 Data Persistence

The long persistence technology used is MariaDB (https://mariadb.com/), a database management system derived from MySQL. With this technology, the main goals to achieve are to keep all of the Observatory's data safe, structured and well-organized, as well as obtain satisfactory response times on query processes.

3.4.3 Data Processing and Analysis

As already outlined above, the data analysis and processing take place at the backend of the system. These backend computations are made through Python Pandas (McKinney, 2011) because it allows in memory column analysis, reporting huge performance benefits.

All the computed data served by the Observatory's new engine is processed before sending it to the user who requested it, avoiding additional computational load on the user's side. Using this method, the system's information dashboards can show real-time data, unlike in the previous version of the system, where the statistics were pre-calculated, leading to flexibility and scalability problems.

3.4.4 Data Visualizations

The information dashboards have been built using D3, a Javascript library for visualizing data (Bostock, Ogievetsky, & Heer, 2011). D3 provide the appropriate methods to visualize the Observatory's bank of information, accomplishing the requirements and necessities of the users.

3.4.5 Data Exploitation

The interoperability component makes easy the external data retrieval and data exploitation through different tools independent of the system.

For example, data analysts can retrieve information from the Observatory's API and process the raw data using external tools like Jupyter notebooks (Kluyver et al., 2016), which could be very useful for publishing results, considering their capacity to show live code and outcomes even without being executed and the reproducibility feature it provides regarding to data analysis processes. These notebooks take advantage of the Observatory's API, facilitating the data retrieval process and hiding details of the system's persistent storage.

The ecosystem also has methods to allow universities to retrieve their students' data in CSV format, in order to use the downloaded files in other external tools like IBM SPSS, Microsoft Excel, etc.

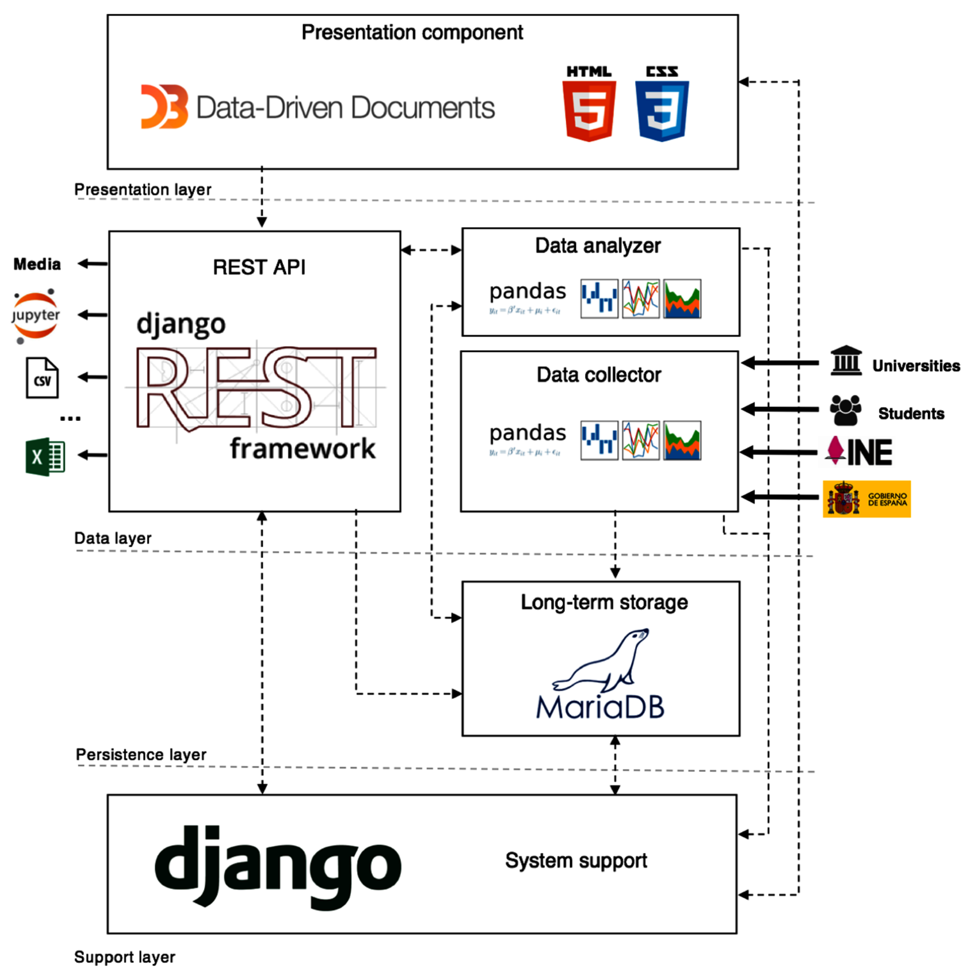
The technologies involved in the Observatory's ecosystem are summarized in Figure 2.

4. RESULTS

4.1 Observatory's Results

The Observatory for University Employability and Employment has been running since 2013, and started to collect degree student's information between the end of 2014 and the beginning of 2015, merging both

Figure 2. Summary of components and technologies that compose the Observatory's data-driven ecosystem



universities' administrative data and students' information collected through questionnaires. This data gathering corresponds to students that graduated in bachelor level in the 2009-2010 course.

At the end of 2016, a new period for data collection started, beginning with universities sending administrative information about students who finished their master's degree in the 2013-2014 academic year. As well as in the 2015 study, another data gathering process through questionnaires began in the early months of 2017, complementing the administrative information from universities.

The following results were reached during the Observatory's 2015 study (Michavila et al., 2016):

- The Observatory involved 49 Spanish universities, both public and private ones.
- The administrative data provided from all of these 49 universities reached at the end of the data collection stage 134,129 records about graduated students in the 2009-2010 season. Information about gender, sex, nationality, parents' studies, place where students live, average score in studies, mobility, internships, etc. are included among the total of data variables given (35 altogether).
- Moreover, of these 134,129 students, 13,006 started to answer their questionnaires, which is the 9,70% of the total registered students expressed as a percentage. Finally, 9,617 of them finished it (7,17% of the students in this case), giving a finalization rate of the questionnaires of 73,94%. The information provided by the completed questionnaires include around 400 variables regarding studies and employment. These variables keep information about the type of employment, relation with their jobs and their studies, wage, competencies, other studies done, opinions about the university and teaching methods, ratings about their studies, languages spoken, etc.
- Also, the Observatory keeps a knowledge bank with information related to Spanish degrees, information about branches of knowledge, as well as countries' data (economic, demographic, etc.), fields of employment and other relevant information about companies or employers. This information has been organized by the Observatory and can be useful to reach wider results data analysis.

On the other hand, the Observatory's 2017 study has reached other promising results during these last months of data gathering:

- In this case, The Observatory has involved 51 Spanish universities so far, also public and private. The difference on the number of participants between the two studies has its explanation in the fact that some of the universities that collaborated in the first study had decided not to participate in this last one and others that did not participate were enrolled in this new study. In the new study, universities added administrative data about its own master's graduated students.
- This time, the administrative data provided from these 51 universities involved 47,822 records about master students.
- Of these 47,822 students, approximately 6700 started questionnaires and 5200 finished them, making the questionnaire's finalization rate reach the more than 77%. The completion of the master's degree (MsC) questionnaires provides more than 200 variables related, again, ranging from information about their jobs, studies, competencies and other employment and university studies attributes.
- As in the 2015 study, variables given by the Observatory knowledge bank are used to allow deeper data analysis.

4.2 Results Regarding the System's Evolution

Regarding the results of the Observatory's information ecosystem itself, including the transition to a more powerful architecture supported by the concept of digital ecosystems and data-driven culture, the technological achievements reached are remarkable:

- The Observatory provides tools to help the universities to send its administrative information, as well as validation methods that give real-time feedback about possible errors within uploaded files.
- The Observatory has a system to generate personalized questionnaires for the involved students. The personalization is based on the student responses and profile. The personalization goes from the questionnaires' style to its interaction methods, among other changes. It also has procedures to gather information about the students' interaction with the questionnaires that could lead to future researches about usability and adaptivity.
- The Observatory has implemented components to analyze the collected data on-demand, achieving low response times even when computing large datasets.
- Finally, regarding the data presentation, the Observatory has a system for data visualization that handles the information for different stakeholders (universities, students or staff from the Observatory itself, among others). These presentations vary depending on the stakeholder.
    - The staff from the Observatory can access to all the data collected in order to obtain general results, as well as to manage individual universities' results or to monitor the questionnaires participation data.
    - Universities have access to personalized dashboards with information tables and visualizations adapted to its own data. This allow universities to consult effortless their results based on its administrative information and its students' answers.
    - The Observatory also provides public websites (http://datos.oeeu.org) that show the different studies overall results in an open way for any user. The websites make possible to disseminate all the knowledge discovered by the system.

The Figure 3 and Figure 4 show examples of the main Observatory's dashboards and its data visualizations, respectively. As explained before, these dashboards are also available for each participant University to check their own results.

5. DISCUSSION

Due to the upgrade of the Observatory's system, several issues associated to the complexity of the goals and technical challenges have been solved.

The Observatory's system has experienced an important performance increase in server's response, data analysis and data presentation, as this last one feature consumes data from the data analyzer. Since the nature of the project and the Observatory itself implies a continuous growth of the data amount, this was a priority issue to be resolved to avoid future problems regarding response and computation times.

Figure 3. Example of the administrative dashboard for the Observatory's staff users (contents in Spanish)

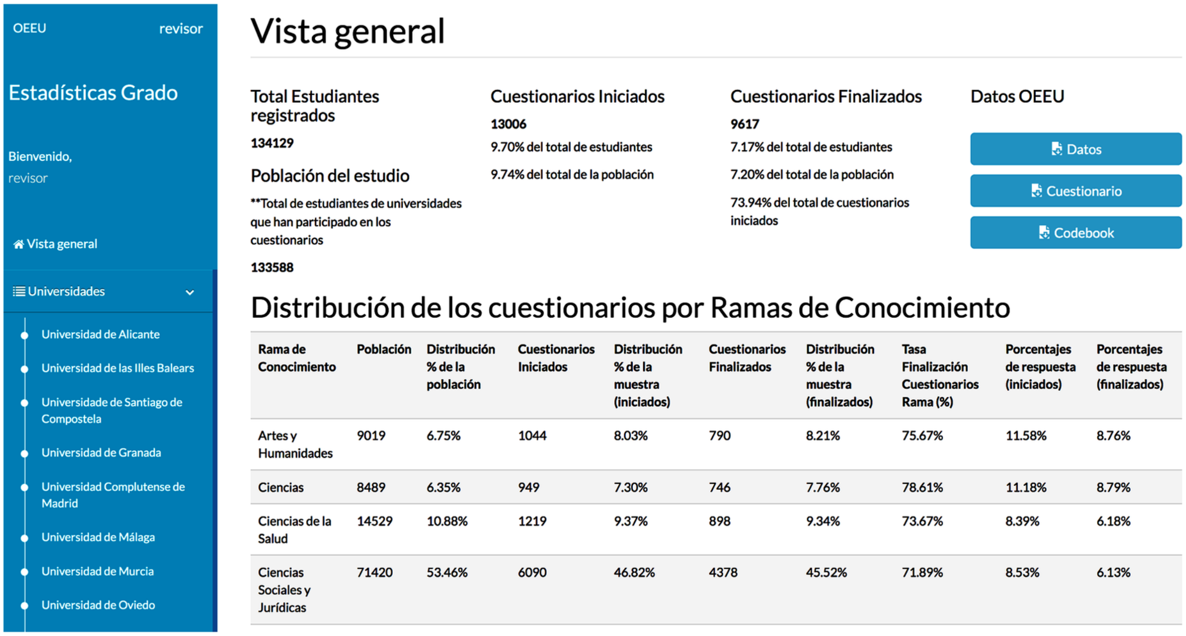


Figure 4. Example of information dashboards for the Observatory's data (contents in Spanish)

On the other hand, the data validation stage in the administrative data collector has provided great results, accelerating the first phase of the study and giving real-time feedback to the universities about the validity of their data. Before introducing an automatic data validator, manual fixes were essential and critical, because erroneous data could introduce noise in the studies. This manual validation consumed notable human and time resources from the Observatory, and it even could not be enough to fix all errors. Having an automatic validation stage, the errors are showed directly to the universities, making them responsible of the fixes and unburdening the Observatory from this task.

However, the main results come from the increase of interoperability of the system and the transition to a data-driven ecosystem. As it was introduced at the beginning of this chapter, the Observatory deals with high amounts of data coming from different sources, being the data the backbone of the system and the basis to make evidence-based decisions. The components within the system need to communicate with each other in a proper way to exchange the results of their tasks, but they also need to maintain high levels of independence, following technological ecosystems' guidelines and creating a data-as-a-service environment.

The data-driven ecosystem implemented allow software components to evolve individually and collectively, that made possible the addition of new functionalities and features without major difficulties. It reports remarkable benefits regarding communication between components and it reduces integration issues. Other of the results given by the interoperability and independence of the components is the enhancement of the information dashboards and the communication with third-party tools and systems. Data flows between visualizations and other components of the ecosystem enable dashboards to show more complex information, and even filter it depending on the user.

Although all the problems encountered after the first edition of the study have been covered, the needs of the Observatory are not static, and they are evolving continuously. The data-driven ecosystem implemented is a good basis to begin addressing more ambitious challenges. The decoupling of the components, especially the decoupling between the data analyzer and the dashboards opens many doors regarding information visualizations. Regarding that, some next challenges could include the evolution of the API responses to a graph approach for complex data (Vázquez-Ingelmo, Cruz-Benito, & García-Peñalvo, 2017), like it is being developed by main organizations as Facebook (i.e. with the GraphQL project https://facebook.github.io/graphql/).

One of the next challenges for the medium- or long-term is to develop linked data views for the visualization on the information dashboards. The philosophy behind linked visualizations lies in the creation of several simpler views (instead of creating one complex view) and their linkage, so that when the user interacts with one view, the other views will update and show the results of such an interaction (Wills, 2008). The Observatory owns a lot of information collected in each study, and it is difficult to reach valuable insights without the support reached through visualizations and interaction. Linked data views would make easier for the users to understand the Observatory's results. Also, could not be discarded the addition of complex data visualizations for the most complex problems regarding the information kept by the Observatory (multidimensional problems, etc.).

The data-driven ecosystem approach also allows the dissemination of information and knowledge gained by the Observatory among other external systems. Media, journals, data analysts etc., could link their own systems or tools to the Observatory's ecosystem to obtain results and insights of employment and university employability.

6. CONCLUSION

The Spanish Observatory for University Employability and Employment is generating, analyzing and disseminating information about the employability and employment of university graduates in Spain through a unified methodology. To support the data collection, as well as its analysis and dissemination, an information system was built in 2015 to accomplish the Observatory's goals and technological requirements. However, giving the challenges of the Observatory regarding its mission and information requirements, the traditional information system built has been transformed into a data-driven ecosystem. As it was presented throughout this chapter, this data-driven technological ecosystem presents a collaborative environment between software components and users that supports adequately the Observatory's vision and mission.

Technological ecosystems empower this kind of collaborative environments, enabling the connection of different components through data flows, resulting in a series of individual components with well-defined task working together to achieve the ecosystem's goals. They also allow the modification and update of components without affecting the rest of the ecosystem's elements, and makes easier the creation of new data flows, both internal and external to the ecosystem. On the other hand, data-driven philosophy focuses the attention on data in order to support decision making.

The combination of these two concepts make a data-driven ecosystem a powerful and suitable approach for the Observatory. This combination matches well the data-as-a-service approach implemented in the project, due to the interoperability, independence and decoupling gained through the transition to a data-driven ecosystem, allowing data to be served on-demand to different components and users.

With its data-driven ecosystem, the Observatory is now able to discover and understand better the variables that influence the employment and employability in graduate students. It also promotes the creation of data flows to disseminate the knowledge and wisdom gained during the analysis to internal and external users, different stakeholders and third-party information systems or tools, as well as it supports the evolution of its own components to meet new requirements.

ACKNOWLEDGMENT

The research leading to these results has received funding from "la Caixa" Foundation. Also, the author Juan Cruz-Benito would like to thank the European Social Fund and the *Consejería de Educación* of the *Junta de Castilla y León* (Spain) for funding his predoctoral fellow contract. This work has been partially funded by the Spanish Government Ministry of Economy and Competitiveness throughout the DEFINES project (Ref. TIN2016-80172-R).

REFERENCES

Bostock, M., Ogievetsky, V., & Heer, J. (2011). D^3 data-driven documents. *IEEE Transactions on Visualization and Computer Graphics*, *17*(12), 2301–2309. doi:10.1109/TVCG.2011.185 PMID:22034350

Chajri, M., & Fakir, M. (2014). Application of Data Mining in e-Commerce. *Journal of Information Technology Research*, *7*(4), 79–91. doi:10.4018/jitr.2014100106

Cruz-Benito, J., Borrás-Gené, O., García-Peñalvo, F. J., Fidalgo Blanco, Á., & Therón, R. (2015). Extending MOOC ecosystems using web services and software architectures *Proceedings of the XVI International Conference on Human Computer Interaction* (pp. 1-7). Vilanova i la Geltrú, Spain: ACM. doi:10.1145/2829875.2829923

Cruz-Benito, J., García-Peñalvo, F. J., & Therón, R. (2014). *Defining generic data collectors for Learning Analytics: Facing up the heterogeneous data from heterogeneous environments*. Paper presented at the International Conference on Advanced Learning Technologies (ICALT) 2014, Athens, Greece. doi:10.1109/ICALT.2014.108

Cruz-Benito, J., Maderuelo, C., García-Peñalvo, F. J., Therón, R., Pérez-Blanco, J. S., Gómez, H. Z., & Martín-Suárez, A. (2016). Usalpharma: A Software Architecture to Support Learning in Virtual Worlds. *IEEE Revista Iberoamericana de Tecnologias del Aprendizaje*, *11*(3), 194–204. doi:10.1109/RITA.2016.2589719

Cruz-Benito, J., Therón, R., & García-Peñalvo, F. J. (2016). Software Architectures Supporting Human-Computer Interaction Analysis: A Literature Review. In P. Zaphiris & A. Ioannou (Eds.), *Learning and Collaboration Technologies: Third International Conference, LCT 2016, Held as Part of HCI International 2016, Toronto, ON, Canada, July 17-22, 2016, Proceedings* (pp. 125-136). Cham: Springer International Publishing. doi:10.1007/978-3-319-39483-1_12

Cruz-Benito, J., Therón, R., García Peñalvo, F. J., Maderuelo, C., Pérez-Blanco, J. S., Zazo, H., & Martin-Suarez, A. (2014). Monitoring and feedback of Learning Processes in Virtual Worlds through analytics architectures: A real case. In Á. Rocha, D. Fonseca, E. Redondo, L. P. Reis, & M. P. Cota (Eds.), Sistemas y Tecnologías de Información. Actas de la 9ª Conferencia Ibérica de Sistemas y Tecnologías de Información (Vol. I Artículos, pp. 1126-1131). Barcelona, España: AISTI (Asociación Ibérica de Sistemas y Tecnologías de Información). doi:10.1109/CISTI.2014.6877097

Cruz-Benito, J., Therón, R., García-Peñalvo, F. J., & Pizarro Lucas, E. (2015). Discovering usage behaviors and engagement in an Educational Virtual World. *Computers in Human Behavior*, *47*(0), 18–25. doi:10.1016/j.chb.2014.11.028

Cruz-Benito, J., Therón, R., García-Peñalvo, F. J., Sánchez-Prieto, J. C., Vázquez-Ingelmo, A., Martín-González, M., & Martínez, J. M. (2017). Improving Success/Completion Ratio in Large Surveys: A Proposal Based on Usability and Engagement. In P. Zaphiris & A. Ioannou (Eds.), *Learning and Collaboration Technologies. Technology in Education: 4th International Conference, LCT 2017, Held as Part of HCI International 2017, Vancouver, BC, Canada, July 9-14, 2017*, *Proceedings, Part II* (pp. 352-370). Cham: Springer International Publishing. doi:10.1007/978-3-319-58515-4_28

Django Software Foundation. (2015). *Django Web Framework*. Retrieved from https://www.django-project.com/

Fielding, R. T., & Taylor, R. N. (2002). Principled design of the modern Web architecture. *ACM Transactions on Internet Technology*, *2*(2), 115–150. doi:10.1145/514183.514185

García-Holgado, A., Cruz-Benito, J., & García-Peñalvo, F. J. (2015). *Analysis of Knowledge Management Experiences in Spanish Public Administration*. Paper presented at the Third International Conference on Technological Ecosystems for Enhancing Multiculturality (TEEM'15), Porto, Portugal. doi:10.1145/2808580.2808609

García-Holgado, A., & García-Peñalvo, F. J. (2014a). *Architectural pattern for the definition of eLearning ecosystems based on Open Source developments.* Paper presented at the Computers in Education (SIIE), 2014 International Symposium on. doi:10.1109/SIIE.2014.7017711

García-Holgado, A., & García-Peñalvo, F. J. (2014b). *Patrón arquitectónico para la definición de ecosistemas de eLearning basados en desarrollos open source*. Academic Press.

García-Holgado, A., García-Peñalvo, F. J., Hernández-García, Á., & Llorens-Largo, F. (2015). *Analysis and improvement of knowledge management processes in organizations using the Business Process Model Notation.* Paper presented at the Annual Conference of the Global Innovation and Knowledge Academy. doi:10.1007/978-3-319-22204-2_9

García-Peñalvo, F. J., Cruz-Benito, J., Maderuelo, C., Pérez-Blanco, J. S., & Martín-Suárez, A. (2014). Usalpharma: A Cloud-Based Architecture to Support Quality Assurance Training Processes in Health Area Using Virtual Worlds. *The Scientific World Journal*, *10*. doi:10.1155/2014/659364 PMID:24778593

Garcia-Peñalvo, F. J., & Garcia-Holgado, A. (2016). *Open Source Solutions for Knowledge Management and Technological Ecosystems*. IGI Global.

García-Sánchez, F., Cruz-Benito, J., Therón, R., & Gómez-Isla, J. (2015). *Designing and building systems and tools to analyze visual communications on social networks*. Paper presented at the Third International Conference on Technological Ecosystems for Enhancing Multiculturality (TEEM'15), Porto, Portugal. doi:10.1145/2808580.2808629

Holovaty, A., & Kaplan-Moss, J. (2009). *The definitive guide to Django: Web development done right*. Apress. doi:10.1007/978-1-4302-1937-8

Keim, D., Andrienko, G., Fekete, J.-D., Görg, C., Kohlhammer, J., & Melançon, G. (2008). Visual analytics: Definition, process, and challenges. *Information Visualization*, 154–175.

Keim, D., Kohlhammer, J., Ellis, G., & Mansmann, F. (2010). *Mastering the information age solving problems with visual analytics*. Eurographics Association.

Kluyver, T., Ragan-Kelley, B., Pérez, F., Granger, B., Bussonnier, M., Frederic, J., . . . Corlay, S. (2016). Jupyter Notebooks—a publishing format for reproducible computational workflows. *Positioning and Power in Academic Publishing: Players, Agents and Agendas*, 87.

LaValle, S., Lesser, E., Shockley, R., Hopkins, M. S., & Kruschwitz, N. (2011). Big data, analytics and the path from insights to value. *MIT Sloan Management Review*, *52*(2), 21.

Ley Orgánica 15/1999. (1999). Ley Orgánica 15/1999, de 13 de diciembre, de Protección de Datos de Carácter Personal. *Boletín Oficial Del Estado, 298*(2), 43088-43099.

McKinney, W. (2011). pandas: a foundational Python library for data analysis and statistics. *Python for High Performance and Scientific Computing*, 1-9.

Michavila, F., Martín-González, M., Martínez, J. M., García-Peñalvo, F. J., & Cruz-Benito, J. (2015). Analyzing the employability and employment factors of graduate students in Spain: The OEEU Information System. *Proceedings of the 3rd International Conference on Technological Ecosystems for Enhancing Multiculturality*. doi:10.1145/2808580.2808622

Michavila, F., Martínez, J. M., Martín-González, M., García-Peñalvo, F. J., & Cruz-Benito, J. (2016). *Barómetro de Empleabilidad y Empleo de los Universitarios en España, 2015 (Primer informe de resultados)*. Academic Press.

Patil, D. (2012). *Data Jujitsu*. O'Reilly Media, Inc.

Patil, D., & Mason, H. (2015). *Data Driven*. O'Reilly Media, Inc.

Peñalvo, F. J. G., Conde, M. J. R., Pardo, A. M. S., González, M. Á. C., Zangrando, V., & Holgado, A. G. (2012). GRIAL (GRupo de investigación en InterAcción y eLearning), USAL. *IE Comunicaciones: Revista Iberoamericana de Informática Educativa,* (15), 85-94.

Stieger, S., & Reips, U.-D. (2010). What are participants doing while filling in an online questionnaire: A paradata collection tool and an empirical study. *Computers in Human Behavior*, *26*(6), 1488–1495. doi:10.1016/j.chb.2010.05.013

Terzo, O., Ruiu, P., Bucci, E., & Xhafa, F. (2013). *Data as a service (DaaS) for sharing and processing of large data collections in the cloud.* Paper presented at the Complex, Intelligent, and Software Intensive Systems (CISIS), 2013 Seventh International Conference on.

Touya, K., & Fakir, M. (2014). Mining Students' Learning Behavior in Moodle System. *Journal of Information Technology Research*, *7*(4), 12–26. doi:10.4018/jitr.2014100102

Vázquez-Ingelmo, A., Cruz-Benito, J., & García-Peñalvo, F. J. (2017). *Improving the OEEU's data-driven technological ecosystem's interoperability with GraphQL.* Paper presented at the 5th International Conference on Technological Ecosystems for Enhancing Multiculturality, Cádiz, Spain.

Wickham, H. (2014). Tidy data. *Journal of Statistical Software*, *59*(10), 1–23. doi:10.18637/jss.v059.i10 PMID:26917999

Wills, G. (2008). Linked data views. Handbook of data visualization, 217-241.

Zeleny, M. (1987). Management support systems: Towards integrated knowledge management. *Human Systems Management*, *7*(1), 59–70.

Zhang, H., Chen, G., Ooi, B. C., Tan, K.-L., & Zhang, M. (2015). In-memory big data management and processing: A survey. *IEEE Transactions on Knowledge and Data Engineering*, *27*(7), 1920–1948. doi:10.1109/TKDE.2015.2427795

7.2 Appendix B. Improving the OEEU's data-driven technological ecosystem's interoperability with GraphQL

Improving the OEEU's data-driven technological ecosystem's interoperability with GraphQL

Andrea Vázquez-Ingelmo
GRIAL Research Group, Department of Computer and Automatics, University of Salamanca
Paseo de Canalejas 169, 37008 Salamanca, Spain
andreavazquez@usal.es

Juan Cruz-Benito
GRIAL Research Group, Department of Computer and Automatics, University of Salamanca
Paseo de Canalejas 169, 37008 Salamanca, Spain
juancb@usal.es

Francisco J. García-Peñalvo
GRIAL Research Group, Department of Computer and Automatics, University of Salamanca
Paseo de Canalejas 169, 37008 Salamanca, Spain
fgarcia@usal.es

ABSTRACT

A crucial part of data-driven ecosystems[1] is the management and processing of complex data structures, as well as the proper handling of the data flows within the ecosystem. To manage these data flows, data-driven ecosystems need high levels of interoperability, as it allows the collaboration and independence of both internal and external components. REST APIs are a common solution to achieve interoperability, but sometimes they lack flexibility and performance. The arising of GraphQL APIs as a flexible, fast and stable protocol for data fetching makes it an interesting approach for data-intensive and complex data-driven (eco)systems. This paper outlines the GraphQL protocol and the benefits derived from its use, as well as it presents a case of study of the improvement experienced by the Observatory of Employment and Employability (also known as *OEEU*) ecosystem after including GraphQL as main API in several components. The results of the paper show promising improvements regarding the flexibility, maintainability and performance, among other benefits.

CCS CONCEPTS

• **Software and its engineering → Interoperability • Information systems → Data Analytics • Information systems → RESTful web services**

KEYWORDS

GraphQL; API; Technological ecosystems; Data-driven; Interoperability

ACM Reference format:

A. Vázquez-Ingelmo, J. Cruz-Benito, and F. J. García-Peñalvo. 2017. Improving the OEEU's data-driven technological ecosystem's interoperability with GraphQL. In *Proceedings of 5th International Conference on Technological Ecosystems for Enhancing Multiculturality, TEEM 2017, Cádiz, Spain, October 18-20, 2017*, 8 pages.
https://doi.org/10.1145/3144826.3145437

1 INTRODUCTION

Data-driven [1, 2] technological ecosystems have to continually deal with complex data structures and different data flows in order to achieve the ecosystem's main purpose. These data-driven ecosystems need to rely on a collaborative technological environment made up of different components that gather, analyze and disseminate the problem's domain data [3].

Due to the existence of different and heterogeneous components with separated tasks within technological ecosystems, it is important having the support of communication methods and high levels of interoperability to reach the components' collaboration without losing their own independence [4].

The implementation of REST [5] APIs (Application Programming Interface) to retrieve, create or modify the ecosystem's data fosters high levels of interoperability by creating well-defined interfaces and endpoints for data transactions. REST APIs decouple data consumers from data sources, connecting them through data flows independently of their platforms or technical characteristics.

However, REST APIs - although well implemented - can present lack of flexibility. They must have endpoints for the data requests contemplated in the requirements. But requirements can evolve through time, as well as the domain data's structure and the ecosystem's components (or users), making it necessary to rewrite the REST API interfaces.

TEEM 2017, October 18–20, 2017, Cádiz, Spain

ACM ISBN 978-1-4503-5386-1/17/10…$15.00
https://doi.org/10.1145/3144826.3145437

Even if the REST APIs' interfaces evolve along with the requirements, some components (or users) might need only certain parts or fields of the whole (eco)system's data, or even a combination of fields that belong to different data objects, and find out that none of the implemented REST API's endpoints satisfies their specific request. In that case, the components (or users) need to make a request to the endpoint that returns the closest set of fields required or make a series of requests to different endpoints to gather all the data they need [6].

The previous scenario could be solved by implementing an endpoint for every possible data request, but, again, this will result in flexibility and maintainability issues.

This flexibility, maintainability and performance problems derived from REST APIs are what the GraphQL language is trying to solve. GraphQL (https://facebook.github.io/graphql/) is "a query language designed to build client applications by providing an intuitive and flexible syntax and system for describing their data requirements and interactions" [7].

The main goal of GraphQL is to unify the data requests of an (eco)system into one unique endpoint, reducing the number of requests needed to retrieve data. On the other hand, GraphQL allows components and users to specify in their own requests the particular fields they are asking for, retrieving solely what they need for their purposes.

In data-driven technological ecosystems, the components can individually and collectively evolve through time, but they need to keep collaborate in order to achieve the whole ecosystem's purpose [4]. That is why the features of GraphQL makes this query language suitable for data-driven technological ecosystems, as it provides interoperability (necessary for the components' communication and their data flows), flexibility (making smoother the ecosystem's continuous evolution) and also it improves performance by reducing the number and size (most of the times in broadband) of data requests.

The rest of this paper is structured as follow: the second section outlines de basics of the GraphQL protocol and its queries; the section 3 presents the data issues related to the Observatory of Employment and Employability's ecosystem and describes the APIs evolution from a REST API to a GraphQL API; section 4 shows the results and benefits resulting from the implementation of the GraphQL API in the Observatory's ecosystem. Finally, the fifth section discusses the problems addressed by the change, followed by the sixth and last section, where the conclusions reached are presented.

2 GRAPHQL BASICS

As explained before, GraphQL is a query language located in the application layer, which means that it can be used against any backend that meet the protocol's specification [7].

In GraphQL, "products are described in graphs and queries, instead of the REST notion of endpoints" [8]. This "graph nature" is what makes (eco)systems that implement GraphQL more flexible and scalable. The modification of the data objects enclosed in this new API paradigm is similar to the modification of a graph; the addition or removal of data fields is trivial because the GraphQL approach consider these operations as additions or removals of nodes in a graph, simplifying the modifications and evolution of the domain's data.

With the GraphQL approach, data instances are represented as a set of fields (nodes), which also can have nested fields and relations with other nodes. It is the responsibility of the GraphQL backend to specify the data objects and the fields that are available for the clients to query.

Keeping in mind the graph nature of the language and the relations between nodes, a GraphQL query is a set of hierarchical fields of data that a client wants to retrieve, and as a result, only that specified fields will be returned.

GraphQL queries are performed against a unique endpoint which unifies all possible data requests. That means that only one URL is needed to serve all the (eco)system's data available through the API [7]. This approach allows the clients to design their specific queries, and frees the (eco)systems and its developers from taking into account and implement endpoints for all the possible data requests combinations, resulting in an increase of flexibility and maintainability.

A simple GraphQL query example is presented in the Fig. 1. Querying it against a server that supports GraphQL (and holds information about people) would result in a response with the structure presented in the Fig. 2. These queries are sent to the target backend through an HTTP POST request, where the payload is the JSON-structured query specified by the client.

```
{
    person(id: some_id)
    {
        name,
        age,
        height,
        weight,
        parents
        {
            name
        }
    }
}
```

Figure 1: Example of a GraphQL query executed to retrieve information of a specific person.

```
{
    "person"
    {
        "name": "John",
        "age": 23,
        "height": 1.75,
        "weight": 63,
        "parents": [
            {
                "name": "Alice"
            },
            {
                "name": "Bob"
            }
        ]
    }
}
```

Figure 2: Example of a possible GraphQL query response with information of a specific person.

The GraphQL queries are client-specific [7], which means that the response's structure is led by the initial query's structure, and the results are returned in the same order as requested, providing a lot of freedom to the client.

If the clients' data requirements change at some point and they need to retrieve other fields, they simply have to modify their query's structure and send it to the same endpoint.

Another characteristic of GraphQL is that it is strong-typed; the types of the returned response's fields are specified at the backend. By this way, consumers are able to know the shape and nature of the responses.

GraphQL also provides mechanisms to filter results. As pointed out in the GraphQL specification, the query's fields can be seen as "functions" (implemented at the backend) that returns values; and as "functions", they can accept arguments to alter their behavior or the returned values themselves [7]. For instance, in the Fig. 1, "id" is an argument that tells the backend which particular person the client is requesting. Of course, there must be mechanisms implemented in the backend to filter the data objects by an identifier and, consequently, make the argument "id" available.

But this language not only provides a flexible and efficient manner to read data from backend sources, it also allows the four basic functions of CRUD: create, read, update and delete. In contrast with REST, where the CRUD functions are specified in the HTTP method (PUT, POST, GET and DELETE), GraphQL defines three types of operations [7] independent of the HTTP request:

- Queries: read-only retrieval (already explained).
- Mutation: a modification on the backend followed by a response with results.
- Subscriptions: requests in response to source events.

An example of a mutation can be seen in the Fig. 3. Once this mutation request arrives to the backend, a new data object will be created in the database with the fields provided (name and age in this example), and the client would receive a response with the identifier of the person created (or with any other field specified in the initial request). The backend is responsible of implementing the operations available (create, update, delete, etc.).

```
mutation {
  createPerson(input: {
    name: "Alan",
    age: 32,
  }) {
    id
  }
}
```

Figure 3: Example of a GraphQL mutation used to create a new person named "Alan" in the backend.

Subscriptions, on the other hand, are data responses that are sent to the clients subscribed when a particular event fires.

These are the basic characteristics and concepts behind GraphQL for the scope of this paper, for a wider view of this query language and all its possibilities, authors refer to its technical specification [7].

3 CASE OF STUDY: THE OEEU'S DATA-DRIVEN ECOSYSTEM

A GraphQL API has been implemented for the OEEU's data-driven ecosystem to test the benefits derived from the use of this language.

The first subsection introduces the Observatory of Employability and Employment and its data handling issues.

Subsections 3.2 and 3.3 outline the Observatory's REST API (and its limitations) and the Observatory's GraphQL API implemented to improve the collaboration between their data-driven ecosystem's components, respectively.

3.1 The OEEU's Problem

The Observatory of University Employability and Employment (also known as *OEEU* using the Spanish initials for *Observatorio de Empleabilidad y Empleo Universitarios*) http://oeeu.org/, is an organization formed by a group of researchers and technicians with the vision of becoming the information reference for understanding and exploiting knowledge about variables related to employability and university employment (and its behavior) [9, 10].

The organization itself is data-driven, as all its activities are backed up by data. To achieve its goals, the Observatory needs to manage significant amounts of data from different information sources, including Spanish universities and their students. But raw data is not enough to reach insights about employment and employability; all the collected data must be analyzed to allow evidence-based decision making.

With the evidences and knowledge gained by the analysis of the gathered data, the Observatory's studies can help universities to improve its policies, metrics and results regarding their students' employability, as well as their preparation and satisfaction with those important aspects. This analysis of organizational and students' data from universities frames the Observatory in the scope of areas like Academic Analytics [11] and knowledge management in the university scope [12, 13].

The Observatory's studies are supported by a data-driven technological ecosystem formed by a series of components responsible for collecting, storing, analyzing, disseminating and visualizing the domain's data [2, 14]. This ecosystem is based on the collaboration of different components to achieve the purposes of the Observatory.

However, employment and employability data is continuously increasing, and the issues related to them regarding the technological ecosystem needs to stick to evolving requirements while maintaining a collaborative environment.

The transition from a monolithic system to a data-driven ecosystem through the improvement of the interoperability levels between its components (by the implementation of a REST API) made the Observatory's technological support a powerful tool. But, as introduced before, REST APIs could not provide enough flexibility, scalability and performance levels for such changing study fields and technologies.

3.2 The OEEU's REST API

One of the main goals of the Observatory's ecosystem is to disseminate the knowledge gained by the analysis of the gathered data. To accomplish this goal, the Observatory's system includes different presentation and data analysis components that provide support to reach insights about graduates' employability.

The purpose of the presentation components in the ecosystem is to show information through tables and visualizations after the raw data has been analyzed by the data analyzer component. It is obvious that it was necessary to connect these components to make the analyzed information available on demand for its visualization. But not only the communication between these components was a concern; other components of the Observatory's ecosystem also needed communication methods to handle their data flows inputs and outputs.

Furthermore, the Observatory's vision of becoming an information source reference for employment and university employability needed communication methods to allow the connection of external components and to create a wider technological environment to gain more knowledge and wisdom through the collaboration of different information (eco)systems.

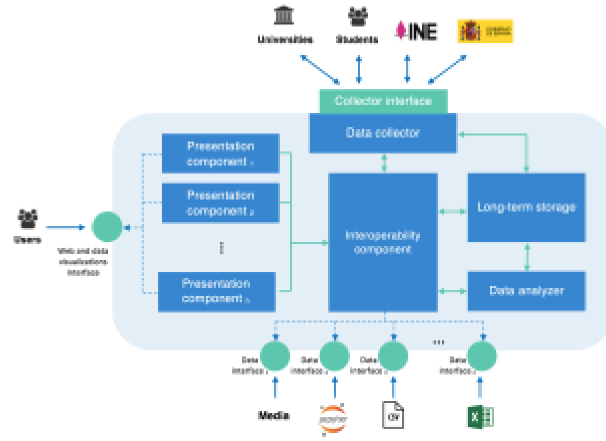


Figure 4: Overview of the OEEU's data-driven ecosystem after introducing the interoperability component (REST API).

As a consequence of these problems, the Observatory's introduced a REST API to handle the communication between both internal and external components. This REST API was designed according to the possible data flows and data requests happening within (and against) the ecosystem. With the REST API handling the interoperability, the components of the system were able to decouple from each other, converting the initial OEEU's monolithic information system into a data-driven technological ecosystem, as presented in the Fig. 4.

The REST API accomplished its goal and the ecosystem's interoperability improved, but the API stuck to the requirements at that time. This meant that any change on the data requirements or on the components of the ecosystem (i.e. when data analysts are prototyping new or different analysis or data approaches) would entail major changes on the REST API's endpoints (or even the creation of new versions of the REST API). Another limitation of the REST API was its performance. Due to the high number of metrics, variables, categories and information collections involved in the Observatory's data analysis, the number of API endpoints to retrieve them all was significant (and therefore the number of API calls needed to retrieve specific data). The REST API also lacked flexibility and reutilization, because the endpoints (their outputs) were adapted to the information needs of every component of the ecosystem. This design simplified the collaboration between the internal components, but also made it very difficult to reutilize the API endpoints for other tasks or even other types of components or software entities. Sometimes, if some component experienced a change, the whole endpoint designed for it had to be updated.

Looking to the future of this data-driven product and project, it was clear that the Observatory's REST API could evolve to a GraphQL API, since the previously outlined characteristics of this query language made it suitable and potentially beneficial for the Observatory's interoperability, scalability and maintainability needs.

3.3 The OEEU's GraphQL API

The GraphQL API for the Observatory's data-driven technological ecosystem has been implemented using Graphene (a GraphQL framework for Python, http://graphene-python.org/).

Currently, the Observatory's GraphQL API is used to create and handle the data flows within the ecosystem, and enables components to retrieve raw data about the students and statistics and metrics calculated on demand. The structure of the ecosystem remains the same (as in the Fig. 4), but now the interoperability component is based on GraphQL instead of REST.

Other significant characteristic introduced by the implementation of the GraphQL API is the fact that only one endpoint is needed (which means that only one URL is needed), against which all the queries are executed. Using the GraphQL API, components can retrieve data fields according to their data requirements. If that components' requirements change at some point, it is only necessary to modify the query of the particular component involved in the change.

While high levels of interoperability are maintained (as with the previous REST API), the independence of the components is even further enhanced. Moving the task of designing the query to the API consumer gives the clients more freedom to evolve without affecting other components.

Furthermore, the unification of the data and the structuration behind GraphQL makes possible the reutilization as well as the predictability of the results of the API calls (one of the characteristics of GraphQL [7]). This response predictability derives benefits regarding the reduction of time spending by clients parsing and restructuring data to fit their requirements.

However, although the whole data-driven ecosystem experienced an improvement, one of the main beneficiaries of the GraphQL API are the presentation components (mainly dashboards in this case), because these components are more likely to evolve and change through time. As explained in the previous subsection, visualizations are an essential tool for understanding the results of the Observatory's studies. There are a series of dashboards (supported by the ecosystem's presentation components) holding different visualizations that show analyzed data on demand. Those dashboards typically consumed data from several REST API endpoints to compose the different visualizations included in the whole screen.

With the GraphQL API providing the communication methods between the data analyzer and the presentation components, any change in the visualizations supported by any presentation component would only imply a change on the query that populates it with data. In contrast with the previous REST API, where any change on the visualizations would involve changes in the API endpoints, GraphQL promotes the decoupling and independence of the components, so any modification in a presentation component would only impact that presentation component (avoiding changes in the backend endpoints). This decoupling and independence characteristics also applies to the rest of the ecosystem's components that use the GraphQL API to handle their data flows. The other main beneficiaries of the GraphQL API are third-party components that can retrieve specific information efficiently without having to adapt themselves to the Observatory's particular endpoints' design.

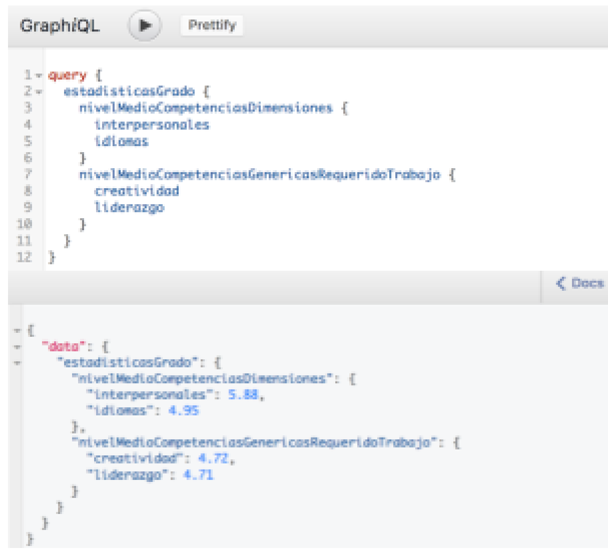


Figure 5: Example of a query (top) and a response (bottom) of The Observatory's GraphQL API (contents in Spanish).

Fig. 5 shows an example of a query to the Observatory's GraphQL API to retrieve statistics about the competencies level of the students involved in the Observatory's 2015 study (as well as the response). The example query was made via GraphiQL (https://github.com/graphql/graphiql), an interactive GraphQL IDE which allows users to explore all the possibilities of a GraphQL API. However, this query can be made through any other tool that is able to send POST requests.

4 RESULTS

The Observatory's data-driven ecosystem has mainly experienced an improvement of flexibility regarding the communication of both internal and external components, and also increased its scalability and maintainability levels, as well as the network performance. The main reason of the performance increase is the reduction of API calls to retrieve data. As pointed out before, only one endpoint is necessary, and thereby one API call is enough to query the backend.

For example, the presentation component (that manages information about the students' generic competencies levels) provides support for a total of 15 visualizations (it is currently the larger dashboard page of the Observatory's ecosystem). With the REST API as a data provider, this presentation component had to send 17 requests to different endpoints in order to retrieve all the analyzed data and visualize them. The number of API calls slowed down the data retrieval process. However, with GraphQL, a single query is issued against the backend to retrieve the same data for the entire dashboard. This supposed an increase of network performance (32.26% of average network time has been decreased) and a decrease regarding the requests' size (47.86% less of broadband loading is required).

The results summarized in Table 1 show the network average times (obtained through the network developer tool of the Mozilla Firefox browser) after 10 accesses to the generic competencies statistics dashboard and therefore, after the execution of the series of data requests needed to populate the visualizations. The analyzed information requested for this test involved data from more than 1000 students.

Table 1: Comparison of performance results between the Observatory's REST API and the GraphQL API during the load of an information dashboard. R1-R10 are the network response times obtained in each access to the dashboard.

	REST API	GraphQL API
Total request size	26.66 KB	13.90 KB
Request 1	6.54s	3.85s
Request 2	7.45s	4.17s
Request 3	5.82s	4.94s
Request 4	5.50s	3.90s
Request 5	6.14s	4.95s
Request 6	5.26s	4.16s
Request 7	5.79s	4.11s
Request 8	5.94s	3.97s
Request 9	7.06s	3.79s
Request 10	5.99s	3.77s
Average network response time	6.15s	4.16s
Improvement with GraphQL:	32.36% of network time decreased 47.86% less of broadband loading required	

Not only the average network time to retrieve all the analyzed data has been reduced by this change, but also the total size of the requests, thanks to the unification of all the previous REST API requests into a single GraphQL API request. The rest of the presentation components of the ecosystem had experienced similar network performance gains.

Although the performance gain could not be vital in some systems, the flexibility gained is vital for this type of ecosystem. If some component changes its structure or even its functionality, it is only necessary to modify the query made by the component itself to fit the new data requirements. This can save time in the development for the future regarding the data-driven ecosystem's components evolution.

The Fig. 6 shows the graphical representation of the test results.

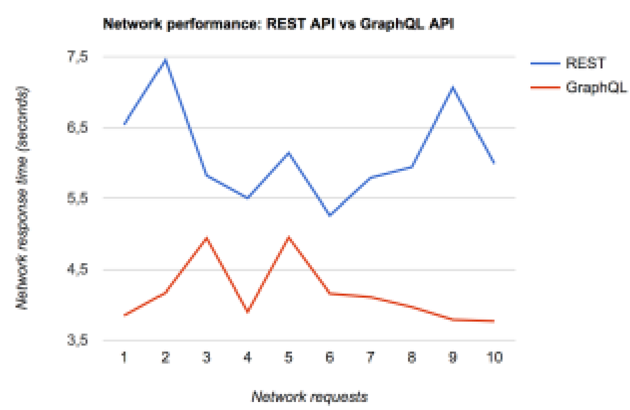


Figure 6: Graphical representation of the network response times obtained after executing the API requests.

5 DISCUSSION

Due to the improvement of the Observatory's ecosystem interoperability, some issues associated to their technical challenges have been solved in the past with the REST API. Although the interoperability levels provided by the REST API implemented were enough at the time, the potential evolution of the requirements and the components of the Observatory asked for more flexibility and scalability levels.

The introduction of the GraphQL API to the Observatory's data-driven technological ecosystem provided flexibility, scalability and maintainability regarding its components' evolution. Also, the network time to retrieve the data has been reduced, which is another important benefit giving the significant amount of data handled.

Scalability and flexibility are characteristics that the previous REST API lacked. These characteristics are crucial for this data-driven ecosystem considering that the Observatory's fields of study (employment and university employability) are continuously evolving and the data amount gathered to analyze is continuously increasing.

With the REST API, the backend developers had to design and specify the data responses for every endpoint of the API before any client could use it. But with the GraphQL API, developers only need to specify the ecosystem's data objects (and their fields) available and even filters or operations, and then the GraphQL framework chosen will handle the clients’ queries.

This approach makes the addition and modification of data objects (and fields) quasi-trivial, because only the GraphQL entities are involved in the change, in contrast with REST APIs’, where a series of endpoints could be affected by the modification of the data entities.

It is task of the clients (data consumers and prosumers) to design their queries based on their requirements and the data available on the backend, and not of the backend to design the endpoints based on the clients’ data requirements. The backend makes available their data resources, and the clients decide how to consume it.

But not only the internal components of the data-driven ecosystem are beneficed by the GraphQL approach; external components (whose data requirements are out of the Observatory’s control) now have a flexible communication method to connect themselves to the internal ecosystem’s components.

GraphQL has also resolved the issue regarding the specificity of the REST endpoints; most of the endpoints of the Observatory’s previous REST API were designed specifically for its internal components, so, although external components could access these endpoints, the results of their request could not fit the requirements to accomplish their purposes. The GraphQL API gives freedom to external components to design their own queries and promotes the exploitation of the knowledge obtained by the Observatory. This also could represent issues regarding the security: The GraphQL API should include object permission and access control depending the client that is using the API.

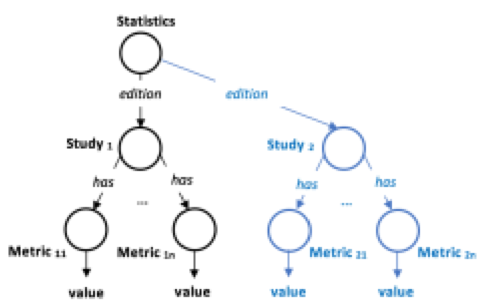


Figure 7: Visual structure of the OEEU's statistics using GraphQL. The addition of new metrics and fields implies the creation of new nodes and new relations between the already existent nodes.

Finally, the data graph-structuration provided by GraphQL gives the Observatory a more scalable and organized way to retrieve and analyze the collected data.

For instance, the Observatory needs a data entity to represent the metrics’ values derived from their collected data in order to ease the reaching of insights about employment and employability. With GraphQL, this data entity can be represented by a root node symbolizing the whole set of statistics generated by the Observatory. However, the Observatory is making a series of study editions through time, and the metrics’ values vary depending on the study edition. To address that problem, another node symbolizing the study edition can be attached to the root node. Then, it is only necessary to define the metrics and, again, attach them to the graph as leaf nodes of the specific study edition.

This approach has two benefits. The first benefit is that metrics are organized by the study edition, making the retrieval more intuitive for the clients. The second benefit is the increase of scalability regarding the Observatory’s data; new data entities representing new editions of the Observatory’s studies can join the initial root node of the data graph, minimizing the impact of the data’s evolution. This structure is conceptually showed in Fig. 7: the black colored elements can be seen as the initial graph created for the first edition of the Observatory’s study. Then, the following editions only need to join the statistics root node, creating a clean structure to browse and retrieve data.

This structure also allows the retrieval of data from different studies’ editions through one unique request, which simplifies the analysis of the evolution of the metrics’ values through time.

There are, though, some concerns to keep in mind derived from the GraphQL API’s implementation. The hierarchical structure of the queries can suppose a threat: the query of deep nested relations could end up in a denial of service attack and consume all the resources of the backend [15].

It is important to keep this kind of attacks in mind if the GraphQL API continues growing, as well as the authentication and authorization methods (as previously pointed out in the case of the Observatory).

Although the GraphQL API has reported significant benefits, its introduction to the ecosystem did not imply the shutdown of the previous REST API. The approach chosen uses the GraphQL API for internal and external components’ communication, which have particular technical requirements, but other users might not have these scalability, flexibility and performance needs.

REST APIs are simple, extended and don’t require the design of a particular query, so they could be useful for general users of the Observatory

6 CONCLUSIONS

GraphQL can be seen as a powerful solution to increase the interoperability of data-driven and data-intensive (eco)systems because it provides high levels of flexibility, which help to support changing requirements along time.

The use of this query language also comes with an increase of performance due to the reduction of the number of requests, as well as higher levels of scalability and maintainability thanks to its "graph nature" [15].

As data-driven and data-intensive (eco)systems are composed by continuously evolving components (which have to stick to changing requirements), their scalability, flexibility and performance needs are crucial. These needs are what make the GraphQL approach suitable for these type of (eco)systems.

With GraphQL, the concept of data-as-a-service (DaaS) [16] is more authentic; data is provided on demand and clients can specify the structure, filters or even operations for the data retrieved.

However, the arising of GraphQL does not have to mean that REST APIs are going to disappear. Although the benefits derived from the use of GraphQL could make this language preferable over REST [15], this last protocol remains a suitable and simple solution for lots of systems that doesn't have critical interoperability and flexibility needs.

ACKNOWLEDGMENTS

The research leading to these results has received funding from "la Caixa" Foundation. Also, the author Juan Cruz-Benito would like to thank the European Social Fund and the *Consejería de Educación* of the *Junta de Castilla y León* (Spain) for funding his predoctoral fellow contract. This work has been partially funded by the Spanish Government Ministry of Economy and Competitiveness throughout the DEFINES project (Ref. TIN2016-80172-R).

REFERENCES

[1] Dj Patil and Hilary Mason. 2015. *Data Driven*. " O'Reilly Media, Inc.".

[2] A. Vázquez-Ingelmo, J. Cruz-Benito, and F. J. García-Peñalvo. In press. Scaffolding the OEEU's Data-Driven Ecosystem to Analyze the Employability of Spanish Graduates. In *Global Implications of Emerging Technology Trends*, F.J. García-Peñalvo Ed. IGI Global.

[3] A. García-Holgado and F. J. García-Peñalvo. 2016. Architectural pattern to improve the definition and implementation of eLearning ecosystems. *Science of Computer Programming* 129, 20-34. DOI:http://dx.doi.org/10.1016/j.scico.2016.03.010.

[4] F. J Garcia-Peñalvo and A. Garcia-Holgado. 2016. *Open Source Solutions for Knowledge Management and Technological Ecosystems*. IGI Global.

[5] R. T Fielding and R. N. Taylor. 2002. Principled design of the modern Web architecture. *ACM Transactions on Internet Technology (TOIT)* 2, 2, 115-150.

[6] M Faassen. 2015. GraphQL and REST, Secret Weblog.

[7] Facebook. 2016. GraphQL.

[8] S. Taylor. 2017. React, Relay and GraphQL: Under the Hood of the Times Website Redesign. In *Times Open*.

[9] F. Michavila, M. Martín-González, J. M Martínez, F. J. García-Peñalvo, and J. Cruz-Benito. 2015. Analyzing the employability and employment factors of graduate students in Spain: The OEEU Information System. In *Proceedings of the 3rd International Conference on Technological Ecosystems for Enhancing Multiculturality* ACM, 277-283.

[10] F, Michavila, J. M. Martínez, M. Martín-González, F. J. García-Peñalvo, and Juan Cruz-Benito. 2016. Barómetro de Empleabilidad y Empleo de los Universitarios en España, 2015 (Primer informe de resultados).

[11] J. Bichsel. 2012. *Analytics in higher education: Benefits, barriers, progress, and recommendations*. EDUCAUSE Center for Applied Research.

[12] Á. Fidalgo-Blanco, M. L. Sein-Echaluce, and F. J. García-Peñalvo. 2014. Knowledge Spirals in Higher Education Teaching Innovation. *International Journal of Knowledge Management* 10, 4, 16-37. DOI:http://dx.doi.org/10.4018/ijkm.2014100102.

[13] Á. Fidalgo-Blanco, M. L. Sein-Echaluce, and F. J. García-Peñalvo. 2015. Epistemological and ontological spirals: From individual experience in educational innovation to the organisational knowledge in the university sector. *Program: Electronic library and information systems* 49, 3, 266-288. DOI:http://dx.doi.org/10.1108/PROG-06-2014-0033.

[14] A. García-Holgado, J. Cruz-Benito, and F. J. García-Peñalvo. 2015. Analysis of knowledge management experiences in spanish public administration. In *Proceedings of the 3rd International Conference on Technological Ecosystems for Enhancing Multiculturality* ACM, 189-193.

[15] S. Buna. 2017. Rest APIs are REST-in-Peace APIs. Long Live GraphQL.

[16] O. Terzo, P. Ruiu, E. Bucci, and F. Xhafa. 2013. Data as a service (DaaS) for sharing and processing of large data collections in the cloud. In *Complex, Intelligent, and Software Intensive Systems (CISIS), 2013 Seventh International Conference on* IEEE, 475-480.

7.3 Appendix C. How different versions of layout and complexity of web forms affect users after they start it? A pilot experience

Cruz-Benito, J., Sánchez-Prieto, J. C., Vázquez-Ingelmo, A., Therón, R., García-Peñalvo, F. J., & Martín-González, M. (2018). How different versions of layout and complexity of web forms affect users after they start it? A pilot experience. In Á. Rocha, H. Adeli, L. P. Reis, & S. Costanzo (Eds.), Trends and Advances in Information Systems and Technologies (Vol. 2, pp. 971-979). Cham: Springer. doi:10.1007/978-3-319-77712-2_92

How Different Versions of Layout and Complexity of Web Forms Affect Users After They Start It? A Pilot Experience

Juan Cruz-Benito[1,3,4(✉)], José Carlos Sánchez-Prieto[1,4], Andrea Vázquez-Ingelmo[1,3,4], Roberto Therón[1,2,3], Francisco J. García-Peñalvo[1,3,4], and Martín Martín-González[5]

[1] GRIAL Research Group, University of Salamanca, Salamanca, Spain
{juancb, josecarlos.sp, andreavazquez, theron, fgarcia}@usal.es
[2] VisUSAL Research Group, University of Salamanca, Salamanca, Spain
[3] Department of Computer Science, University of Salamanca, Salamanca, Spain
[4] Research Institute for Educational Sciences, University of Salamanca, Salamanca, Spain
[5] UNESCO Chair in University Management and Policy, Technical University of Madrid, Madrid, Spain
martin.martin@upm.es

Abstract. This paper presents a research work that analyzes the effect of redirecting users between two different versions of a web form after they have started the questionnaire. In this case, we used a web form proposed by the Spanish Observatory for Employability and Employment (OEEU) that is designed to gather information from Spanish graduates. These two versions are different as follows: one of them is very simple and the other one includes several changes that appeared in the literature related to users' trust, usability/user experience and layout design. To test the effect of redirecting users between both versions of the web form, we used a group of users that already have started the questionnaire and redirect them to the other version; this is, we changed the web form version they use to the other version and measure how this change affects them. This experiment has shown some promising results, which lead to enhance and extend the experience to bigger populations and other kind of changes in the user interfaces.

Keywords: Human-computer interaction · HCI · Web forms Online questionnaire · User experience · Performance

1 Introduction

Currently, the web forms are one of the most used ways to get information from the users [1]. The easiness of deploying web forms in websites and the users' habit to use them have converted online questionnaires in a pervasive tool to gather information. Thus, the research on engaging users to fulfil questionnaires and web forms is a research area that evolves continuously and relates to other areas like user experience [2], psychology [3], data retrieval [4, 5], etc.

Regarding the user experience (UX), the work done to improve web forms in recent years has been carried to advance in issues like how to properly design the web layout [6, 7], how to design interesting user experiences [8–10], on how to communicate effectively with the user to improve the trust on the web form [3, 11], formalize usability standards [9, 12, 13], etc.

Following some of these research lines, we are working on how to propose and design different versions of the same web form to measure and detect which versions are the best regarding to improve the users' performance [14, 15]. From a methodological perspective, our approach is based on A/B tests. Following the A/B tests foundations, we show different variations of a website (in this case of a web form) randomly to different users and measuring what variation is the most effective (in terms of click-performance, task-performance, etc.). In our current research, we have developed two main versions: one based on extreme simplicity (with no visual effects or transitions, a simple layout, etc.) and other less simple that include characteristics like transitions, elements that enhance the user's trust on the website, a more elaborated layout, etc. In the following section (methodology) both versions will be explained in depth.

The different versions have been developed for a web form used by the Spanish Observatory for Employability and Employment (OEEU in its Spanish acronym) [16]. The web form is intended to gather data about how graduates get employment after they left the university. In this case, this online questionnaire is the most important tool for the Observatory to obtain data and information, without it the Observatory would not have data to develop their studies about employability and employment.

In the case of this research, we have applied the two different versions randomly to the graduates that participate in the OEEU's data gathering questionnaire. After a while, we began a reinforcement phase where the graduates that dropped out the questionnaire (or did not start it) would be given an opportunity to participate again. In this reinforcement phase, we redirected users between both versions (swapping users between the simpler and the less simple) depending on their performance, to test how varying the web form's features and complexity would affect the users' performance in completing the questionnaire.

So, this paper presents the results of a pilot study carried by the Spanish Observatory for Employability and Employment and the GRIAL Research Group at the University of Salamanca (Spain), with the objective of examining the effects of changing the web form layout and features on the finalization rate of the users that have already initiated the answering process.

The article is organized in three sections. The first one is dedicated to describing the methodology. It details the description of the different versions of the questionnaire, and the redirection process, as well as the research design, and the sample. After that, we present the results obtained, including the hypothesis testing using three-dimensional contingency tables. Finally, we will close the communication with a discussion and a brief series of conclusions.

2 Methodology

This section presents the methodology and other relevant aspects of this research.

2.1 Different Versions of the Web Form

The two versions of the web form, as explained before, are differentiated basically because the first version is the simplest one (in the case of a web form) and the other is a bit more complex and has several features designed to engage user and develop an effective communication and relationship. The different changes proposed between both versions are based on different proposals retrieved from the literature and design guidelines, as explained in [14].

In the case of the simplest version, which we call "A" version, the web form is a basic form built using Bootstrap 3, with only one logotype (from OEEU), and a simple combination of visual elements with basic colors like white, blue and green (following the Bootstrap's style). A basic example of how is the layout of this "A" version can be found in this PDF (content in Spanish) or in [14].

In the case of the "B" version (the second one), it changes several things aimed at developing a closer relationship with the user (as proposed in the Social Exchange Theory [1]), enhancing the user's trust on the web form owner and its intentions [3, 11], improving issues related to user experience [17], usability [18] and interface design of the questionnaire [7, 19]. Specifically, the changes introduced in the "B" version of the web form were:

1. **Adequacy of the image to the other digital products of the Observatory**. In this change, related to enhance the users' trust, we planned to update the visual layout of the web form to meet the OEEU's design guidelines used in other of their digital products.
2. **Inclusion of the Observatory's logo and university's logo.** In this case, this is a change also related to building trust. It proposes to include the OEEU's logo in the web form header, as well as the logotype from the university where the student graduate.
3. **Inclusion of a progress bar in the questionnaire.** In this case, the proposed change was focused on improving the user experience with the web form. It is a simple change that consists (only) in including a progress bar that informs users about their progress in the task of finalizing the questionnaire.
4. **Present a visual focus animation on concrete actions.** Another proposal related to the usability and user experience. In this case, this change was designed to get the user attention and minimize the effort on using the web form. In this case, for example, the web form will auto scroll smoothly to the following question after the user responds to the previous one.
5. **Deactivation of control elements when an action is initiated.** This proposal consists on deactivating visual elements (like buttons) while they respond or complete an action requested by the user. For example, deactivating a button after the user clicks on it while the action triggered is completed.

6. **In related elements, instead of having smaller and more specific groupings, use some larger grouping, following the Gestalt principles on grouping.** This change was specially designed for large groups of questions/answers. Usually, in the web form, questions that include subquestions and nested response options are arranged in tables. For example, following the proposal, the header of a table would be fixed while the content can be scrolled up and down. It seeks to ensure that the large dimensions of analysis in some points of the questionnaire are grouped to avoid user fatigue and reducing the users' cognitive load when dealing with large tables or complex visual elements.

To get more information or find visual examples of these changes, we refer the reader to [14].

2.2 Redirection Process

As previously commented, the users were initially assigned randomly to use the "A" or "B" version of the web form. While users were using the web form, we analyzed what kind of factors (users' personal factors, technological aspects, etc.) were related to the users' performance in completing the questionnaire. By using predictive models and clustering techniques, we figured out the behaviors shared among users, what were their common characteristics, etc., to find patterns that define what lead users to achieve better performance metrics. As an example, in this previous research to know the most relevant user factors regarding to performance, in general we found that users have better average performance using the simplest version ("A"), except for those users that employ mobile or other devices with special specs like big screens or screens with an extremely good resolution, etc. Specifically, we found that users that meet the following criteria had better performance metrics in the "B" version:

1. Users that utilize Android devices with screens of 3- or 4-pixel ratio.
2. Users that accessed to the web form using large iPhone devices (iPhone 6 Plus, 6 s Plus, or 7 Plus).
3. Users that use Android tablets.

So, in general, in the reinforcement phase, all users that randomly were assigned to use the "B" version and did not meet these conditions were redirected to the "A" version. On the other hand, all users that used the "A" version in the initial stage and meet those conditions (or rules) were redirected to the "B" version in the reinforcement phase. This kind of rules were used to change users between both versions.

2.3 Research Design and Sample

The study presented in this paper is framed within another big (and more generalist) study about web forms and user experience. The whole experiment was conducted with more than 6700 users (graduates). Specifically, the questionnaires about employment were initiated by 6738 users, from which 5214 finalized the process (finalization ratio of 77.38%).

As previously commented, we ran a reinforcement phase after the first round of questionnaires; in this reinforcement stage were invited again to participate all those students that did not completed (or started) the web forms at the first round.

In the case of the study, 123 users were involved that participated in the first round and did not finalized the questionnaire, reentering again on the questionnaire during the reinforcement phase.

Analyzing these users, we studied the users' performance related to each version of the web form and we swapped users between both versions to test what is the effect of this change in their performance.

To do so, we proposed a quasi-experimental research design with a control group. Following this design, we divided the users in two groups: the experimental group (89), composed by the users that were redirected from one questionnaire design to a different one, and the control group (34) composed by those users that remained in the same questionnaire design.

After the application of the different treatment to each one of the groups we compared the differences in the finalization rate using three-dimensional-contingency tables and chi-squared to analyze the impact of increasing or decreasing the complexity of the questionnaire. In consequence, we pose the following hypotheses:

H1 The redirection to a different version of a questionnaire will have an impact on the finalization rate.

H2 The redirection of users from a text plain questionnaire to one with more complex elements will have an impact on the finalization rate.

H3 The redirection of users from a questionnaire with complex elements to a plain text one will have an impact on the finalization rate.

3 Results

As mentioned before, to assess the general effect of the redirection of the users on the improvement of the finalization rate we have used an approach based on the use of three-dimensional-contingency tables, a methodology of analysis useful to compare the effect of a variable in the relationship of the other two variables.

In this case, we will begin analyzing the effect of the web form version in the users' finalization rate to decide which the users will be redirected following our rules in the redirection phase. In other words, we analyze the effect of the redirection considering the version of the questionnaire to decide which users will be redirected during the reinforcement phase.

As a first step for the analysis, we elaborated the three-dimensional contingency table for this variable to see if there are observable differences at plain sight between the control and the experimental group considering the version of the questionnaire to which the users were redirected (Table 1).

As we can see in Table 1, there were some minor differences in the finalization rates of the control and the experimental groups in both versions of the questionnaire. In consequence, we proceed with the calculation of the chi-squared index to find out

Table 1. Three-dimensional contingency table for questionnaire redirection.

Version redirected to	Group	Finalized	
		Yes	No
A	Control	10	4
	Experimental	38	33
	Total	48	37
B	Control	9	11
	Experimental	7	11
	Total	16	22

wether there were any relation between being redirected and the finalization of the questionnaire (Table 2). The results lead to the rejection of the hypothesis H1 (*the redirection to a different version of a questionnaire will have an impact on the finalization rate*) in both versions for a significance level of 0.05.

Table 2. Results of chi-squared for questionnaire redirected to

Vertical redirected to	Value	df	Significance
A	1.526	1	0.217
B	0.145	1	0.703

Lastly, to contrast the last two hypotheses, we perform the same procedure, but considering the complexity change. This way, we measured if the users changed, from a simpler version of a questionnaire to a more complex one or if at the contrary, the users change from a complex questionnaire to a simpler one. As in the previous case, we begin elaborating the three-dimensional contingency table (Table 3), but only with the users that were redirected from questionnaire A to questionnaire B or from questionnaire B to questionnaire A.

Table 3. Three-dimensional contingency table for questionnaire redirected from.

Version redirected from	Group	Finalized	
		Yes	No
A to B	Control	10	4
	Experimental	1	8
	Total	11	12
B to A	Control	9	11
	Experimental	17	19
	Total	26	30

After analyzing the results, we performed the correlational analysis to know if there were any relation between being redirected and the finalization of the questionnaire. For the case of the change from questionnaire A to questionnaire B we used Fisher's exact test, due to the size of the groups (Table 4). The results support hypothesis H2 (*the redirection of users from a text plain questionnaire to one with more complex elements will have an impact on the finalization rate*), but reject hypothesis H3 (*the redirection of users from a questionnaire with complex elements to a plain text one will have an impact on the finalization rate*).

Table 4. Results of correlation for questionnaire redirected from.

Vertical redirected to	Value	df	Significance
A to B	–	–	0.009
B to A	0.026	1	0.873

4 Discussion

The results obtained in the present research entail a series of implications both for theory and practice of the design of online questionnaires.

Firstly, we would like to highlight the rejection of the hypothesis H1. This can be caused by the fact that the redirection rules were based on the behavior of the users that entered the questionnaire for the first time, which indicates the need to deepen in the analysis of the behavior of the people that resume the questionnaire completion process to know how this variable may impact the finalization rate of this kind users.

In this line, the results of the analysis of the hypotheses 2 and 3 suggests that increasing the number of design elements in the questionnaire has a negative effect on the finalization rate, while the redirection to a plain text questionnaire does not have any effect.

A possible explanation may lay in the users' motivation. The users participating in this pilot study are those that have already tried to complete the questionnaire but abandoned the process, which make very likely that their motivation levels were low. Taking this into account, it is logical to think that when these users resume the process, finding a questionnaire with more design elements that the one that they initiated in the first place, may feel discouraged.

As a consequence, we believe that avoiding this kind of redirections in the case of users that are resumed the questionnaire is advisable.

Another possible explanation could be related to some of the elements introduced or removed when changing the web form version. In this sense, the experiment in its current setup cannot allow us to ascertain which changes affect more or less to user performance. For that reason, one of the main shifts and improvements in this experiment and research could be to divide more the versions, achieving the same number of versions than the total changes. So, in this case, we would be able to detect how affects each individual change to the user's performance compared to the simplest version.

5 Conclusions

This paper presents a novel research work that analyzes the effect of redirecting users between two different versions of a web form after they have started the questionnaire. We used a web form proposed by the Spanish Observatory for Employability and Employment (OEEU) that is designed to gather information from Spanish graduates. To test the effect of redirecting users between both versions of the web form, we used a group of users that already have started the questionnaire and redirect them to the other version; this is, we changed the web form version they use to the other vertical and measure how this change affects them.

In general, the results are quite promising and encourage us to continue the labor of researching how different changes in web forms affect users' performance. In this case, we can conclude that if we redirect users between two versions of a web form, the change will be negative if the user is redirected to a more complex version and will not have effect if it is redirected to a simpler version. In the future, we would like to enhance and extend the experiment to bigger populations and other kind of changes in the user interfaces to verify these initial results. Also, we would like to test how users feel the change and what is their opinion about the change (to compare also their feelings and perception to their performance).

Acknowledgments. The research leading to these results has received funding from "la Caixa" Foundation. The author Juan Cruz-Benito would like to thank the European Social Fund and the *Consejería de Educación* of the *Junta de Castilla y León* (Spain) for funding his predoctoral fellow contract. This research has been cofounded by the University of Salamanca through the program of financial aid for the predoctoral contract of José Carlos Sánchez-Prieto (*Programa III: Ayudas para contratos Predoctorales*) cofounded by Banco Santander. This work has been partially funded by the Spanish Government Ministry of Economy and Competitiveness throughout the DEFINES project (Ref. TIN2016-80172-R).

References

1. Dillman, D.A.: Mail and Internet Surveys: The Tailored Design Method. Wiley, New York (2000)
2. Jarrett, C., Gaffney, G.: Forms That Work: Designing Web Forms for Usability. Morgan Kaufmann, Amsterdam (2009)
3. Seckler, M., Heinz, S., Forde, S., Tuch, A.N., Opwis, K.: Trust and distrust on the web: user experiences and website characteristics. Comput. Hum. Behav. **45**, 39–50 (2015)
4. Stieger, S., Reips, U.-D.: What are participants doing while filling in an online questionnaire: a paradata collection tool and an empirical study. Comput. Hum. Behav. **26**, 1488–1495 (2010)
5. Kveton, P., Jelínek, M., Klimusová, H., Voboril, D.: Data collection on the internet: evaluation of web-based questionnaires. Studia Psychol. **49**, 81 (2007)
6. Flavián, C., Gurrea, R., Orús, C.: The effect of product presentation mode on the perceived content and continent quality of web sites. Online Inf. Rev. **33**, 1103–1128 (2009)

7. Seckler, M., Tuch, A.N., Opwis, K., Bargas-Avila, J.A.: User-friendly locations of error messages in web forms: put them on the right side of the erroneous input field. Interact. Comput. **24**, 107–118 (2012)
8. Bargas-Avila, J.A., Brenzikofer, O., Tuch, A.N., Roth, S.P., Opwis, K.: Working towards usable forms on the worldwide web: optimizing multiple selection interface elements. Adv. Hum. Comput. Interact. **2011**, 4 (2011)
9. Bargas-Avila, J., Brenzikofer, O.: Simple but crucial user interfaces in the world wide web: introducing 20 guidelines for usable web form design (2011)
10. Anderson, S.P.: Seductive Interaction Design: Creating Playful, Fun, and Effective User Experiences, Portable Document. Pearson Education, Berkeley (2011)
11. Shneiderman, B.: Designing trust into online experiences. Commun. ACM **43**, 57–59 (2000)
12. ISO/DIS: Draft BS ENISO 9241-220 Ergonomics of human-computer interaction. Part 220: Processes for enabling, executing and assessing human-centred design within organizations (2016)
13. ISO/DIS: Ergonomics of human-system interaction — Part 125: Guidance on visual presentation of Information (2013)
14. Cruz-Benito, J., Therón, R., García-Peñalvo, F.J., Sánchez-Prieto, J.C., Vázquez-Ingelmo, A., Martín-González, M., Martínez, J.M.: Improving success/completion ratio in large surveys: a proposal based on usability and engagement. In: Zaphiris, P., Ioannou, A. (eds.) Proceedings of 4th International Conference on Learning and Collaboration Technologies, LCT 2017, Technology in Education, Held as Part of HCI International 2017, 9–14 July 2017, Vancouver, BC, Canada, Part II, pp. 352–370. Springer International Publishing, Cham (2017)
15. Cruz-Benito, J., Vázquez-Ingelmo, A., Sánchez-Prieto, J.C., Therón, R., García-Peñalvo, F. J., Martín-González, M.: Enabling adaptability in web forms based on user characteristics detection through A/B testing and machine learning. IEEE Access **5**, 2251–2265 (2017)
16. Michavila, F., Martínez, J.M., Martín-González, M., García-Peñalvo, F.J., Cruz-Benito, J.: Barómetro de Empleabilidad y Empleo de los Universitarios en España, 2015 (Primer informe de resultados) (2016)
17. Recabarren, M., Nussbaum, M.: Exploring the feasibility of web form adaptation to users' cultural dimension scores. User Model. User-Adapt. Interact. **20**, 87–108 (2010)
18. Huber, W., Vitouch, P.: Usability and accessibility on the internet: effects of accessible web design on usability. In: International Conference on Computers for Handicapped Persons, pp. 482–489. Springer (2008)
19. Pengnate, S.F., Sarathy, R.: An experimental investigation of the influence of website emotional design features on trust in unfamiliar online vendors. Comput. Hum. Behav. **67**, 49–60 (2017)

7.4 Appendix D. Enabling adaptability in web forms based on user characteristics detection through A/B testing and machine learning

Received October 23, 2017, accepted November 29, 2017, date of publication December 27, 2017,
date of current version February 14, 2018.

Digital Object Identifier 10.1109/ACCESS.2017.2782678

Enabling Adaptability in Web Forms Based on User Characteristics Detection Through A/B Testing and Machine Learning

JUAN CRUZ-BENITO[1], (Member, IEEE), ANDREA VÁZQUEZ-INGELMO[1], JOSÉ CARLOS SÁNCHEZ-PRIETO[2], ROBERTO THERÓN[3], FRANCISCO JOSÉ GARCÍA-PEÑALVO[1], (Member, IEEE), AND MARTÍN MARTÍN-GONZÁLEZ[4]

[1]GRIAL Research Group, Computer Science Department, Research Institute for Educational Sciences, University of Salamanca, 37008 Salamanca, Spain
[2]GRIAL Research Group, Research Institute for Educational Sciences, University of Salamanca, 37008 Salamanca, Spain
[3]GRIAL Research Group, Computer Science Department, and VisUSAL Research Group, University of Salamanca, 37008 Salamanca, Spain
[4]UNESCO Chair in University Management and Policy, Technical University of Madrid, 28003 Madrid, Spain

Corresponding author: Juan Cruz-Benito (juancb@usal.es)

This work was supported in part by the la Caixa Foundation, in part by the Spanish Government Ministry of Economy and Competitiveness throughout the DEFINES Project under Project TIN2016-80172-R, and in part by the University of Salamanca through the Program of Financial Aid for the Predoctoral Contract of José Carlos Sánchez-Prieto (*Programa III: Ayudas para contratos Predoctorales*) cofounded by Banco Santander. The work of J. Cruz-Benito was supported by the European Social Fund and the Consejería de Educación of the Junta de Castilla y León, Spain.

ABSTRACT This paper presents an original study with the aim of improving users' performance in completing large questionnaires through adaptability in web forms. Such adaptability is based on the application of machine-learning procedures and an A/B testing approach. To detect the user preferences, behavior, and the optimal version of the forms for all kinds of users, researchers built predictive models using machine-learning algorithms (trained with data from more than 3000 users who participated previously in the questionnaires), extracting the most relevant factors that describe the models, and clustering the users based on their similar characteristics and these factors. Based on these groups and their performance in the system, the researchers generated heuristic rules between the different versions of the web forms to guide users to the most adequate version (modifying the user interface and user experience) for them. To validate the approach and confirm the improvements, the authors tested these redirection rules on a group of more than 1000 users. The results with this cohort of users were better than those achieved without redirection rules at the initial stage. Besides these promising results, the paper proposes a future study that would enhance the process (or automate it) as well as push its application to other fields.

INDEX TERMS Adaptability, machine learning, user profiles, web forms, clusters, hierarchical clustering, random forest, A/B testing, human-computer interaction, HCI.

I. INTRODUCTION

Understanding what users do within a system is now a fundamental task in the digital world [1]. Most aspects of modern development workflows include users as a centric part of the design and development process of digital products (i.e., user-centered design [2], [3]). Not only knowing what users do (clicks, workflows, interactions, etc.) within a system is valuable for software developers and designers, but these stakeholders must also pay attention to other related-aspects, like user experience, satisfaction, and trust [4]–[8]. Understanding what users do or feel when they use a system is extremely valuable to validate and improve a system. Analyzing users' interactions or their opinion about what they use makes it possible to ascertain the system's strengths or weaknesses regarding users' experience (mostly user interfaces and parts alike) to improve the system based on evidence.

Besides using the analysis of users' interactions and opinions to improve the worst-perceived parts of a system, developers can use these data to build custom or adaptive solutions for different kinds of users [9]–[11]. Using this idea, software engineers could develop versions of the system in which different version are showed to each kind of user. By knowing user profiles and identifying users' behavior and desires, the system could adapt its components to better match users' expectations and likings, and (probably) boost user performance and satisfaction [8], [9], [12].

For a better understanding of the current paper, the context for this experiment is presented. The research has been conducted using a system that belongs to the Spanish Observatory for University Employability and Employment (OEEU in its Spanish acronym) [13]. This observatory gathers data about employment and employability parameters among the Spanish graduates (after they leave the university) to analyze the information they provide and understand what the employment trends and most important employability factors are for this population. To accomplish this mission, the observatory has developed a complex information system [14], [15] that collects and analyzes data to present the insights to the researchers. The system is implemented using the Python language through the Django framework [16] and many other software libraries; it also keeps the information in a MariaDB relational database. To gather data from Spanish universities and students, the OEEU information system has two main tools: one tool is devoted to obtaining students' raw data provided by the university; the other one is a system that generates custom web forms and questionnaires that are to be completed by the graduates after they leave the university. The problem of these web forms is their length, as they typically include between 30 and 70 questions. This second tool for gathering data (the questionnaires) is a centric part in this research.

The goal of this paper is to present a new approach for enabling adaptability in web-based systems using A/B testing methods and user-tracking and machine-learning algorithms that could lead to improving user performance in completing a (large) web form, validating the obtained results through statistical tests. As a secondary goal, the research presented in this paper also aims to produce all machine learning processes in a white-box way, using algorithms and techniques that allow researchers to understand what is happening in every moment. Moreover, to allow readers and other researchers to follow or reproduce the entire process, this paper provide all the code used in the analysis process in Jupyter notebooks available publicly in Github.

The paper has the following structure: section two (Materials and Methods) explains the different algorithms, data, and research framework. Section three (Results) presents the outcomes obtained in the different steps involved in the research: the results regarding the predictive models that provide the most important users' characteristics on completing the web form, those regarding users' profiles found, and those regarding the guidance of users over the different versions of the system to enhance their performance. The fourth section (Discussion) presents different authors' thoughts, proposals, and considerations about this research and its implications, as well as some future works and general conclusions.

II. MATERIALS AND METHODS

This section outlines the materials and methods used for this research. In the case of materials, the data used and the analysis software are described. In the case of the methods, the different steps needed to apply the machine-learning approach to the analysis process as well as the statistics used to prove the validity and significance of the results are presented.

A. MATERIALS

This subsection presents the different materials involved in this research. The materials can be categorized into two main groups: materials related to the experimentation framework and the software tools used to make the proper analysis and support the research process.

The questionnaires and custom web forms included in the OEEU information system gather data from students in two ways: information provided explicitly by the students (the information provided directly) and *paradata* [17]. The paradata from these questionnaires are the auxiliary data that describe the filling process, such as response times, clicks, scrolls, and information about the device used when using the system. All the data used in this research are taken from these two available sources: the raw input tool used by universities and the web forms tool (providing user inputs and their paradata).

Regarding the data used in this research, it is worth noting that to generate the predictive models needed to characterize the main factors that affect users in completing the questionnaires, the authors have chosen only those available before the users began the questionnaire. This is because the research is focused on investigating which factors predetermine participants' success or failure in completing the form, considering all the factors related only to personal context and device and software used to access the web forms. The data about the personal context of the user are provided by the OEEU's system and include information submitted by the university where the user (graduated) studied. All the information that could be used to create the models that predict whether the user will complete the questionnaire (before starting it) is presented in Table 1. Table 1 also explains the data variables used and whether they were valuable for the models. This research has been carried out with a total of 7349 users (all who have some type of experience with the web forms). Of them, the data from 5768 users were considered initially. Finally, data from 3456 users (those resultant after cleaning the data) were used to train and try the machine-learning algorithms (as will be explained in the following section); 1165 users were the cohort introduced in a phase of reinforcement for the questionnaires that validated the rules generated to adapt the web form to users. This number (1165) includes users who did not complete the web form in the first stage as well as users that joined the experiment during the reinforcement and validation phase. Other users (416) only viewed the web forms without starting them. For that reason, were not considered in the experimental report.

TABLE 1. Initial variables gathered from the OEEU information system to build the predictive models of questionnaires' completion.

Name of the variable in the code	Explanation	Type of information that it provides	Was this variable used finally to build the predictive models?
estudiante_id	ID number of student	Personal information	Yes
annoNacimiento	Year of birth	Personal information	No
sexo_id	Gender (male / female)	Personal information	Yes
esEspannol	Is the student Spanish?	Personal information	No
universidad_id	ID of the university where the graduate studied	Personal information	Yes
estudiosPadre_id	Maximum educational level achieved by the graduate's father	Personal information	No
estudiosMadre_id	Maximum educational level achieved by the graduate's mother	Personal information	No
situacionLaboralPadre_id	Current employment status of the graduate's father	Personal information	No
situacionLaboralMadre_id	Current employment status of the graduate's mother	Personal information	No
oficioProfesionPadre_id	Occupation of the graduate's father	Personal information	No
oficioProfesionMadre_id	Occupation of the graduate's mother	Personal information	No
residenciaFamiliar_id	Place of residence of the graduate's family	Personal information	No
residencia_id	Place of residence of the graduate during studies	Personal information	No
idMaster_id	ID number of the master study	Personal information	Yes
especializacionMaster_id	Specialization of the graduate's master	Personal information	No
masterHabilitante	Is an enabling master?	Personal information	No
titularidadMaster_id	Public or not master	Personal information	No
modalidadMaster_id	Modality of the master (online, physical, etc.)	Personal information	No
cursoInicioMaster	Season of the beginning of the master	Personal information	No
cursoFinalizacionMaster	Season of the completion of the master	Personal information	Yes
notaMedia_id	Average grade of the student	Personal information	No
realizacionPracticasMaster	Did the student professionally practice during the master?	Personal information	No
tiempoDuracionPracticasMaster	Time spent by the student in professional practices during the master	Personal information	No
realizacionErasmusMaster	Did the student do an Erasmus stay?	Personal information	No
tiempoDuracionErasmus_id	Time spent by the student in an Erasmus stay	Personal information	No
paisErasmusMaster_id	Country where the student did an Erasmus stay	Personal information	No
viaAccesoMaster_id	Way of accessing the master	Personal information	No
verticalAsignado	Vertical assigned in the A/B testing for the student	Experiment configuration	Yes
cuestionarioFinalizado	Did the student finalize the questionnaire?	Experiment configuration	Yes
numUniversidades	Number of universities involved in the master	Personal information	Yes
numUniversidadesEspannolas	Number of Spanish universities involved in the master	Personal information	Yes
ramaConocimiento_id	Knowledge branch of the master (healthcare, social sciences, engineering, etc.)	Personal information	Yes
realDecreto	Official statement approving of the master studies program	Personal information	Yes
browser_language	Language of the browser used	Device information	Yes
browser_name	Name of the browser used	Device information	Yes
browser_version	Version of the browser used	Device information	Yes
device_pixel_ratio	Device pixel ratio of the browser	Device information	Yes
device_screen_height	Device screen height	Device information	Yes
device_screen_width	Device screen width	Device information	Yes
landscape	Is the device in landscape mode?	Device information	No
os	Operative system of the device	Device information	Yes
os_version	Version of the operative system used	Device information	Yes
portrait	Is the device in portrait mode?	Device information	No
push_notification	Did accept the graduate push notifications for the web form?	Device information	No
push_notification_id	ID number for the push notification subscription	Device information	No
tablet_or_mobile	Is the device tablet or mobile?	Device information	Yes
userAgent	User agent of the device used	Device information	Yes
viewport_height	Height of the window browser	Device information	Yes
viewport_width	Width of the window browser	Device information	Yes

The variables excluded to build the predictive models are those that have more than 10% of their observations with the null value.

The programming language used to conduct all the analyses and calculations was Python. The concrete Python software tools and libraries used to code and execute the different algorithms and statistics were:

- Pandas software library [18]–[20], to manage data structures and support analysis tasks.
- Scikit-learn [21] library, to accomplish the machine learning workflow [22].
- Jupyter notebooks [23]–[25], to develop the Python code used in this research.

All the code developed to analyze user interactions and create machine-learning models, etc. is available at https://github.com/juan-cb/paper-ieeeAccess-2017 [26].

B. METHODS

As found in the bibliography, the concept of A/B testing (also known as bucket testing, controlled experiment, etc.) applied to websites and the Internet could be explained as follows: "show different variations of your website to different people and measure which variation is the most effective at turning them into customers (or people that complete successfully a task in the website, like in this experiment). If each visitor to your website is randomly shown one of these variations and you do this over the same period, then you have created a controlled experiment known as an A/B test" [27]–[29]. In this case, the authors have prepared three different variations, called verticals A, B, and C. In each variation, the authors introduced several changes related to enhancing the users' trustiness, engagement, make the user interface more conversational, etc. All these changes, introduced in the different variations of the web forms (the verticals) used in this research, were proposed by the authors in previous works [30]. These verticals are used as the website variations in which users (students responding to the questionnaires) are meant to test which version is the best regarding the users' performance in the initial stage. To do so, before the experiment, 5768 users were redirected randomly to the different vertical. In the last part of the experiment, the verticals were used to check whether the rules and users' analysis performed during the machine-learning analytics process improve the users' performance in completing the web forms. In this validating phase (which also will be called reinforcement in this paper), 1165 users were redirected to the verticals using the rules generated analyzing the interaction data from the users that acceded randomly to the verticals.

In general, the performed analysis (based on statistics and machine learning) follows common principles in data science regarding data structuration, tidy data approaches, etc. [18], [20], [31]. As stated in the introduction, the machine-learning process has been implemented in a white-box way; thus, the researchers have selected algorithms and methods to make the workflow explainable. This is extremely important, from the authors' point of view, in a research project like this, as it allows humans to provide feedback to the algorithmic process.

Moreover, these main principles, the different details for the analysis pipeline, and methods used in this research are presented.

To find the best models and most accurate parameters, researchers have tried the following approaches:

1. Create predictive models using all the data together. In this approach, researchers tried to use different groups of variables to create the model: all the variables collected from the users, using derived variables (like whether the browser or operative system used to access were modern), etc.
2. Create predictive models using the verticals gap. In this case, researchers generated a predictive model per each vertical of the A/B test. In this case, the most relevant configuration regarding the variables to build up the model in the previous step is included.

Using the most accurate models, the researchers applied all the stages that will be described below (as well as the details for building the predictive models) to generate the different clusters and obtain the rules used to redirect users within the system.

The workflow established (available at https://github.com/juan-cb/paper-ieeeAccess-2017 [26]) is as follows:

1. Retrieve datasets about users from OEEU's information system.
2. Filter the desired fields from the datasets and merge datasets in a single data frame (a data structure like a table).
3. Data cleaning: remove noise data, remove columns (variables) with too many null (*NaN*) values, and remove all users who have only partial information and not all presented in Table 1.
4. Normalize data with the One-hot encoding algorithm for categorical values in columns [22]. To apply the One-hot encoding, researchers used the get_dummies() function from Pandas library, as presented in [26].
5. Considering the data gathered and the kind of variable (labeled) to predict, the algorithm to use must be related to supervised learning. This is because this kind of algorithm makes predictions based on a set of examples (that consist of a labeled training data set and the desired output variable). Moreover, regarding the dichotomous (categorical) character of the variable to predict, the supervised learning algorithm to apply must be based on classification (binary classification, as we have a label of finalization equal to *true* or *false*). According to the authors' previous experience, the possibility of explaining results and the accuracy desired for the classification, a Random Forest classifier algorithm [32] was selected. In this step, the Random Forest algorithm was executed repeatedly, using a custom method based [26] on *GridSearch* functions from Scikit-learn, to determine the best setup for the dataset given (obtaining the most valuable parameters for the execution).
6. With the best configuration found, train the random forest algorithm (with 33.33% of the dataset) and obtain the predictive model.
7. Using the predictive model, obtain the most important features for the predictive model. To obtain these features authors applied *feature_importances_* method from the Random Forest classifier implemented in Scikit-learn library [26].
8. Using the most important features (those that have an importance higher than a custom threshold value of 0.05—the importance score varies between 0 and 1, where 0 is the worst score and 1 the best one), generate clusters applying hierarchical clustering [33]. The reason to use hierarchical clustering is that the algorithm does not require deciding upon the number of clusters to obtain (so, it does not require also to fix previously Euclidean distances and other parameters); it obtains all possible clusters showing the Euclidean distance between them. These clusters represent the groups of users who have participated in the questionnaire according to the most important factors found in the classification.
9. With these clusters, the researchers investigate which clusters exhibit low performance.
10. Using this knowledge about groups of users with low performance and the heuristics observed, software engineers responsible for the OEEU's information system and its web forms could propose changes and

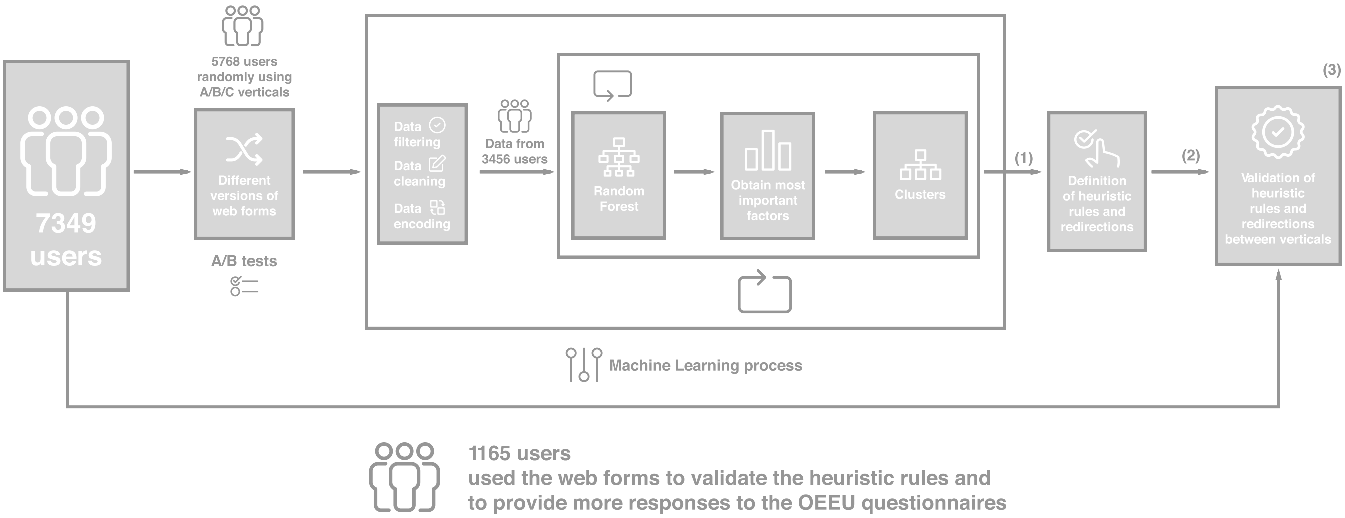


FIGURE 1. Overview of the process followed. Summary of the materials and methods.

fixes (rules, redirections, etc.) in the platform that might help users to improve their performance in the future.

11. Once the data-gathering process is finished, the researchers performed a statistical analysis of the finalization rate of the individuals to determine whether the application of the rules had any impact in the improvement of the finalization of the questionnaires. With this purpose, and considering the characteristics of the variables, the authors applied the Chi-squared test given that it is the most convenient alternative for the analysis of the relationship of two nominal variables.

All these steps and a summary of all methods and materials are presented in the Figure 1.

III. RESULTS

This section presents the main results obtained during the research. The outcomes are divided into three subsections: one related to the results obtained during the machine-learning process (best setup, best ways of building predictive models, the predictive models themselves, the most important variables to finalize or the questionnaire, etc.). The second subsection explains the heuristic rules obtained at the end of the machine-learning workflow inferred from the machine-learning results previously explained. These rules were applied to redirect users within the different verticals of the A/B tests. Finally, the results of the redirections are presented, explaining whether they really affected to the users' finalization of the questionnaire.

A. RESULTS REGARDING MACHINE-LEARNING PROCEDURES: PREDICTIVE MODELS AND CLUSTERING

As previously explained, the researchers performed several attempts to find the most accurate predictive models that better explain whether users will finalize the questionnaire. The first attempt was based on using all the data together focusing in primary variables (excluding those that have too many void values); the second one was based on using all variables and derived variables (constructed from primary ones). The third attempt was based on creating separated predictive models depending on the vertical. In this way, the researchers predicted users' behavior regarding the finalization depending on the vertical / interaction features that they experience. In this last approach, the researchers used the best set of variables found previously to build the model.

TABLE 2. Results of the first predictive model built.

	Precision[a]	Recall[b]	F1-score[c]	Support[d]
False	0.84	0.38	0.52	378
True	0.77	0.97	0.86	815
Avg / total	0.79	0.78	0.75	1193

[a]The precision is the ratio *tp / (tp + fp)* where *tp* is the number of true positives and *fp* the number of false positives. The precision is intuitively the classifier's ability of not labeling as positive a sample that is negative. This score reaches its best value at 1 and worst score at 0.

[b]The recall is the ratio *tp / (tp + fn)* where *tp* is the number of true positives and *fn* the number of false negatives. The recall is intuitively the ability of the classifier to find all the positive samples. This score reaches its best value at 1 and its worst score at 0.

[c]The F1 score can be interpreted as a weighted average of the precision and recall, where an F1 score reaches its best value at 1 and its worst score at 0. The relative contribution of precision and recall to the F1 score is equal. This score reaches its best value at 1 and its worst score at 0.

[d]The support is the number of occurrences of each class in each predicted label.

The results achieved in this phase would correspond to those expected in the (1) mark in Figure 1.

Regarding the first attempt to build the best predictive model, the researchers used all the variables (excluding the cleaned ones applying the rules defined in the methods sections). As presented in https://github.com/juan-cb/paper-ieeeAccess-2017/blob/master/machinelearning-results.ipynb [26], the predictive model generated had an average precision of 0.79 (Table 2 shows the results and explanations of the results metrics) in predicting whether users will finalize the web form before starting it (in fact, this 0.79 is a fairly good precision score). In the case of this research, the authors use the precision score as the main metric to make decisions, as it is focused on penalizing false positives [34].

The crosstab (that expresses the number of good and bad predictions) for this first predictive model can be found in Table 3.

TABLE 3. Crosstab for the first predictive model built.

	False (predictions)	True (predictions)
False (actual)	142	236
True (actual)	27	788

In this first attempt and its 0.79-precision predictive model, the most important factors in the model were (the importance score varies between 0–1, where 1 is the best score and 0 the worst one):

1. *device_screen_width*: 0.297189
2. *viewport_width*: 0.292615
3. *browser_name_Firefox*: 0.100000
4. *device_pixel_ratio*: 0.098356
5. *viewport_height*: 0.096237

In the second attempt, the researchers used the same variables plus two derived variables composed using the primary ones. The derived variables were *modern_browser* and *modern_os*. Those variables were calculated using the versions of operative systems and browsers used by users. In this case, the researchers calculated the median version of the operative system or browser (the midpoint between the oldest version and newest one present) and classified the browser or operative system as modern or not depending on whether its version is equal or superior to the mid version or is lower. These derived variables were prepared because it was impossible to use the literal version of each browser or operative system in the random forest algorithm due their heterogeneous expressions (each browser or OS has its own version's description and format, etc.). In this second attempt, the precision of the predictive model was higher—specifically, a precision of 0.81 (Table 4). The crosstab for this second model is presented in Table 5.

TABLE 4. Results of the second predictive model built.

	Precision	Recall	F1-score	Support
False	0.91	0.34	0.50	378
True	0.76	0.98	0.86	807
Avg / total	0.79	0.78	0.75	1185

In general, this second model performed better than the previous one (at least it was most precise). In this case, the most important factors that define the model were:

1. *tablet_or_mobile*: 0.179032
2. *device_pixel_ratio*: 0.159406
3. *device_screen_height*: 0.097580
4. *device_screen_width*: 0.095784

TABLE 5. Crosstab for the second predictive model built.

	False (predictions)	True (predictions)
False (actual)	129	249
True (actual)	13	794

5. *viewport_height*: 0.089050
6. *os_Android*: 0.063415

Since the variables used to build the predictive model were different from the previous one, it is normal that the factors that define the model could differ.

In the third approach to generate the best predictive model, the researchers generated a predictive model per each vertical in the A/B test applied to the users. In this case, the researchers included all the variables that produced the best predictive model previously: this is, the variables from the second attempt (including the variables *modern_os* and *modern_browser*). In this case, the researchers have trained three different random forest algorithms, found the best setup for each one depending on the data to analyze, and produced a model for each vertical. The results of these predictive models are presented in Tables 6, 7, and 8, and their precision varied between 0.79 and 0.87. The average precision in the three models was of 0.8233, which is higher than the precision achieved in the previous attempts of generating predictive models. Tables 9, 10, and 11 present the crosstabs for each model; they explain how much effective was the prediction depending on the finalization in the web form.

TABLE 6. Results of the predictive model for the vertical A.

	Precision	Recall	F1-score	Support
False	0.92	0.35	0.51	69
True	0.85	0.99	0.92	263
Avg / total	0.87	0.86	0.83	332

TABLE 7. Results of the predictive model for the vertical B.

	Precision	Recall	F1-score	Support
False	0.92	0.37	0.52	161
True	0.74	0.98	0.85	301
Avg / total	0.81	0.77	0.73	462

Regarding the most important factors per each predictive model generated in the third attempt, the results were the following:

Most influential factors for the predictive model for vertical A:

1. *viewport_width:* 0.267931
2. *tablet_or_mobile:* 0.139438
3. *os_iOS:* 0.132425
4. *device_screen_height:* 0.118814
5. *device_screen_width:* 0.067581
6. *device_pixel_ratio:* 0.066577
7. *os_Android:* 0.054088

TABLE 8. Results of the predictive model for the vertical C.

	Precision	Recall	F1-score	Support
False	0.89	0.37	0.52	132
True	0.74	0.97	0.84	238
Avg / total	0.79	0.76	0.73	370

TABLE 9. Crosstab of the predictive model results for vertical A.

	False (predictions)	True (predictions)
False (actual)	24	45
True (actual)	2	261

TABLE 10. Crosstab of the predictive model results for vertical B.

	False (predictions)	True (predictions)
False (actual)	59	102
True (actual)	5	296

TABLE 11. Crosstab of the predictive model results for vertical C.

	False (predictions)	True (predictions)
False (actual)	49	83
True (actual)	6	232

Most influential factors for the predictive model for vertical B:

1. *viewport_height:* 0.294176
2. *viewport_width:* 0.167701
3. *device_screen_height:* 0.102463
4. *device_pixel_ratio:* 0.085122
5. *os_Android:* 0.076196

Most influential factors for the predictive model for vertical C:

1. *device_screen_width:* 0.193903
2. *viewport_height:* 0.143456
3. *device_screen_height:* 0.108721
4. *tablet_or_mobile:* 0.100000
5. *viewport_width:* 0.093584
6. *device_pixel_ratio:* 0.088479
7. *os_Windows:* 0.055153

Analyzing the results, researchers found that the best way, in this case, to obtain the most-precise predictive models for users' interaction, was obtained by splitting the dataset using the vertical criteria. That is, separating the dataset into three datasets, each one including the data from each user cohort that experienced each one of the A/B tests versions. For that reason, the resultant models were selected to generate the clusters and study them to produce the rules to be used in redirecting users among the different visual representations of the web forms. Using these profiles (clusters) and the rules generated, the researchers found what kind of user (and its technological aspects) fits better (is more inclined to finalize) in each version of the questionnaires, forwarding the users using these criteria to each vertical.

After producing the predictive models, the researchers clustered users depending on their finalization ratio and the most important factors extracted in the predictive models. Explaining all clusters generated after producing each predictive model is out of the scope of this paper (but available at https://github.com/juan-cb/paper-ieeeAccess-2017/blob/master/machinelearning-results.ipynb [26]). Thus, only the clusters obtained after finding the best predictive models will be explained (those generated separately per each vertical). As discussed in the methods section, the clusters were generated using hierarchical clustering techniques because these techniques do not require configuring the target number of clusters. This permits all the relevant clusters (relevance due to the Euclidean distance among them) to be obtained regardless of the number. Figures 2, 3, and 4 present the dendrograms corresponding to each set of clusters.

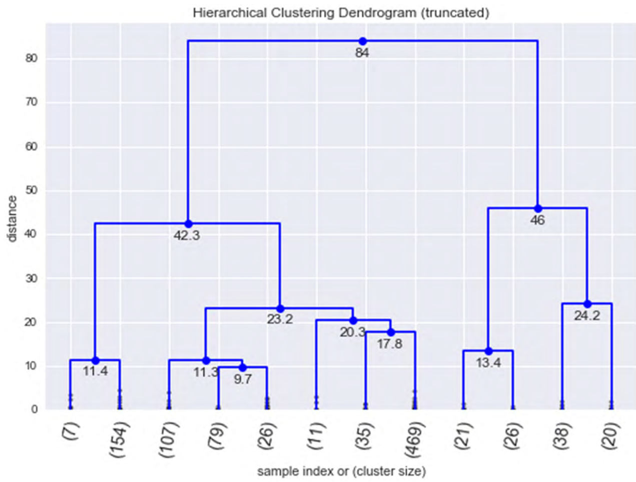


FIGURE 2. Dendrogram representing the clusters found with the predictive model generated using the data from vertical A. Each leaf represents a different cluster obtained (except, in this figure, clusters 8 and 9 that are represented together in the dendrogram due to their closeness in the 9th leaf). The different values that appear near the claves display the Euclidean distance that explains the separation between the different clusters. Finally, the numbers below the leaves (at the bottom of the figure) present the number of users included in the corresponding cluster. Source and full resolution image with all the clusters are available in [26].

After applying the hierarchical clustering algorithm (https://github.com/juan-cb/paper-ieeeAccess-2017/blob/master/machinelearning-results.ipynb [26]) the following numbers of clusters were found: 13 clusters for the vertical A predictive model, 12 clusters for the vertical B model, and 12 clusters for the vertical C.

Analyzing the generated clusters, the researchers found the features that define each cluster and compared them among

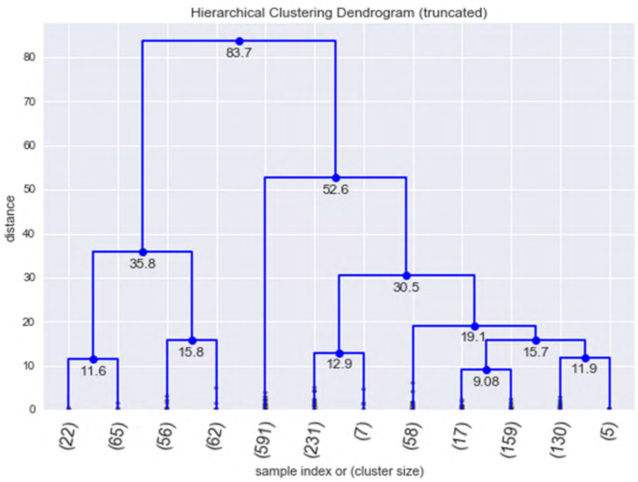


FIGURE 3. Dendrogram representing the clusters found with the predictive model generated using the data from vertical B. The meaning of the different visual elements is the same than those presented in Fig 2. Source [26].

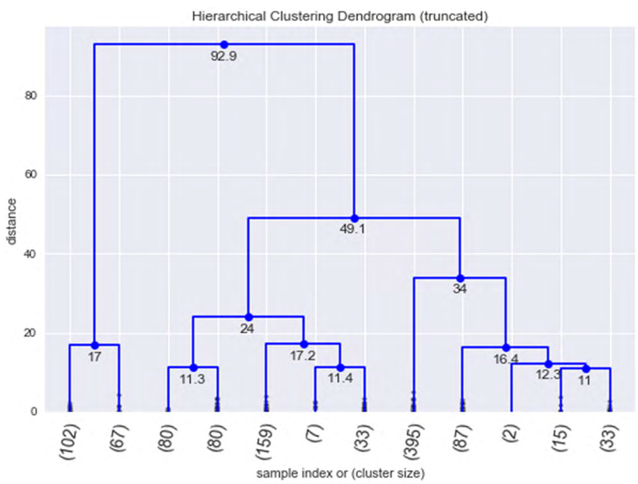


FIGURE 4. Dendrogram representing the clusters found with the predictive model generated using the data from vertical C. The meaning of the different visual elements is the same as those presented in the previous dendrogram figures. Source [26].

the different models to define the redirection rules. This analysis of clusters and rule generation will be explained in the following subsection.

B. RESULTS REGARDING CRITERIA FOR REDIRECTING USERS WITHIN A/B TESTING VERTICALS

Once the clusters were identified through the produced predictive models, the researchers started to analyze the features of each cluster to establish the proper redirection rules based on the heuristics observed. In the case of this study, these rules were not generated automatically, although using the code and procedures previously presented, it would be possible. The results achieved at this stage correspond to those expected in mark (2) in Figure 1.

First, the most important values of these features were obtained through descriptive statistics and distribution plots

```
Cluster 8 || feature: os_Windows
count    395.0
mean       1.0
std        0.0
min        1.0
25%        1.0
50%        1.0
75%        1.0
max        1.0
Name: os_Windows, dtype: float64
Mean of feature :os_Windows: 1.0
```

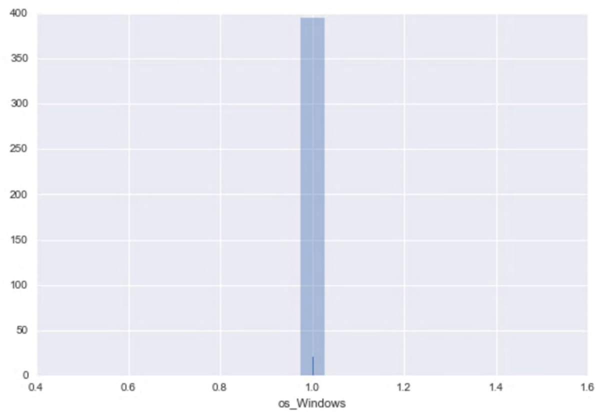


FIGURE 5. Descriptive statistics and distribution of values for cluster 8 (vertical C), regarding the use of the Windows operating system. Source [26].

(for every identified cluster), as included in [26]. As an example of the features' identification, Figure 5 shows that in vertical C's 8th cluster, the device's operating system of the clustered users is Windows (the most repeated value is 1, i.e., *True*). With this information (and the rest of information obtained through the same process on the rest of features) the researchers could determine the possible devices used by the students in every cluster. In this case, the authors will refer mainly to these factors as *technical features* or *technical info*, as the factors were all related to the technological aspects of the device and software used by users completing the questionnaires.

The descriptive statistics and distribution plots for every technical feature within each cluster are available at https://github.com/juan-cb/paper-ieeeAccess-2017/blob/master/machinelearning-results.ipynb [26].

Once the values (technical specs mainly) of the devices were obtained, the finalization rates of the questionnaires of all clusters were calculated, identifying the performance achieved by users in each of them. This allowed the identification, for example, of the clusters whose finalization rate were smaller than the finalization rate of the whole questionnaire vertical.

In this way, researchers identified the factors (the most relevant features of each vertical's predictive model) linked to the clusters that performed worse than the rest. This information is summarized in Tables 12, 13, and 16 for verticals A, B, and C, respectively.

These tables (12, 13, and 14) helped the researchers to define the redirection rules. For example, Android devices with a 2-pixel ratio (i.e., Android devices with good screen

TABLE 12. Cluster characteristics identification in vertical A. Clusters that performed below the general completion rate of the vertical are marked in red.

Vertical	Vertical completion rate		Total users		Completed questionnaires			Uncompleted questionnaires		
A	76.23% (average)		993		757			236		
Cluster number	**Users count**	**Completion rate**	**Viewport width**	**Tablet or mobile?**	**iOS?**	**Screen height (px)**	**Screen width (px)**	**Pixel ratio**	**Android?**	**Possible device**
1	7	71%	2569	False	False	1440	1560	1 or 2	False	Windows computer
2	154	86.36%	1920	False	False	1080	1920	1	False	Windows computer
3	107	83.17%	1440	False	False	900	1440 or 1600	1	False	Windows computer
4	79	79%	1260	False	False	1024	1280	1	False	Windows computer
5	26	80%	1250	False	False	1080	1800	1	False	Windows computer
6	11	81.81%	896 or 1280	False	False	800 or 1024	896 or 1280	1	True	Convertible device
7	35	82.85%	1290	False	False	800 or 900	1280 or 1440	2	False	Retina Mac computer
8	35	80%	1024	False	False	768	1024	1	False	Windows computer
9	434	82.02%	1366	False	False	768	1366	1	False	Windows computer
10	21	9%	366	True	False	640	360	3 or 4	True	Android mobile (very high resolution)
11	26	15%	360	True	False	600 or 700	360	2	True	Android mobile (good resolution)
12	38	2%	500–400	True	True	600	375	2 or 3	False	iPhone
13	20	84.99%	768 or 1024	False	True	1024	768	1 or 2	False	iPad

TABLE 13. Cluster characteristics identification in vertical B. Clusters that performed below the general completion rate of the vertical are marked in red.

Vertical	Vertical completion rate		Total users		Completed questionnaires		Uncompleted questionnaires	
B	66.5% (average)		1403		933		470	
Cluster number	**Users count**	**Completion rate**	**Viewport height (px)**	**Viewport width (px)**	**Screen height (px)**	**Pixel ratio**	**Pixel ratio**	**Possible device**
1	22	72.72%	628	414	736	3	False	Large iPhone (iPhone 6 Plus, 6s Plus or iPhone 7 Plus)
2	65	1.5%	450–500 or 550–600	320 or 375	480, 568 or 667	2	False	iPhone
3	56	8.9%	550	360	640	2	True	Android mobile (good resolution)
4	62	11.29%	537	360	640	3-4	True	Android mobile (very high resolution)
5	591	75.8%	649	1366	768	1	False	Windows computer
6	231	73.5%	955	1860	1080	1	False	Windows computer
7	7	71.42%	1290	1960	1440	1	False	Non-retina Mac computer
8	58	77.58%	720	1134	800-900 or 1024	2	False	iPad
9	159	76.72%	620	1260	1024, 1080 or 1200	1	False	Windows computer
10	17	76.47%	780	1440 or 1600	900	1	False	Windows computer
11	130	74.61%	894	1260	1024	1	False	Windows computer
12	5	80%	936 or 1144	768-800	1024 or 1080	1	True	Android tablet

resolution), despite their low rate performance, obtain better finalization ratios in vertical A (finalization rate of 15%) than in verticals B and C (finalization rates of 8.9% and 10.7%, respectively), leading to the conclusion that the users with devices that meet these characteristics should be redirected to vertical A.

Repeating this methodology for every device identified, the following rules were obtained:

TABLE 14. Cluster characteristics identification in vertical C. Clusters that performed below the general completion rate of the vertical are marked in red.

Vertical	Vertical completion rate		Total users		Completed questionnaires			Uncompleted questionnaires		
C	65% (average)		1060		689			371		
Cluster number	**Users count**	**Completion rate**	**Screen width (px)**	**Viewport height (px)**	**Screen height (px)**	**Tablet or mobile?**	**Viewport width (px)**	**Pixel ratio**	**Windows?**	**Possible device**
1	102	10.7%	360	550	640	True	360	2	False	Android mobile (good resolution)
2	67	20.84%	360	570	640	True	370	3	False	Android mobile (very high resolution)
3	80	71.25%	1280	895	1024	False	1280	1	True	Windows computer
4	80	77.5%	1440, 1600 or 1920	760	900	False	1440 or 1600	1	True	Windows computer
5	159	72.95%	1920	950	1080	False	1920	1	True	Windows computer
6	7	71.42%	2560	1240	1440	False	1892	1	False	iMac
7	33	72.72%	1920	928	1080	False	1700	1	False	Non-retina Mac computer
8	395	76.96%	1366	645	768	False	1366	1	True	Windows computer
9	87	71.26%	1350	670	800–900	False	1280-1300	1	False	Non-retina Mac computer
10	2	50%	1080	500	1848	True	360	3	False	Android tablet
11	15	66.66%	768	950	1024	False	768	1 or 2	False	Mac computer
12	33	69.69%	1148	1280	800 or 1024	False	1280	2	False	Retina Mac computer

- Redirection to vertical A:
  - Android devices with a 2-pixel ratio.
  - Computers with an operating system different from Android, iOS and Mac OS.
  - Mac OS computers.
  - iPad devices.
  - Convertible devices (those that could be used as tablet or as laptop depending on whether a keyboard or mouse is attached to them).
- Redirection to vertical B:
  - Android devices with a 3- or 4-pixel ratio.
  - Large iPhone devices (iPhone 6 Plus, 6s Plus, or 7 Plus).
  - Android tablets.

If the devices of the users who participate in the reinforcement (validate) phase did not meet any of these characteristics, the redirection was randomly made between verticals A and B (maintaining a 50% distribution).

No users were redirected to vertical C due to the low finalization rates of the clusters in this questionnaire variant. There was only one rule that did not follow this assumption: the case of an Android device with a very high resolution (a pixel ratio of 3 or 4). Despite this case, the researchers decided to close this vertical C, as all the mobile or tablet devices with a very high resolution (like iPhone 6 Plus, 6s Plus, 7 Plus, or Android tablets) work better in vertical B.

The final established heuristic rules were the following (presented as a kind of pseudocode):

1. If the operating system is Android and the device's pixel ratio is 2, the user is redirected to vertical A.
2. If the operating system is Android and the device's pixel ratio is 3 or 4, the user is redirected to vertical B.
3. If the operating system of the device is iOS and its pixel ratio is 3 (iPhone 6 Plus, 6s Plus, or 7 Plus), then the user is redirected to vertical B.
4. If the operating system is neither Android nor Mac OS, iOS, the user is redirected to vertical A.
5. If the operating system of the device is Mac OS, the user is redirected to vertical A.
6. If the operating system is Android and the device's screen height is greater than 1000px, the user is redirected to vertical B.
7. If the operating system is iOS, the device's screen width is 1024px, the device's screen height is 768px, and the device's pixel ratio is 1 or 2 (iPad), the user is redirected to vertical A.
8. If the device's operating system is Android and the device type is neither a mobile nor a tablet (convertible device), the user is redirected to vertical A.
9. If a device does not fit any of the previous conditions, the user is randomly redirected to vertical A or B (with equal probability of being redirected to any of them).

These rules were implemented in the OEEU's ecosystem to apply them whenever a new user enters or resumes the questionnaire.

C. RESULTS REGARDING ADAPTABILITY AND USERS' REDIRECTION WITHIN A/B TEST VERTICALS

After the experiment took place (analyzing the interaction and performance of users who used the system previously), all the users who entered or returned to the questionnaire (and therefore, the target users of the experiment) were sought to obtain the results regarding the application of redirection criteria within the questionnaire verticals. The calculation and validation presented in this phase correspond to the (3) mark in Figure 1.

Before this phase (called reinforcement because the participants are users who access the web forms in a reinforcement made by the OEEU to obtain more responses to the questionnaires) and the application of the redirection rules based on heuristics, 5768 users had started the questionnaire; 4410 of them finished it, leaving a total of 1358 uncompleted questionnaires (and reaching a completion rate of 76.46%). All the data related to this subsection are available at https://github.com/juan-cb/paper-ieeeAccess-2017/blob/master/reinforcement-results.ipynb [26]

In these previous results, the users who *entered* the questionnaire (i.e., reached the welcome page but never started it) were not taken into account. If these users were considered, the results would be as follows:

- Number of students who have *entered* the questionnaire: 6360.
- Number of students who have *not finished* the questionnaire: 1950.
- Number of students who have *finished* the questionnaire: 4410.
- Completion rate *before* reinforcement: 69.34%.

By the time the questionnaires were closed, the final results were the following: 6738 started questionnaires, of which 5214 were completed and 1524 uncompleted. Consequently, the study achieved a questionnaire completion rate of 77.38%, improving the previous rate.

Again, these are the results for the started questionnaires; considering all the users (including the ones who reached the welcome page), the study yields the following results:

- Number of students who have *entered* the questionnaire: 7349.
- Number of students who have *not finished* the questionnaire: 2135.
- Number of students who have *finished* the questionnaire: 5214.
- Completion rate *after* reinforcement: 70.95%.

The total number of target users who entered the questionnaire after the incorporation of the system redirection support was 1165. These 1165 users were classified into three groups:

- Users who *entered* the questionnaire *after* reinforcement (considered as "new users"). There were 1003 new users, becoming the larger group of users who have taken part in the experiment.
- Users who resumed the questionnaire *after* reinforcement and were redirected to a different vertical; 110 users satisfied this criterion.
- Users who resumed the questionnaire *after* reinforcement but were *not* redirected to a different vertical. There were 52 users of this type.

These general results are summarized in Table 15.

TABLE 15. General results in the reinforcement phase.

User type	Total	Results	Completion rate
New users	1003	718 finished questionnaires 285 *not* finished questionnaires	71.59%
Redirected users	110	61 finished questionnaires 49 *not* finished questionnaires	55.45%
Not redirected users	52	25 finished questionnaires 27 *not* finished questionnaires	48.08%

As can be seen in Table 15, the new users' sample reached a completion rate of 71.59%.

This sample includes users who (at least) reached the welcome page of the questionnaire after reinforcement. An improvement in the completion results could be seen when comparing this completion rate (71.59%) with the completion rate before the reinforcement (that includes all the users who entered the questionnaire, 69.34%). Furthermore, it is necessary to consider that these new users are more reluctant in completing the questionnaire, as they have been invited to participate at least twice previously (and they had ignored the invitations), so these results are even more valuable.

Once the participant finalization rates were calculated, the researchers proceeded with the analysis of the impact of the rules formulated to improve the finalization rate, taking as a reference the groups of users who accessed the questionnaire presentation page both before and after the reinforcement phase.

These users were grouped into categories according to the way in which they were assigned to their vertical. To generate these categories, the researchers applied the assignment rules to the group of users who participated prior to the reinforcement and compared the results (ideal vertical assignment) with the vertical to which these individuals were actually sent (actual vertical assigned). Thus, the following three groups of individuals were obtained:

- **Pre-reinforcement users randomly assigned to the wrong vertical (G1, n = 3833):** Composed of users who

TABLE 16. Correlation between the vertical assignment and the finalization rate.

	Finalization rate	Chi-squared	Significance
G1-G2	67.9-74.9	25.927	0.000
G2-G3	74.9-71.6	3.442	0.064
G1-G3	67.9-71.6	5.130	0.024

accessed the questionnaire before the reinforcement and were assigned to a vertical to which they would not have been assigned had the redirection rules been applied.

- **Pre-reinforcement users randomly assigned to the right vertical (G2, n = 1542):** Comprised of users who accessed the questionnaire before the reinforcement and who, despite having been randomly directed, were assigned to the vertical to which they would have belonged to, had the redirection rules been applied.
- **Post-reinforcement users (G3, n=1003):** Users who accessed the questionnaire for the first time after the reinforcement, thus being consequently assigned to the right vertical.

In the case of rule 9, researchers classified all individuals who were randomly directed to vertical C as members of group 1; individuals who were directed to verticals A or B were classified as missing values, as the distribution of those verticals was defined differently from the one defined for the reinforcement phase.

Once the users were classified, the researchers calculated the finalization rate of each group, using the Chi-square statistic to study whether the vertical assignation method influenced the finalization rate. The Chi-square test is the most reliable in this scenario, given that there are two categorical variables (questionnaire finalization and success in the assignment). This statistical test was applied to the three possible combinations of pairs (Table 16).

First, as we can observe in the table, the results of the Chi-square test reflect a significant correlation between the vertical assignation method and the finalization of the questionnaire in pair G1–G2 for a significance level (s.l.) of 0.001. This result is consistent with the methodology employed, given that the clustering process and the later rules of assignment were carried out using the pre-reinforcement users.

Second, for the pair G2–G3, the results indicate no correlation between the assignment method and the finalization rate (s.l. 0.05) which, again, confirms the adequacy of the established rules, as individuals in group 3 were grouped with the same criterion that those in group 2, although the assignment was done in an intentional way rather than randomly.

Finally, it is noticeable that there is also a correlation (s.l. 0.05) between the assignment method and the finalization rate in the case of the pair G1–G3, which confirms that the application of the established rules significantly contributes to the finalization of the questionnaire by the participants.

TABLE 17. Correlation between the application rule and the finalization rate.

	Finalization rate		N		Chi-squared	Significance
	G1	G3	G1	G3		
Rule 1	67.11	72.64	374	106	1.167	0.280
Rule 2	70.99	72.22	362	126	0.069	0.793
Rule 3	62.75	56.52	51	23	0.258	0.612
Rule 4	68.03	76.25	2196	421	11.215	0.001
Rule 5	71.69	73.47	325	49	0.067	0.796
Rule 6	*	*	*	*	*	*
Rule 7	67.12	74.07	73	27	0.445	0.505
Rule 8	72.00	80.00	25	10	0.643**	0.488**
Rule 9	62.53	63.07	427	241	0.019	0.890

*No individuals in group 2. **Fisher's exact test (odds ratio and p-value)

As a final data analysis step, the researchers carried out an in-depth study of the behavior of each of the proposed rules, aiming to delve into the individual effect of each of them on the finalization of the questionnaire.

To this end, a process like the previous analysis was used with each one of the rules, the difference being that only the pair G1–G3 was used (Table 17).

As illustrated in Table 16, although there are differences in all finalization rates, they are significant (s.l. 0.01) only in the case of rule 4. For said rule, the rate of finalization in group 3 is approximately 8% greater than the rate in group 1, which suggests that directing the individuals who access the questionnaire from a non-Mac PC improves their chances of completing the questionnaire.

IV. DISCUSSION

This section presents the discussion of all the issues found in the research, discussing the foundations and effects of some decisions made by the authors. It also includes several future lines of work, suggests a set of recommendations, and closes with a general conclusion.

A. GENERAL DISCUSSION

Regarding the research carried out by the authors, there are several issues to comment on in this paper. To facilitate the comprehension, these issues will be discussed following the same structure of the paper (first, issues related to the methodology; second, those related to the results, and so forth).

First, the authors pose a question: Is it advisable to apply this kind of machine-learning method to this kind of problem? In this case, the researchers were inspired by other authors who have applied these types of processes to a wide range of problems. As an example, this kind of machine-learning algorithmic approach has being used in other fields, such as education [35], with promising results. Beyond the benefits that machine-learning approaches bring to many problems, by also including white-box procedures, the researchers ensure explainable and reproducible results that could be improved or discussed by the scientific community. All these considerations and precedents encouraged the authors to employ this kind of approach to address the problem of improving users' performance within a complex system like that presented. According to the results, the question can be answered positively, as the findings have been valuable and prove the validity of the approach.

Following the discussion, the authors would like to comment that the A/B testing approach used for this research is not a *pure* application of such methodology. While A/B tests are commonly based on singular changes between the different experimentation groups (or verticals), in the presented approach the authors grouped different changes into the same verticals. In this case, this variation of A/B tests does not influence this experimentation, as the researchers attempt to maximize user performance in the questionnaire finalization without a special focus on small changes, but using important differences between the different verticals. Despite that, it is worth noting that this kind of application of A/B tests for the experiment has been previously validated by experts [29].

Regarding the generated predictive models, the cut-off value for their relevant factors to later include in the clusters, the authors stated 0.05 as the minimum value to consider since this is the most common value in classical literature to ensure reliable results. Also in this case, the authors use this cut-off value to generate the clusters using only the most important factors (those that have a specific weight of more than 0.05 in the predictive model), thus excluding less important ones that could introduce noise when building the groups.

Concerning the most important factors that characterize the predictive models and explain the users' profile and preferences while completing the questionnaire, it should be remarked that technical aspects were more important than personal ones. At the beginning of the research and for the predictive models' generation, researchers included personal aspects, such as gender, age, and issues related to education, as part of the dataset. According to the results, such aspects do not have special relevance while modeling the users' behavior in completing the web form. Instead, the present findings indicated that the most important factors for the users were the size of the device screen and the browser window. Moreover, other aspects, like the screen resolution, concrete browser, or operative system, were important, but with a lesser effect. Nevertheless, these are the most important factors for the population of this study and cannot be considered general and valid for other populations. To apply the approach presented in this research in other experiments, the predictive models should be generated again.

Regarding the generation of rules based on heuristics, and as a future study, the researchers would like to automate this process. This will help to reproduce the same process with the same experimental conditions and remove any kind of bias introduced by researchers or administrators. This will be explained in depth in the following subsection.

Related to the reinforcement phase and other conditions of the experiment, with the aim of enhancing users' participation in the questionnaires, the OEEU offered participation in a raffle (the prize would be seven smartwatches) to all graduates completing the web form as a reward. This incentive was used also to promote the reinforcement process where the redirection rules were applied.

Regarding the effectiveness of the use of rules based on cluster analysis during the reinforcement period, cluster analysis was found to be a very useful tool to guide the redirection of users to the version of the questionnaire best suited to the features of the technology with which they completed it.

First, the results of this study confirm that the rules established improved the answer rate by comparing the performance of users who participated after the reinforcement with those who participated before the last reinforcement and were directed to the wrong questionnaire. Additionally, the authors could observe that there are no significant differences between groups G2 and G3, which leads to the understanding that the application of the rules during the reinforcement has maintained the good results regarding to the finalization among the users who would have been randomly assigned to the right vertical.

Second, if the researchers delve into the analysis of the individual behavior of each rule, the results suggest that the improvement in the finalization rate is due to rule four, which redirects users who access the form from non-Mac computers to vertical A, given that the rest of rules have not yielded significant correlations.

Regarding this point, it must be remarked that the users who participated in a reinforcement phase were commonly more reluctant to complete the questionnaire, as they left it in previous stages or were not initially attracted to fulfill it. This also could render even more valuable the results obtained in this research concerning the improvement of users' performance. However, for future studies it would be interesting to apply a research design that includes an experimental and a control group from the beginning to be able to assess the effect of the rules under the same conditions.

Another interesting future line of research would be an analysis of the threshold cut-off to perform the factor selection, given that a higher minimum value may simplify the number of rules and make more efficient the redirection process. As a first step, the authors intend to analyze rule four to gain a better understanding of the predictive importance of the elements behind its formulation.

Finally, the authors believe that the approach and procedures presented in this research are transferable to other application fields. The process presented in this paper follows some *traditional* approaches and methods within the machine-learning research field, and the prediction challenge is present in many other problems beyond web form completion. The proposed methodology may also help to transfer this experience to other problems with the additional value of providing a white-box approach for the algorithms used. In the future, the authors would like to attempt to apply such methodology to predict the employability of Spanish graduates. This will also validate the genericity of the methodology, which will only require some minor changes depending on the dataset.

B. HOW TO APPLY THIS RESEARCH IN PRODUCTION IN THE REAL WORLD

One of the main concerns related to this research could be stated as follows: Is it possible to use this contribution in a real industry setting? Is it possible to integrate this kind of approach in production systems and enable an automated process? From the point of view of the researchers, the answer is yes to both questions. There are many examples in the industry on how data sciences processes can be transformed from Jupyter notebooks to enterprise-ready systems put in production. In this case, the researchers outline the approach proposed by the Airbnb engineering team on how their ML Automator [36] tool helped in translating a Jupyter notebook into an Airflow machine learning pipeline [37] and use this kind of analytics process in production systems. This automating effort must include—apart from the machine-learning algorithms and process—rule generation or the identification of the proper Euclidean distance to separate the clusters generated. To automate the rule generation, probably researchers would have to employ artificial intelligence techniques such as neuronal networks, that could learn to generate these rules as done by humans in this paper.

C. GENERAL CONCLUSION

This paper presents a novel study in the field of Human-Computer Interaction. The main results achieved have been quite promising and encourage authors to continue the labor of improving users' performance in completing large web forms. Adaptability can be achieved by detecting users' behaviors, preferences, and profiles using machine-learning techniques and offering the best user interface and user experience to each kind of user detected. Based on the results, the authors also propose several future works that could push this research to be adopted in the industry and other application fields.

ACKNOWLEDGMENTS

The authors thank Felicidad García-Sánchez for designing Figure 1.

REFERENCES

[1] R. Colomo-Palacios, F. J. García-Peñalvo, V. Stantchev, and S. Misra, "Towards a social and context-aware mobile recommendation system for tourism," *Pervasive Mobile Comput.*, vol. 38, pp. 505–515, Jul. 2017.

[2] D. A. Norman and S. W. Draper, Eds., *User Centered System Design: New Perspectives on Human-Computer Interaction*. Hillsdale, NJ, USA: L. Erlbaum Assoc. Inc., 1986.

[3] D. A. Norman, *The Design of Everyday Things: Revised and Expanded Edition*. New York, NY, USA: Basic Books, 2013.

[4] J. A. Bargas-Avila, O. Brenzikofer, S. P. Roth, A. N. Tuch, S. Orsini, and K. Opwis, "Simple but crucial user interfaces in the World Wide Web: Introducing 20 guidelines for usable Web form design," in *User Interfaces*, R. Matrai, Ed. Rijeka, Croatia: InTech, 2010. [Online]. Available: https://www.intechopen.com/books/user-interfaces/simple-but-crucial-user-interfaces-in-the-world-wide-web-introducing-20-guidelines-for-usable-web-fo, doi: 10.5772/9500.

[5] B. Shneiderman and C. Plaisant, *Designing the User Interface*, 4th ed. Reading, MA, USA: Addison-Wesley, 2005.

[6] M. Seckler, S. Heinz, S. Forde, A. N. Tuch, and K. Opwis, "Trust and distrust on the Web: User experiences and website characteristics," *Comput. Human Behavior*, vol. 45, pp. 39–50, Apr. 2015.

[7] B. Shneiderman, "Designing trust into online experiences," *Commun. ACM*, vol. 43, no. 12, pp. 57–59, Dec. 2000.

[8] S. P. Anderson, *Seductive Interaction Design: Creating Playful, Fun, and Effective User Experiences, Portable Document*. London, U.K.: Pearson Education, 2011.

[9] S. Krug, *Don't Make Me Think: A Common Sense Approach to Web Usability*. Bengaluru, India: Pearson Education, 2000.

[10] D. Malandrino, F. Mazzoni, D. Riboni, C. Bettini, M. Colajanni, and V. Scarano, "MIMOSA: context-aware adaptation for ubiquitous Web access," *Pers. Ubiquitous Comput.*, vol. 14, no. 4, pp. 301–320, May 2010.

[11] G. Pruvost, T. Heinroth, Y. Bellik, and W. Minker, "User interaction adaptation within ambient environments," in *Next Generation Intelligent Environments*. New York, NY, USA: Springer, 2016, pp. 221–263.

[12] C. Flavián, R. Gurrea, and C. Orús, "The effect of product presentation mode on the perceived content and continent quality of Web sites," *Online Inf. Rev.*, vol. 33, no. 6, pp. 1103–1128, 2009.

[13] F. Michavila, J. M. Martínez, M. Martín-González, F. J. García-Peñalvo, and J. Cruz-Benito, "Barómetro de Empleabilidad y Empleo de los Universitarios en España, 2015 (Primer informe de resultados)," Universidad Politécnica Madrid, Madrid, Spain, 2016.

[14] F. Michavila, M. Martín-González, J. M. Martínez, F. J. García-Peñalvo, and J. Cruz-Benito, "Analyzing the employability and employment factors of graduate students in Spain: The OEEU Information System," presented at the 3rd Int. Conf. Technol. Ecosyst. Enhancing Multiculturality (TEEM), Porto, Portugal, 2015.

[15] A. Vázquez-Ingelmo, J. Cruz-Benito, and F. J. García-Peñalvo, "Scaffolding the OEEU's data-driven ecosystem to analyze the employability of Spanish graduates," in *Global Implications of Emerging Technology Trends*, F. J. García-Peñalvo, Ed. Hershey, PA, USA: IGI Global, 2018.

[16] Django Software Foundation. (Mar. 15, 2015). *Django Web Framework*. [Online]. Available: https://www.djangoproject.com/

[17] S. Stieger and U.-D. Reips, "What are participants doing while filling in an online questionnaire: A paradata collection tool and an empirical study," *Comput. Human Behavior*, vol. 26, no. 6, pp. 1488–1495, Nov. 2010.

[18] W. McKinney, *Python for Data Analysis: Data Wrangling With Pandas, NumPy, and IPython*. Sebastopol, CA, USA: O'Reilly Media, Inc., 2012.

[19] W. McKinney. (2017). *Pandas, Python Data Analysis Library*. Accessed: 2017. [Online]. Available: http://pandas.pydata.org/

[20] W. McKinney, "Data structures for statistical computing in python," in *Proc. 9th Python Sci. Conf.*, vol. 445. Austin, TX, USA, 2010, pp. 51–56.

[21] F. Pedregosa *et al.*, "Scikit-learn: Machine learning in Python," *J. Mach. Learn. Res.*, vol. 12, pp. 2825–2830, Oct. 2011.

[22] S. Raschka, *Python Machine Learning*. Birmingham, U.K.: Packt Publishing Ltd., 2015.

[23] T. Kluyver *et al.*, "Jupyter notebooks—A publishing format for reproducible computational workflows," in *Proc. ELPUB*, 2016, pp. 87–90.

[24] M. Ragan-Kelley *et al.*, "The Jupyter/IPython architecture: A unified view of computational research, from interactive exploration to communication and publication," in *Proc. AGU Fall Meeting Abstracts*, 2014.

[25] F. Perez and B. E. Granger, "Project Jupyter: Computational narratives as the engine of collaborative data science," Tech. Rep., 2015. [Online]. Available: https://blog.jupyter.org/project-jupyter-computational-narratives-as-the-engine-of-collaborative-data-science-2b5fb94c3c58

[26] J. Cruz-Benito, A. Vázquez-Ingelmo, and J. C. Sánchez-Prieto. (2017). *Enabling Adaptability in Web Forms Based on User Characteristics Detection Through A/B Testing and Machine Learning*, Code Repository That Supports the Research Presented in the Paper. [Online]. Available: https://github.com/juan-cb/paper-ieeeAccess-2017

[27] Y. Xu, N. Chen, A. Fernandez, O. Sinno, and A. Bhasin, "From infrastructure to culture: A/B testing challenges in large scale social networks," presented at the 21th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining, Sydney, NSW, Australia, 2015.

[28] E. Dixon, E. Enos, and S. Brodmerkle, "A/b testing of a webpage," U.S. Patent 7 975 000, Jul. 5, 2011.

[29] D. Siroker and P. Koomen, *A/B Testing: The Most Powerful Way to Turn Clicks Into Customers*. Hoboken, NJ, USA: Wiley, 2013.

[30] J. Cruz-Benito *et al.*, "Improving success/completion ratio in large surveys: A proposal based on usability and engagement," in *Proc. 4th Int. Conf Learn. Collaboration Technol. Technol. Edu. (LCT)*, 2017, pp. 352–370.
[31] H. Wickham, "Tidy data," *J. Statist. Softw.*, vol. 59, no. 10, pp. 1–23, 2014.
[32] L. Breiman, "Random forests," *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, Oct. 2001.
[33] S. C. Johnson, "Hierarchical clustering schemes," *Psychometrika*, vol. 32, no. 3, pp. 241–254, Sep. 1967.
[34] Scikit-Learn. (Sep. 4, 2017). *API Reference—Scikit-Learn Documentation: Metrics*. [Online]. Available: http://scikit-learn.org/stable/modules/classes.html
[35] A. Zollanvari, R. C. Kizilirmak, Y. H. Kho, and D. Hernández-Torrano, "Predicting students' GPA and developing intervention strategies based on self-regulatory learning behaviors," *IEEE Access*, vol. 5, pp. 23792–23802, 2017.
[36] R. Chang. (Sep. 4, 2017). *Using Machine Learning to Predict Value of Homes On Airbnb*. [Online]. Available: https://medium.com/airbnb-engineering/using-machine-learning-to-predict-value-of-homes-on-airbnb-9272d3d4739d
[37] M. Beauchemin. (Sep. 4, 2017). *Airflow: A Workflow Management Platform*. [Online]. Available: https://medium.com/airbnb-engineering/airflow-a-workflow-management-platform-46318b977fd8

JUAN CRUZ-BENITO received the M.Sc. degree in intelligent systems from the University of Salamanca, Spain, in 2013, where he is currently pursuing the Ph.D. degree in computer sciences. He is one of the youngest members of the Research Group Interaction and eLearning, where he specializes in software solutions based on technology ecosystems and open source software. He is involved in human–computer interaction, educational virtual worlds and technologies for educational purposes, and disciplines that he has developed in many innovation and research projects. He has participated in many European and national research and development projects, such as TRAILER, VALS, USALSIM Virtual Campus, and the Spanish Observatory for University Employability and Employment, as a Software Engineer, Researcher, and Developer.

ANDREA VÁZQUEZ-INGELMO was born in Salamanca, Castilla y León, Spain, in 1994. She received the bachelor's degree in computer engineering from the University of Salamanca, Salamanca, in 2016, where she is currently pursuing the master's degree in computer engineering. She is a member of the Research Group of Interaction and eLearning. Since 2016, she has been a part of the National Project Spanish Observatory for University Employability and Employment as a Developer and a Researcher.

JOSÉ CARLOS SÁNCHEZ-PRIETO received the bachelor's degree in pedagogy and the master's degree in ICT applied in education from the University of Salamanca, Spain. He is currently pursuing the Ph.D. degree with the Faculty of Education, Said University, within the Program on Education in the Knowledge Society. His area of research is the assessment of attitudes among in-service and pre-service teachers.

ROBERTO THERÓN received the Diploma degree in computer science from the University of Salamanca, the B.A. degree from the University of A Coruña, the bachelor's degree in communication studies and the bachelor's degree in humanities from the University of Salamanca, and the Ph.D. degree from the Research Group Robotics, University of Salamanca. His Ph.D. thesis was on parallel calculation configuration space for redundant robots. He is currently the Manager of the VisUsal Group (within the Recognized Research Group GRIAL), University of Salamanca, which focuses on the combination of approaches from computer science, statistics, graphic design, and information visualization to obtain an adequate understanding of complex data sets. He has authored of over 100 articles in international journals and conferences. In recent years, he has been involved in developing advanced visualization tools for multidimensional data, such as genetics or paleo-climate data. In the field of visual analytics, he develops productive collaborations with groups and institutions internationally recognized as the Laboratory of Climate Sciences and the Environment, France, or the Austrian Academy of Sciences, Austria. He received the Extraordinary Doctoral Award for his Ph.D. thesis.

FRANCISCO JOSÉ GARCÍA-PEÑALVO received the degrees in computing science from the University of Salamanca and the University of Valladolid, and the Ph.D. degree from the University of Salamanca. He is currently the Head of the Research Group Interaction and eLearning. He was the Vice Chancellor of Innovation with the University of Salamanca from 2007 to 2009. He has led and participated in over 50 research and innovation projects. He has authored over 300 articles in international journals and conferences. His main research interests focus on eLearning, computers and education, adaptive systems, Web engineering, semantic Web, and software reuse. He is a member of the Program Committee of several international conferences and a reviewer of several international journals. He was a Guest Editor of several special issues of international journals, such as *Online Information Review*, *Computers in Human Behavior*, and *Interactive Learning Environments*. He is currently the Editor-in-Chief of the *International Journal of Information Technology Research* and the *Journal of Education in the Knowledge Society*. He is also the Coordinator of the multidisciplinary Ph.D. Program on education in the Knowledge Society.

MARTÍN MARTÍN-GONZÁLEZ received the master's degree in economic development and public policy and the Ph.D. degree in economics from the Autonomous University of Madrid (UAM), and the degree in economics from the University of La Laguna. He was a Researcher with the Faculty of Economics, UAM. He is currently a Researcher with the UNESCO Chair in University Management and Policy, Technical University of Madrid. His areas of research are the economics of education, the economics of higher education, employability, the evaluation of educational policies, public economics, applied economics, economic development, higher education, and vocational training.

7.5 Appendix E. Domain engineering for generating dashboards to analyze employment and employability in the academic context

Vázquez-Ingelmo, A., García-Peñalvo, F. J., & Therón, R. (2018). Domain engineering for generating dashboards to analyze employment and employability in the academic context. In F. J. García-Peñalvo (Ed.), TEEM'18 Proceedings of the Sixth International Conference on Technological Ecosystems for Enhancing Multiculturality (Salamanca, Spain, October 24th-26th, 2018) (pp. 896-901). New York, NY, USA: ACM. doi:10.1145/3284179.3284329

Domain engineering for generating dashboards to analyze employment and employability in the academic context

Andrea Vázquez-Ingelmo
GRIAL Research Group, Department of Computer Science, Research Institute for Educational Sciences. University of Salamanca
Paseo de Canalejas 169, 37008
Salamanca, Spain
andreavazquez@usal.es

Francisco J. García-Peñalvo
GRIAL Research Group, Department of Computer Science, Research Institute for Educational Sciences. University of Salamanca
Paseo de Canalejas 169, 37008
Salamanca, Spain
fgarcia@usal.es

Roberto Therón
GRIAL Research Group, Department of Computer Science, Research Institute for Educational Sciences. University of Salamanca
Paseo de Canalejas 169, 37008
Salamanca, Spain
theron@usal.es

ABSTRACT

Data analysis is a key process to foster knowledge generation regarding particular domains or fields of study. With a strong informative foundation derived from the analysis of collected data, decision-makers can make strategic choices with the aim of obtaining valuable benefits in their specific areas of action. However, given the steady growth of data volumes, data analysis needs to rely on powerful tools to enable knowledge extraction. Dashboards offer a software solution for visually analyzing large volumes of data in order to identify patterns and relations and make decisions according to the presented information. But decision-makers may have different goals and, consequently, different necessities regarding their dashboards. Having a methodology to efficiently generate dashboards taking into account differing needs would add a customization layer to allow particular users to reach their own goals. This approach can be achieved through domain engineering and automatic code generation processes. This paper presents the application of domain engineering within the dashboards' domain through a case study in the context of the Spanish Observatory for University Employment and Employability, in which a set of dashboards can be generated to exploit different perspectives of employment and employability data in the academic context.

CCS CONCEPTS

• **Software and its engineering → Reusability** • **Human-centered computing → Visualization toolkits**

KEYWORDS

Domain engineering; software product lines; information dashboards; information systems; university employment; university employability.

1 INTRODUCTION

Informed decision-making processes have gained relevance over the years given the potential benefits of using data for building strong informative foundations [1]. Data collection is a crucial stage in data-driven [2] activities, but until its analysis, data has no real value. The analysis of the collected data opens up the possibility of generating knowledge from it and, consequently, to improve and obtain benefits from the execution of strategic choices [3]. However, the introduction of information systems to support a great diversity of processes has caused an exponential growth of generated data.

This situation has led to the necessity of powerful tools for managing and analysing large volumes of collected data in order to support and ease knowledge generation processes.

Information dashboards are one of the most commonly used software tools to explore data in an interactive and friendly way [4], providing a solution for visually analysing datasets and identifying relevant factors or patterns at a glance.

There are certain fields of study, like employability, that could take advantage from these tools. This research area has not yet a strong theoretical foundation given the complexity of acquiring influential indicators to evaluate it. In addition, several variables need to be taken into account in order to obtain a wide and complete view of this field: from identifying different skills that individuals could need in their careers to sociodemographic variables [5].

Having technological support to explore employability and employment data (through information dashboards, for example) could ease the recognition of relevant or influential factors within the domain. What is more, the study of these fields in an academic context can help to reach insights about the linkage between university training and the career path of the graduates.

Universities (as they have a key role related to the employability of their students) can benefit from the introduction of information dashboards to conduct well-informed knowledge management [6, 7] and to support decision-making processes. Specifically, policymakers and institutions can improve and promote identified factors that affect the employability and employment of the students, placing these processes in the context of emergent areas like the Academic Analytics [8, 9] or Institutional Intelligence [10, 11].

However, developing information dashboards is not a trivial task. Several requirements can be involved and can vary among the different user profiles that would use the dashboard.

It is important to take into account all requirements and necessities in order to provide a good user experience to enable knowledge generation.

Software engineering paradigms like software product lines (SPL) [12, 13] provide solutions for managing sets of differing requirements, focusing on the reutilization and composition of base software assets (also known as core assets) to improve scalability and maintainability of particular products that share commonalities.

This paper describes the application of domain engineering to obtain a software product line of dashboards for analyzing university employment and employability data, in the context of the Spanish Observatory for University Employment and Employability (known in Spanish as OEEU, https://oeeu.org/). This organization has the vision to become an information reference for understanding the behavior of the variables related to employment and employability of students from Spanish universities through the recollection of data from the Universities' administrative records and the students themselves [14, 15]. The variety of users that consume information from the Observatory makes the SPL paradigm a potential solution for developing customized information dashboards to explore its data.

The rest of this work is organized as follows. Section 2 describes the methodology followed to develop the dashboard software product line for the Observatory. Section 3 presents the results of the application of the SPL paradigm within the dashboards domain. Section 4 discusses the results obtained, followed by the final section (Section 5), where the conclusions derived from this work are presented.

2 METHODOLOGY

2.1 Domain engineering

The main stage of the SPL paradigm is called domain engineering. In this stage, the domain of the product line to be developed is studied in order to identify the commonalities and variability points of the products that will conform the product family.

By identifying these characteristics, it is possible to model the SPL through a feature diagram [16] which allow the developers to generate the final products through the combination of the different features identified.

During the domain engineering stage, a series of core assets (software assets that will conform the foundation of the product line) are implemented. These core assets are generally configurable, which means that developers will be able to reuse and adapt them given specific requirements of particular users.

For the pilot product line of dashboards presented in this paper, three components have been modeled. These components are three kind of information visualizations that could have different data sources, functionalities or even layout: a scatter diagram, a heat map and a chord diagram, as shown in Figure 1. The generated dashboards need support from the base system of the Observatory to manage user permissions and data retrieval from the persistent storage.

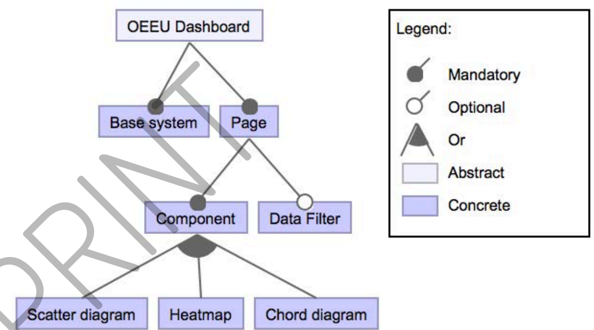


Figure 1. Top-level feature model of the SPL for the Observatory's dashboard product line. The individual features of every individual component have been omitted for simplicity

Once the core assets for these components have been implemented, developers are able to combine and configure them with different parameters to build dashboards pages that fit particular requirements or necessities.

2.2 Code generation

Once the domain engineering phase is done and a set of core assets is available, the application engineering stage starts. In this stage particular products of the line are generated through the combination and configuration of software components.

This process can be automated through code generators fueled by configuration files, obtaining the source code adapted to the specified requirements.

There are several strategies to materialize the variability points previously modeled in the domain engineering phase [17]. One of these strategies is to implement the core assets as a series of code templates [18, 19] that will be filled once the requirements for the product are defined.

In this case, a domain specific language (DSL) has been implemented with XML technology [20] to provide a structured file to the code generator so it can easily extract the features and inject them into the code templates (with Jinja2 [21] as the template engine). This code generator is implemented in Python, and it is in charge of processing the XML configuration files and inject the functionalities through the code templates logic.

The outcomes of this process are the source files (HTML and JavaScript files) that will be deployed to make them accessible to the final users, conforming the personalized information dashboard. The general workflow followed to generate the source code can be seen in the Figure 2.

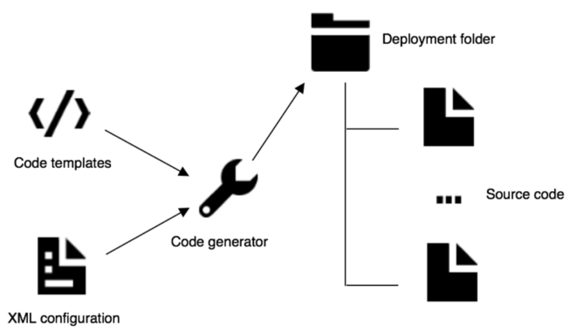


Figure 2. Code generation workflow.

2.3 Interoperability

The backbone of an information visualization is the data showed. One of the main concerns of this approach applied to the dashboards domain is the interoperability of the data sources that will fuel the information visualizations.

Having good interoperability levels means that any modification on the data sources would not imply several or critical changes in the visualizations' code.

To retrieve the data to be presented from the Observatory's bank of knowledge, a GraphQL API [22] has been implemented in order to decouple the presentation components from the persistent storage.

GraphQL provides a flexible query language to build requests that can be parameterized. What is more, GraphQL allows to specify the fields that will be retrieved, saving bandwidth by including only the necessary data in the responses [23, 24].

The API calls are executed by specific GraphQL connectors implemented to add an abstraction layer and to be able to modify (if necessary) the data requests without affecting the actual components or even change the data sources.

3 RESULTS

The following section presents the results derived from the application of the SPL paradigm to create a software product line for the Observatory in order to generate customized dashboards. The results are presented through three perspectives: the results obtained regarding the customization of functionalities, layout and data sources.

3.1 Results regarding functional personalization

One of the main goals of this approach is to provide an automated method to manage the generation of products with different functionalities.

Having the possibility to change the components' functional features gives freedom to the users to choose and obtain components that fit best their requirements.

For example, a potential user could need a scatter diagram to explore three variables at the same time through the X and Y axes and the radius of the points being represented, in order to study potential patterns or relations among them.

On the other hand, another particular user could only need to explore two variables at the same time, because the addition of a third variable could aggravate the analysis process.

By only changing the configuration of a scatter diagram component it is possible to achieve (through the specific core asset that supports this component) two different versions of the diagram to fit these particular requirements (an example is showed in the Figure 3).

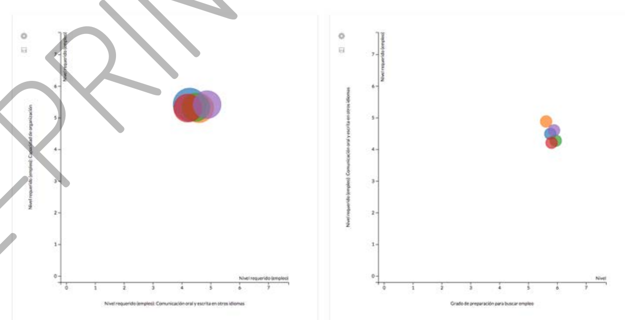


Figure 3. Comparison of two scatter diagram components. On the left, the radius of the elements within the diagram represents the value of a variable. On the right, the radius is not used to represent any variable and every point has a default radius.

There are other kind of features that can be configured, like a set of controls or filters available to explore data with more or less detail or freedom given the final user necessities.

All these variations are introduced through the templates previously implemented; functionalities are injected or ignored on the core assets to produce components with different features from the same template.

3.2 Results regarding layout personalization

This approach also allows the customization of the dashboards pages' layout. A generated dashboard page will be composed by a series of selected components previously configured to fit particular requirements. Once this task is done, these components need to be placed on the final dashboard page.

From an abstract point of view, a dashboard page is composed by containers (rows or columns) that will hold the different components or graphical resources.

Through the implemented DSL it is possible to specify the page layout once the components have been configured by referencing them within the containers that will hold them. There is an example of this syntax on the Figure 4.

```xml
<Page page_id="1">
    <Components>
        <Component>
            <ScatterDiagram component_id="ScatterDiagram_1"...>
        </Component>
        <Component>
            <Heatmap component_id="HeatMap_1"...>
        </Component>
        <Component>
            <ChordDiagram component_id="Chord_1"...>
        </Component>
    </Components>
    <Layout>
        <RowGroup>
            <Row width="100%" height="100%">
                <ColumnGroup>
                    <Column width="100%" height="100%">
                        <Component ref="Chord_1"/>
                    </Column>
                    <Column width="100%" height="100%">
                        <Component ref="ScatterDiagram_1"/>
                    </Column>
                </ColumnGroup>
            </Row>
            <Row width="100%" height="100%">
                <Component ref="HeatMap_1"/>
            </Row>
        </RowGroup>
    </Layout>
</Page>
```

Figure 4. Example of a configuration for a dashboard page in which the layout is specified in terms of rows and columns

The configuration specified in the Figure 4 yields the final dashboard page presented in the Figure 5.

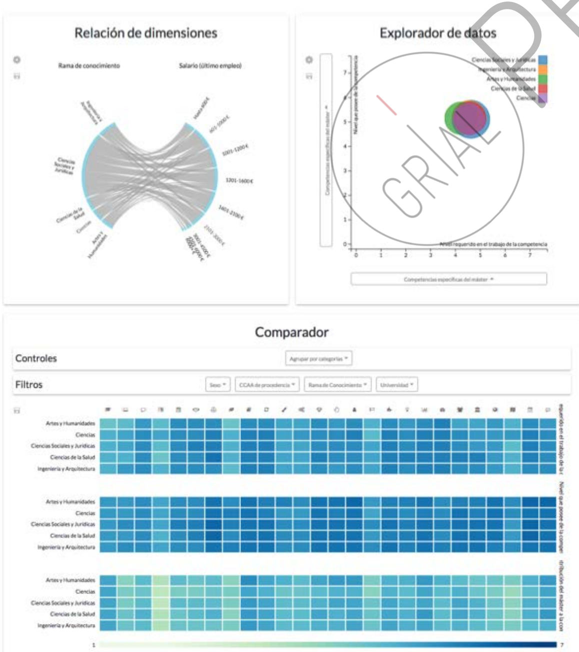


Figure 5. Example of a generated dashboard page given a specific configuration (contents in Spanish)

Thanks to the DSL it is possible to arrange the elements that will conform the dashboard page in terms of rows and columns, allowing developers to create and test different pages' layouts by only specifying them in the configuration files.

3.3 Results regarding data sources personalization

As it has been aforementioned, data sources are vital for dashboards to fulfil their role. There could be users that prioritize specific variables or specific information over another, and it is important to take these data requirements into account, as having too much information on an information dashboard could deteriorate the user experience.

This approach considers data sources as a part of a dashboard's configuration, making also the information presented itself or the information available to show (through a series of interactive controls) a customizable element.

For example, a particular user could require a heat map to have an overview of a series of skills required in the career path of the students (Figure 6).

Figure 6. Example of a heat map with the goal of showing the most required skills in the career path of the students

However, another user could require a heat map to study which are the most popular methods to search a job.

Starting from the same component (a heat map visualization), developers only need to specify through the DSL which data sources will be necessary to consume information from (an example of this syntax showed in the Figure 7). In this case, it is important to specify the Observatory's GraphQL API endpoints and parameters necessary to retrieve the requested data. As it will be discussed, the variety and homogenization of data sources is a challenge for this approach.

```xml
<Heatmap component_id="HeatMap_1">
    <Title>Level of skills</Title>
    <Dimensions>
        <Dimension dimension_id="1">
            <DataResource>
                <DataSource>
                    <label>Required in job</label>
                    <code>C1EMP</code>
                    <metric_code>oneLevelGroupedAverage</metric_code>
                    <endpoint>stats2017</endpoint>
                </DataSource>
            </DataResource>
        </Dimension>
        <Dimension dimension_id="2">
            <DataResource>
                <DataSource>
```

Figure 7. Example of the data source specification in a heat map visualization through the implemented DSL

By solely changing the data resources it is possible to achieve two visualizations adapted to the requested information.

4 DISCUSSION

Applying the software product line approach to the automatic generation of dashboards has led to promising results regarding the management of customization within this domain.

Software interfaces require both the study of the domain in which they will be framed and the study of the target users that will end up using the products. Moreover, interfaces that give support to decision-makers present additional difficulties given the particular requirements and factors that could affect the experience and, consequently, the visual data analysis and decision-making processes.

The main focus must be on providing powerful tools that are not only aesthetic and functional but also helpful for the users. However, there could be several user profiles, and the necessities of one user could differ totally from the necessities of another. Developing a specific dashboard for every user profile constitutes a solution, but it is inviable if the number of user profiles is significant within the particular problem to be addressed. In addition, the requirements of every user could evolve along time and that is also another issue that affects maintainability and consumes resources.

The software product line paradigm helps to manage different and dynamic requirements by providing a theoretical framework for implementing modular, configurable and reusable software components (named core assets of the product line) that can be composed to create final and complete software products. The results derived from this work have proved that this approach can decrease the efforts made during the development processes and improve maintainability and the evolution of the products. The most time-consuming tasks are carried out during the domain engineering phase, when the base core assets have to be designed and implemented to be reusable and configurable. Once this phase is done, the creation of particular products is straightforward.

The main challenges of applying the SPL to the dashboards' domain involve different matters. First, usability has to be a priority. As it has been mentioned before, the finally generated dashboards need to be helpful, and that involves having good levels of usability in order to provide valuable user experiences. However, the automatic generation of user interfaces is still a tough process that require semi-automatic or even manual design processes [25, 26]. Further research will involve usability tests to study the perceived usability levels of the automatically generated user interfaces.

On the other hand, the particular domain of the dashboards makes the data sources a vital element for the product line. There could be different data sources, with different data formats or even different ways or protocols to retrieve the information. It is important to take into account the heterogeneity of the sources involved to decouple the logic of the software components from the information that they will finally hold, in order to avoid critical changes on the components if the data sources are modified [27] in some sort of manner.

Developing a framework to efficiently generate flexible and customizable information dashboards could give a strong foundation to create powerful tools with the main goal of helping decision-makers to take well-informed decisions in order to obtain benefits from them.

5 CONCLUSIONS

In summary, the software product line paradigm has been applied to the Spanish Observatory for University Employment and Employability's system in order to provide an automatic method for generating customized dashboards to analyse the organization's data regarding university employment and employability, given a set of particular requirements.

Having a method for managing differing necessities benefits both developers and target users, increasing maintainability and efficiency in development processes and allowing fine-grained customization in the final products, respectively.

In this particular case, the creation of visualization tools for exploring data about university employment and employability could support policy-makers and institutions to identify factors that affect the students' capacity to obtain a job in order to improve the linkage between higher education and employment.

ACKNOWLEDGMENTS

The research leading to these results has received funding from "la Caixa" Foundation This work has been partially funded by the Spanish Government Ministry of Economy and Competitiveness throughout the DEFINES project (Ref. TIN2016-80172-R).

REFERENCES

[1] S Christian Albright, Wayne Winston, and Christopher Zappe. 2010. *Data analysis and decision making*. Cengage Learning.
[2] Dj Patil and Hilary Mason. 2015. *Data Driven.* " O'Reilly Media, Inc.".
[3] Ramesh Sharda, Dursun Delen, and Efraim Turban. 2013. *Business intelligence: a managerial perspective on analytics*. Prentice Hall Press.
[4] Stephen Few. 2006. Information dashboard design.
[5] Ronald W Mcquaid and Colin Lindsay. 2005. The concept of employability. *Urban studies* 42, 2, 197-219.
[6] Á. Fidalgo-Blanco, M. L. Sein-Echaluce, and F. J. García-Peñalvo. 2014. Knowledge spirals in higher education teaching innovation. *International Journal of Knowledge Management* 10, 4, 16-37. DOI:http://dx.doi.org/10.4018/ijkm.2014100102.
[7] Á. Fidalgo-Blanco, M. L. Sein-Echaluce, and F. J. García-Peñalvo. 2015. Epistemological and ontological spirals: From individual experience in educational innovation to the organisational knowledge in the university sector. *Program: Electronic library and information systems* 49, 3, 266-288. DOI:http://dx.doi.org/10.1108/PROG-06-2014-0033.
[8] J. P. Campbell, P. B. Deblois, and D. G. Oblinger. 2007. Academic Analytics. A new tool for a new era. *Educause Review* 42, 4, 40-42,44,46,48,50,52,54,56-57.
[9] Paul Baepler and Cynthia James Murdoch. 2010. Academic analytics and data mining in higher education. *International Journal for the Scholarship of Teaching and Learning* 4, 2, 17.
[10] Oficina De Cooperación Universitaria. 2013. Libro Blanco Inteligencia Institucional en Universidades OCU (Oficina de Cooperación Universitaria), Madrid, España.
[11] Francisco J García Peñalvo. 2015. Inteligencia Institucional para la Mejora de los Procesos de Enseñanza-Aprendizaje.
[12] Paul Clements and Linda Northrop. 2002. *Software product lines*. Addison-Wesley.
[13] Hassan Gomaa. 2004. *Designing Software Product Lines with UML: From Use Cases to Pattern-Based Software Architectures*. Addison Wesley Longman Publishing Co., Inc.

[14] F. Michavila, Jorge M. Martínez, Martín Martín-González, F. J. García-Peñalvo, and Juan Cruz-Benito. 2016. *Barómetro de empleabilidad y empleo de los universitarios en España, 2015 (Primer informe de resultados).* Observatorio de Empleabilidad y Empleo Universitarios, Madrid.

[15] F. Michavila, J. M. Martínez, M. Martín-González, F. J. García-Peñalvo, J. Cruz-Benito, and A. Vázquez-Ingelmo. 2018. *Barómetro de empleabilidad y empleo universitarios. Edición Máster 2017.* Observatorio de Empleabilidad y Empleo Universitarios, Madrid, España.

[16] Kyo C Kang, Sholom G Cohen, James A Hess, William E Novak, and A Spencer Peterson. 1990. *Feature-oriented domain analysis (FODA) feasibility study.* Carnegie-Mellon Univ Pittsburgh Pa Software Engineering Inst.

[17] Critina Gacek and Michalis Anastasopoules. 2001. Implementing product line variabilities. In *ACM SIGSOFT Software Engineering Notes* ACM, 109-117.

[18] Soheila Bashardoust Tajali, Jean-Pierre Corriveau, and Wei Shi. 2013. A Template-Based Approach to Modeling Variability. In *Proceedings of the International Conference on Software Engineering Research and Practice (SERP)* The Steering Committee of The World Congress in Computer Science, Computer Engineering and Applied Computing (WorldComp), 1.

[19] Timo Greifenberg, Klaus Müller, Alexander Roth, Bernhard Rumpe, Christoph Schulze, and Andreas Wortmann. 2016. Modeling variability in template-based code generators for product line engineering. *arXiv preprint arXiv:1606.02903.*

[20] Tim Bray, Jean Paoli, C Michael Sperberg-Mcqueen, Eve Maler, and François Yergeau. 1997. Extensible markup language (XML). *World Wide Web Journal* 2, 4, 27-66.

[21] Armin Ronacher. 2008. Jinja2 Documentation Welcome to Jinja2—Jinja2 Documentation (2.8-dev).

[22] Facebook. 2016. GraphQL.

[23] Andrea Vázquez-Ingelmo, Juan Cruz-Benito, and Francisco J García-Peñalvo. 2017. Improving the OEEU's data-driven technological ecosystem's interoperability with GraphQL. In *Proceedings of the Proceedings of the 5th International Conference on Technological Ecosystems for Enhancing Multiculturality* (Cádiz, Spain, October 18-20, 2017 2017). ACM, 3145437, 1-8. DOI:http://dx.doi.org/10.1145/3144826.3145437.

[24] Dhaivat Pandya. 2016. GraphQL Concepts Visualized, Web log post.

[25] Andreas Pleuss, Stefan Wollny, and Goetz Botterweck. 2013. Model-driven development and evolution of customized user interfaces. In *Proceedings of the 5th ACM SIGCHI symposium on Engineering interactive computing systems* ACM, 13-22.

[26] Andreas Pleuss, Benedikt Hauptmann, Markus Keunecke, and Goetz Botterweck. 2012. A case study on variability in user interfaces. In *Proceedings of the 16th International Software Product Line Conference-Volume 1* ACM, 6-10.

[27] Shaohua Wang, Iman Keivanloo, and Ying Zou. 2014. How do developers react to restful api evolution? In *International Conference on Service-Oriented Computing* Springer, 245-259.

7.6 Appendix F. Addressing fine-grained variability in user-centered software product lines: a case study on dashboards

Vázquez-Ingelmo, A., García-Peñalvo, F. J., & Therón, R. (2019). Addressing Fine-Grained Variability in User-Centered Software Product Lines: A Case Study on Dashboards. In Á. Rocha, H. Adeli, L. P. Reis, & S. Costanzo (Eds.), Knowledge in Information Systems and Technologies (Vol. 1, pp. 855-864). Springer Nature. https://doi.org/10.1007/978-3-030-16181-1_80

Addressing Fine-Grained Variability in User-Centered Software Product Lines: A Case Study on Dashboards

Andrea Vázquez-Ingelmo[1(✉)], Francisco J. García-Peñalvo[1], and Roberto Therón[1,2]

[1] GRIAL Research Group, Computer Sciences Department, Research Institute for Educational Sciences, University of Salamanca, Salamanca, Spain
{andreavazquez, Fgarcia, theron}@usal.es
[2] VisUSAL Research Group, University of Salamanca, Salamanca, Spain

Abstract. Software product lines provide a theoretical framework to generate and customize products by studying the target domain and by capturing the commonalities among the potential products of the family. This domain knowledge is subsequently used to implement a series of configurable core assets that will be systematically reused to obtain products with different features to match particular user requirements. Some kind of interactive systems, like dashboards, require special attention as their features are very fine-grained. Having the capacity of configuring a dashboard product to match particular user requirements can improve the utility of these products by providing the support to users to reach useful insights, in addition to a decrease in the development time and an increase in maintainability. Several techniques for implementing features and variability points in the context of SPLs are available, and it is important to choose the right one to exploit the SPL paradigm benefits to the maximum. This work addresses the materialization of fine-grained variability in SPL through code templates and macros, framed in the particular domain of dashboards.

Keywords: Software product lines · SPL · Granularity · User interfaces · Dashboards · Customization

1 Introduction

Software product lines (SPLs) address the systematic development of software assets for building families of products that share a specific domain [1, 2]. By reutilizing, configuring and composing these software assets, the time-to-market of new derived products decreases, in addition to an increase in requirements traceability, customization levels, flexibility, maintainability and of course, productivity.

However, implementing and introducing an SPL is not a straightforward job. The domain in which the SPL will be framed must be thoroughly studied to extract significant features and capture the commonalities among the potential products that could be developed through this paradigm. Planning the development of highly configurable software components allows the delay of design decisions, enhancing flexibility regarding the materialization of dynamic or even new requirements. These delayed design decisions are the so-called variability points [3].

The study of the target domain is the first step regarding an SPL design process, but the implementation of the identified variability points within the core assets of the product family remains a crucial and a critical challenge for this paradigm to succeed.

There are several techniques to materialize variability points, and the desired granularity of the SPL features is a relevant factor to choose the right method referring to the ability to modify the products behavior or their underlying functionality. In addition to the desired granularity level, the target domain of the SPL is also a key factor regarding the choice of the implementation technique.

For instance, user-centered tools require high levels of customization, both at functional and at visual design level. Developing these type of tools need further efforts on the requirements elicitation processes, in order to fully understand the final users' necessities and to provide them with helpful interfaces. Customizing user interfaces within the SPL paradigm context, however, is still a complex task, yet requiring semi-automatic or completely manual processes [4]. A large number of possible user profiles (and their associated particular requirements) could make the automatic derivation of interactive systems chaotic regarding its possible features, hampering the evolution and maintainability of the product line. The main issue regarding these interaction-intensive systems is the fine-grained nature of their features: a slight modification on interaction patterns, interface layout, color palette, etc. could be crucial regarding the final perceived usability of a generated product.

A particular case of these interactive systems is dashboards. These tools aim at helping users to reach useful insights about datasets, facilitating the discovery of unusual patterns or significant data points. The potential of dashboards resides in their ability to present information at-a-glance, supporting complex procedures like decision-making processes, communication, and learning, etc. [5]. A lot of profiles could be involved in these procedures though, being difficult to provide a common and general dashboard useful for each of them. That is why the SPL paradigm can ease the development of customized dashboards by reutilizing its different components (i.e., visualizations, controls, filters, interaction patterns, etc.), instead of implementing a single dashboard for each data domain or user involved. However, dashboards need fine-grained variability to provide powerful customizations and to support particular configurations for different user profiles, helping them to reach their own goals regarding data exploration and data explanation.

The remainder of this paper is structured as follows. Section 2 is an overview of a set of available methods for implementing variability within SPLs. Section 3 analyzes the particular case of the dashboards domain regarding the granularity of its features, presenting the case study in which an experimental framework for generating dashboards has been developed in Sect. 4. Finally, Sect. 5 discusses the achieved results regarding granularity, and Sect. 6 presents the conclusions of this work.

2 Variability Mechanisms

There exist different techniques to implement variability points in SPLs. It is important to choose wisely given the requirements of the product line itself (i.e., the complexity of the software to develop, its number of features, their granularity requirements, etc.). Generally, at the code level, the variability points that correspond to a specific feature will be spread across different source files [6]. That is why separating concerns at the implementation level is essential to avoid the variability points to be scattered, as this feature dispersion would decrease code understandability and maintainability. Implementing each feature in individual code modules can help with this separation of concerns [6], but it is difficult to achieve fine-grained variability through this approach. A balance between code understandability and granularity should be devised to choose both a maintainable and highly customizable SPL.

This section will briefly describe different methods that are potentially suitable to the dashboards' domain given their particular features, although there are more approaches to implement variability in SPLs that can be consulted in [6].

2.1 Conditional Compilation

Conditional compilation uses preprocessor directives to inject or remove code fragments from the final product source code. This method allows the achievement of any level of feature granularity due to the possibility of inserting these directives at any point of the code, even at expression or function signature level [7]. Also, although pretended to the C language, preprocessor directives can be used for any language and arbitrary transformations [8]. The main drawback of this approach is the decrease of code readability and understandability as interweaving and nesting these preprocessor directives makes the code maintainability a tedious task [9].

2.2 Frames

Frame technology is based on entities (frames) that are assembled to compose final source code files. Frames use preprocessor-like directives to insert or replace code and to set parameters [6]. An example of a variability implementation method based on frame technology is the XML-based Variant Configuration Language (XVCL) [10]. Through this approach, only the necessary code is introduced in concrete components by specifying frames that contain the code and directives associated with different features and variants. XVCL is independent of the programming language and can handle variability at any granularity level [11].

2.3 Template Engines

Template engines allow the parameterization and inclusion/exclusion of code fragments through different directives. If the template engine allows the definition of macros, features can be refactored into different code fragments encapsulated through these elements, improving the code organization and enabling variability at any level of granularity. Templating engines can also be language-independent, providing a powerful tool for generating any type of source file [12] by using programming directives such as loops and conditions.

2.4 Aspect-Oriented Programming

Aspect-Oriented Programming (AOP) allows the implementation of crosscutting concerns through the definition of aspects, centralizing features that need to be present in different source files through unique entities (aspects) thus improving code understandability and maintainability by avoiding scattered features and “tangled” code [13].

AOP is a popular method to materialize variability points in SPLs due to the possibility of modifying the system behavior at certain points, namely join points [14–16]. However, AOP could lack fine-grained variability (i.e., variability at sentence, expression or signature level) and particular frameworks or language extensions are necessary to implement aspects in certain programming languages.

3 The Dashboards’ Domain

Regarding the present work target domain (i.e., dashboards), the chosen implementation technique was to use a template engine. The decision was made due to the fine granularity that can be achieved through this method, which is necessary to materialize even the slightest variability on the visualization components. Another factor for choosing this technique lies in the straightforward way of implementing variability regarding the products’ features and its language-independent nature.

Framing technology could also be a potential solution within the dashboards’ domain, but the decision of designing a DSL to wrap the features at a higher level made the use of code templates a more suitable solution, providing complete freedom to define the syntax of the DSL (specification x-frames are based on a fixed syntax [11], which could result in lack of flexibility for this work’s approach) as the directives within the templates can be fully parameterized.

The selected template engine was Jinja2 (http://jinja.pocoo.org/docs/2.10/) given its rich API and powerful features such as the possibility of defining macros, importing them, defining custom filters and tags in addition to its available basic directives (loops, conditions, etc.).

4 Results of the Case Study

As it has been aforementioned, a DSL has been designed along with the SPL to abstract and ease the application engineering process. This DSL binds the feature model with the implementation method at code-level [17], enabling the specification of features through XML technology. Designing a DSL not only eases the configuration of variants but also improves the traceability of features through the different SPL paradigm phases (and opens up the possibility of combining the SPL paradigm with model-driven development [16]).

For this case study, it is necessary to provide a configurable SPL that enable automatic generation of dashboards with different features. These features involve a variety of potential requirements: from the modification of the dashboard layout (i.e., including or removing whole visualization components) to the modification of a particular interaction pattern to manage to zoom on visualizations, for example. To achieve the desired levels of granularity and to support the DSL for automating the application engineering phase, a template engine (Jinja2) was selected, as indicated in Sect. 3.

Templating resembles conditional compilation, as their underlying behavior based in programming directives is very similar. The main benefit of templates is that they support these directives and macros in a more sophisticated manner.

As presented in Sect. 2, the main drawback of conditional compilation is the scatter of concerns and features, decreasing code maintainability and readability. One of the benefits of using a powerful template engine like Jinja2 is the possibility of clustering the necessary code fragments that compose a certain feature in sets of macros. This approach improves maintainability, as the code fragments in charge of the features will be contained and organized in associated files.

The practical approach followed to apply this implementation method is exemplified in the remainder of this section.

Figure 1 shows a high-level view of the feature model for the dashboard product line developed for the Spanish Observatory for University Employability and Employment (OEEU, https://oeeu.org) [18, 19] to allow users to explore and reach insights about the data collected by this organization [20–25]. The generated dashboard can have different pages, each one composed of different visualizations and data filters. At this high-level view, features are coarse-grained; whole components can be included or removed from the final generated dashboard.

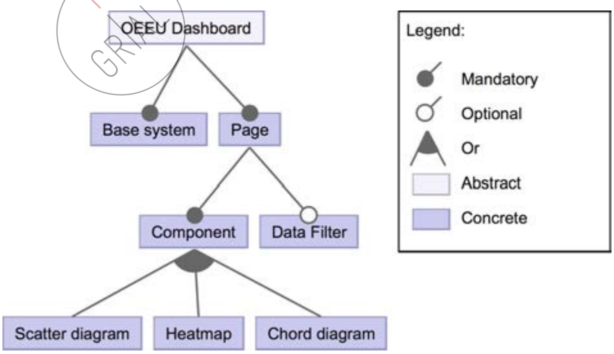


Fig. 1. High-level view of the dashboard SPL's feature diagram.

Low-level features (i.e., leaf nodes of the feature diagram) require fine-grained granularity within the dashboard domain, as these features concern minor visual, functional or interaction characteristics. Figure 2 shows low-level features for a scatter diagram component about the possible functionalities related to its data and behavior.

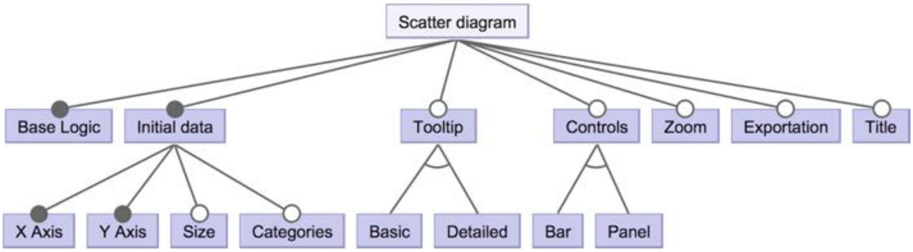


Fig. 2. A snippet of the feature diagram showing lower-level features regarding a scatter diagram component. Some of these features (e.g., the "controls" feature) have their own subsequent features to provide higher customization levels regarding the visualization's functional and information requirements.

To materialize these features at code-level, each feature is arranged in its own file and each file is composed with a set of macros Fig. 3. This set of macros contains the required code fragments associated with an SPL feature.

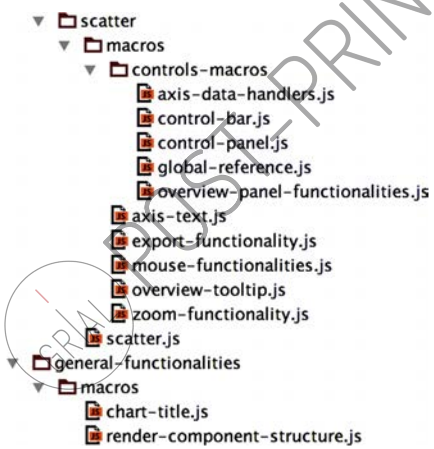


Fig. 3. Example of the code templates organization.

The macros calls are executed within the base logic of the component (in this case the "scatter.js" file contains the basic logic for the scatter diagram, which is mandatory and common for all possible product derivations, as specified in the feature diagram).

The macros themselves are affected by the conditional directives in charge of adapting the code giving particular configurations. This means that the base code will only contain the macro calls, delegating the condition check to the macros and making the code cleaner. By using this approach at the implementation level, concerns are not continuously scattered through the code as it could happen with pure preprocessor directives Fig. 4.

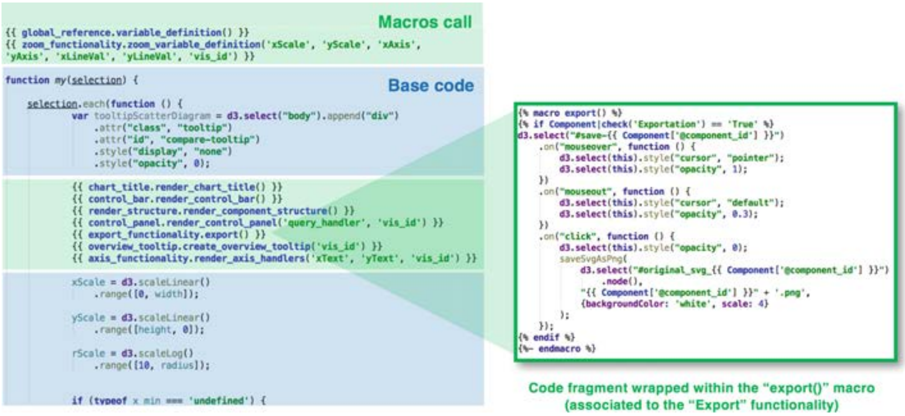


Fig. 4. A snippet of the scatter diagram's JavaScript code. The base code (highlighted in blue) contains macro calls (highlighted in green). If the condition wrapped within the macro is matched, the associated code is injected (i.e., the associated feature will be supported).

Through the DSL and the code templates, a custom code generator can build personalized dashboards that meet the specified requirements automatically Fig. 5.

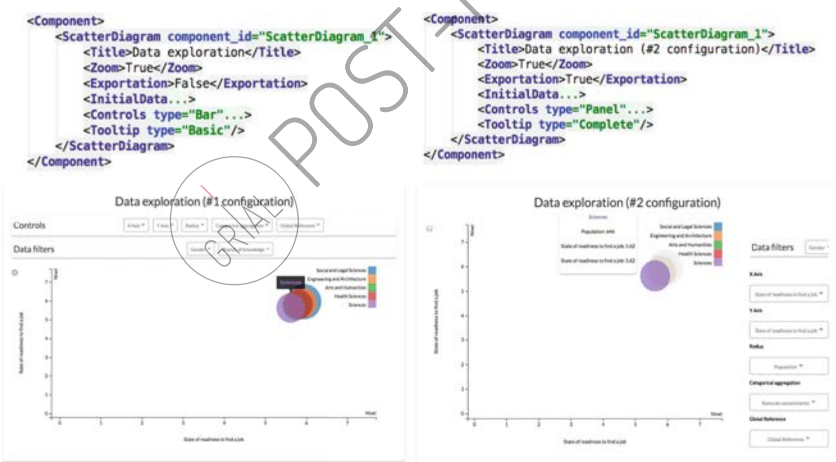


Fig. 5. Two different scatter diagram configurations achieved through the DSL (on top). As it can be seen, the tooltip type, for example, provides different behaviors when interacting with the visualization elements. Also, the layout of the whole visualization can be modified

5 Discussion

SPLs have proved to be a powerful paradigm for managing particular sets of requirements in an efficient and maintainable way. However, these requirements could need different granularity levels, as some important features could be coarse-grained while others could be fine-grained. Choosing the right implementation technique is a complex task because several factors must be taken into account: the levels of granularity, the understandability, and maintainability of the code, the viability of the technique, etc. This work addresses fine-grained granularity in a SPL of dashboards. Dashboards are key tools for reaching of insights regarding particular datasets and to support decision-making processes. Having the power of customizing their features at fine-grained level could be highly valuable, as dashboards usually ask to be user-tailored to provide useful support for particular and individual goals.

In the presented case study, a DSL has been designed for abstracting the configuration process. The use of this DSL to feed a code generator has been one of the determining factors to choose a template engine as the implementation method of the SPL's variability points. Although this approach still lacks powerful maintainability levels, it maintains a proper requirements' traceability by arranging features in a variety of macro definitions. Using XVCL [10] could have been another solution to manage these fine-grained features, but the decision of wrapping the SPL specification through a DSL asked for a more flexible and customizable method such as a template engine. What is more, a combination of the AOP paradigm with the templating method could be highly beneficial providing both customizations regarding directives and a better technique to manage crosscutting concerns (an issue that a template engine could not solve straightforwardly). Also, the approach asks for a method to address data heterogeneity in order to visualize data from any kind of source. However, although presenting these caveats, the results are promising and prove that a robust template engine could be a beneficial method to materialize fine-grained variability within the SPL paradigm context.

Regarding the application on the dashboard domain, having a dashboard SPL could address several problems related to individual personalization, meeting particular requirements. This approach could provide tailored dashboards efficiently after an in-depth elicitation of requirements without consuming many resources, avoiding overwhelming configuration processes delegated to end-users themselves [26].

6 Conclusions

Dashboards are sophisticated tools that require fine-grained features to offer valuable user experiences to their target users. A template-based approach to implement variability points at code level has been applied to an SPL of dashboards.

Creating an SPL of dashboards is not a straightforward task, as different variability dimensions are involved (variability regarding visual design, functionality, layout, data sources, etc.). Using a template engine to implement the core assets of the SPL can address the mentioned fine-grained variability and increase the traceability of features.

This SPL paradigm application to the dashboards' domain opens up different research paths, such as experimenting with different fine-grained configurations to find the best configuration for a particular user profile (A/B testing [23, 27]) or applying machine learning or knowledge bases [28] to provide potentially suitable configurations automatically given certain contexts or user characteristics. Also, developing an automatic link between the feature diagram and the DSL, as well between the DSL and the code templates' directives could further improve maintainability and traceability.

Acknowledgements. This research work has been supported by the Spanish *Ministry of Education and Vocational Training* under an FPU fellowship (FPU17/03276). This work has been partially funded by the Spanish Government Ministry of Economy and Competitiveness throughout the DEFINES project (Ref. TIN2016-80172-R) and the PROVIDEDH project, funded within the CHIST-ERA Programme under the national grant agreement: PCIN-2017-064 (MINECO, Spain).

References

1. Clements, P., Northrop, L.: Software Product Lines. Addison-Wesley, Boston (2002)
2. Pohl, K., Böckle, G., Linden, van Der Linden, F.J.: Software Product Line Engineering: Foundations, Principles and Techniques. Springer, New York (2005)
3. Van Gurp, J., Bosch, J., Svahnberg, M.: On the notion of variability in software product lines. In: 2001. Proceedings. Working IEEE/IFIP Conference on Software Architecture, pp. 45–54. IEEE (2001)
4. Pleuss, A., Hauptmann, B., Keunecke, M., Botterweck, G.: A case study on variability in user interfaces. In: Proceedings of the 16th International Software Product Line Conference, vol. 1, pp. 6–10. ACM (2012)
5. Sarikaya, A., Correll, M., Bartram, L., Tory, M., Fisher, D.: What do we talk about when we talk about dashboards? IEEE Trans. Visual. Comput. Graph (2018)
6. Gacek, C., Anastasopoules, M.: Implementing product line variabilities. In: ACM SIGSOFT Software Engineering Notes, pp. 109–117. ACM (2001)
7. Liebig, J., Apel, S., Lengauer, C., Kästner, C., Schulze, M.: An analysis of the variability in forty preprocessor-based software product lines. In: Proceedings of the 32nd ACM/IEEE International Conference on Software Engineering, vol. 1, pp. 105–114. ACM (2010)
8. Favre, J.-M.: Preprocessors from an abstract point of view. In: Proceedings of the Third Working Conference on Reverse Engineering 1996, pp. 287–296. IEEE (1996)
9. Spencer, H., Collyer, G.: #ifdef considered harmful, or portability experience with C News (1992)
10. Jarzabek, S., Bassett, P., Zhang, H., Zhang, W.: XVCL: XML-based variant configuration language. In: Proceedings of the 25th International Conference on Software Engineering, pp. 810–811. IEEE Computer Society (2003)
11. Zhang, H., Jarzabek, S., Swe, S.M.: XVCL approach to separating concerns in product family assets. In: International Symposium on Generative and Component-Based Software Engineering, pp. 36–47. Springer, Heidelberg (2001)
12. Cisco Blogs. https://blogs.cisco.com/developer/network-configuration-template
13. Kiczales, G., Lamping, J., Mendhekar, A., Maeda, C., Lopes, C., Loingtier, J.-M., Irwin, J.: Aspect-oriented programming. In: European Conference on Object-Oriented Programming, pp. 220–242. Springer, Heidelberg (1997)

14. Waku, G.M., Rubira, C.M., Tizzei, L.P.: A case study using AOP and components to build software product lines in android platform. In: 41st Euromicro Conference on Software Engineering and Advanced Applications (SEAA), pp. 418–421. IEEE (2015)
15. Heo, S.-h., Choi, E.M.: Representation of variability in software product line using aspect-oriented programming. In: Fourth International Conference on Software Engineering Research, Management and Applications, 2006, pp. 66–73. IEEE (2006)
16. Voelter, M., Groher, I.: Product line implementation using aspect-oriented and model-driven software development. In: 11th International Software Product Line Conference, SPLC 2007, pp. 233–242. IEEE (2007)
17. Voelter, M., Visser, E.: Product line engineering using domain-specific languages. In: 15th International Software Product Line Conference (SPLC), pp. 70–79. IEEE (2011)
18. Michavila, F., Martínez, J.M., Martín-González, M., García-Peñalvo, F.J., Cruz-Benito, J., Vázquez-Ingelmo, A.: Barómetro de empleabilidad y empleo universitarios. Edición Máster 2017. Observatorio de Empleabilidad y Empleo Universitarios, Madrid, España (2018)
19. Michavila, F., Martínez, J.M., Martín-González, M., García-Peñalvo, F.J., Cruz-Benito, J.: Barómetro de empleabilidad y empleo de los universitarios en España, 2015 (Primer informe de resultados). Observatorio de Empleabilidad y Empleo Universitarios, Madrid (2016)
20. Vázquez-Ingelmo, A., Cruz-Benito, J., García-Peñalvo, F.J., Martín-González, M.: Scaffolding the OEEU's data-driven ecosystem to analyze the employability of spanish graduates. In: García-Peñalvo, F.J. (ed.) Global Implications of Emerging Technology Trends, pp. 236–255. IGI Global, Hershey (2018)
21. García-Peñalvo, F.J., Cruz-Benito, J., Martín-González, M., Vázquez-Ingelmo, A., Sánchez-Prieto, J.C., Therón, R.: Proposing a machine learning approach to analyze and predict employment and its factors. Int. J. Interact. Multimedia Artif. Intell. **5**(2), 39–45 (2018)
22. Vázquez-Ingelmo, A., García-Peñalvo, F.J., Therón, R.: Generation of customized dashboards through software product line paradigms to analyse university employment and employability data. Learning Analytics Summer Institute Spain 2018 – LASI-SPAIN 2018, León, Spain (2018)
23. Cruz-Benito, J., Vázquez-Ingelmo, A., Sánchez-Prieto, J.C., Therón, R., García-Peñalvo, F. J., Martín-González, M.: Enabling adaptability in web forms based on user characteristics detection through A/B testing and machine learning. IEEE Access **6**, 2251–2265 (2018)
24. Vázquez-Ingelmo, A., García-Peñalvo, F.J., Therón, R.: Domain engineering for generating dashboards to analyze employment and employability in the academic context. In: 6th International Conference on Technological Ecosystems for Enhancing Multiculturality, Salamanca, Spain (2018)
25. Vázquez-Ingelmo, A., García-Peñalvo, F.J., Therón, R.: Application of domain engineering to generate customized information dashboards. In: International Conference on Learning and Collaboration Technologies, pp. 518–529. Springer, Switzerland (2018)
26. Elias, M., Bezerianos, A.: Exploration views: understanding dashboard creation and customization for visualization novices. In: IFIP Conference on Human-Computer Interaction, pp. 274–291. Springer, Heidelberg (2011)
27. Kakas, A.C.: A/B Testing (2017)
28. Moritz, D., Wang, C., Nelson, G.L., Lin, H., Smith, A.M., Howe, B., Heer, J.: Formalizing visualization design knowledge as constraints: actionable and extensible models in Draco. IEEE Trans. Visual. Comput. Graph. **25**, 438–448 (2019)

7.7 Appendix G. Tailored information dashboards: A systematic mapping of the literature

Tailored information dashboards: A systematic mapping of the literature

Andrea Vázquez-Ingelmo
GRIAL Research Group, Department of Computer Science.
University of Salamanca
Salamanca, Spain
andreavazquez@usal.es

Francisco José García-Peñalvo
GRIAL Research Group, Department of Computer Science.
University of Salamanca
Salamanca, Spain
fgarcia@usal.es

Roberto Therón
GRIAL, VisUSAL Research Group, Department of Computer Science.
University of Salamanca
Salamanca, Spain
theron@usal.es

ABSTRACT

Information dashboards are extremely useful tools to exploit knowledge. Dashboards enable users to reach insights and to identify patterns within data at-a-glance. However, dashboards present a series of characteristics and configurations that could not be optimal for every user, thus requiring the modification or variation of its features to fulfill specific user requirements. This variation process is usually referred to as customization, personalization or adaptation, depending on how this variation process is achieved. Given the great number of users and the exponential growth of data sources, tailoring an information dashboard is not a trivial task, as several solutions and configurations could arise. To analyze and understand the current state-of-the-art regarding tailored information dashboards, a systematic mapping has been performed. This mapping focus on answering questions regarding how existing dashboard solutions in the literature manage the customization, personalization and/or adaptation of its elements to produce tailored displays.

CCS CONCEPTS

• Human-centered computing → Visualization systems and tools • Human-centered computing → Visualization theory, concepts, and paradigms

KEYWORDS

Dashboards, Information dashboards, Information visualization, Systematic mapping, Literature review, Customization, Personalization, Adaptation

Interacción 2019, June 25–28, 2019, Donostia, Gipuzkoa, Spain

ACM ISBN 978-1-4503-7176-6/19/06…$15.00
https://doi.org/10.1145/3335595.3335628

ACM Reference format:

Andrea Vázquez-Ingelmo, Francisco J. García-Peñalvo, and Roberto Theron. 2019. Tailored information dashboards: a systematic mapping of the literature. In *Proceedings of Interacción2019. ACM, New York, NY, USA, 8 pages.* https://doi.org/10.1145/3335595.3335628

1 INTRODUCTION

The exponential growth of data sources and the relevance that data has nowadays for a majority of essential activities and tasks has consolidated information dashboards as extremely useful tools to reach insights about large datasets and support decision-making. Information dashboards are composed of a series of graphical components and interaction methods that allow visual analysis of datasets to ease the recognition of interesting patterns or relationships among the presented variables.

However, information dashboards face several challenges. Their spread of use among different contexts and the increase of data sophistication turn their design process a complex task. What is more, dashboards are employed by many different users with different profiles, making difficult the suitability of a general dashboard solution, given the variety of potential user requirements. In recent studies, these challenges have been highlighted. Alper et al. conducted a systematic review of several dashboard solutions to characterize them and identify different types of dashboards [1]. The study proved that dashboard solutions are very diverse in terms of design, components, indicators, interaction patterns and, especially, goals. This nature causes the need for creating domain-specific solutions and even user specific solutions, consuming significant time and resources and being very difficult to adapt and reuse them in different contexts. To address these issues, there are user-friendly tools that enable users to create and customize their dashboard without requiring programming skills, like Tableau (https://www.tableau.com/) or Grafana (https://grafana.com/).

But, regarding dashboards, there is an additional issue; users can lack visualization literacy making the *customization* process of a dashboard a tedious and even arbitrary task that could lead to ineffective dashboards as a result [2].

That is why an adaptive dashboard solution could reduce the cost of creating new dashboards and improve the user experience by providing *personalized* views based on different factors (user knowledge, context, domain, etc.).

It is necessary, nevertheless, to distinguish between the terms "customized" and "personalized". Customization refers to a user-initiated process to tailor their interfaces, functionalities, contents, etc. to fulfill their requirements, while personalization refers to a system-initiated process that uses information to tailor the aforementioned elements without an explicit user intervention [3]. However, "personalized" and "customized" are often misunderstood as interchangeable terms, being necessary to emphasize their differences.

Given the large amount of content customization and/or personalization possibilities, and the potential misconception of those terms, this paper aims to investigate the existing literature regarding the customization, adaptation and/or personalization of information dashboards, focusing on mapping [4] the collected studies to understand the existing solutions and research lines of this area.

The remainder of this paper is organized as follows. Section 2 outlines the research method followed to perform the systematic mapping. Section 3 describes the data extraction process for analyzing the collected works. Section 4 presents the results of the systematic mapping, finishing with Section 5 where the results are discussed and Section 6, where the work's conclusions are shared.

2 RESEARCH METHOD

This study is based on the guidelines suggested by Kitchenham and Charters [5] for systematic literature studies and the guidelines suggested by Petersen [6] for mapping studies. The mapping process is organized in a series of phases; first, the planning phase, where the main goals and research questions to be answered are defined. Second, the conducting phase, where the search strategy is generated and the selection, assessment and data extraction of the studies are performed. The final stage is the reporting phase, where the results are disseminated.

2.1 Research questions

The research goal of this systematic mapping is to analyze proposed solutions regarding information dashboards' adaptation, personalization or any kind of variation regarding their contents (personalization, customization and adaptation processes will be referred to as "variability processes" to enclose them under the same term).

To do so, this systematic mapping aims to answer the following mapping questions:

- **MQ1**. How many studies were published over the years?
- **MQ2**. Who are the most active authors in the area?
- **MQ3**. What type of papers are published?
- **MQ4**. To which contexts have been the variability processes applied? (BI, learning analytics, etc.)
- **MQ5**. Which are the factors that condition the dashboards' variability process?
- **MQ6**. What is the target of the variability process? (visual components, KPIs, interaction, the dashboard as a whole, etc.)
- **MQ7**. At which development stage is the variability achieved?
- **MQ8**. Which methods have been used for enabling variability?
- **MQ9**. How many studies have tested their proposed solutions in real environments?

2.2 Inclusion and exclusion criteria

To discard irrelevant works (in terms of the scope of this paper) from the search results, a set of inclusion criteria (IC) and a set of exclusion criteria (EC) were defined:

- **IC1.** The paper described a dashboard solution (proposal, architecture, software design, model, tool, etc.) **AND**
- **IC2.** The solution was applied to information dashboards (omitting any other kind of "dashboard") **AND**
- **IC3.** The solution supported or addressed tailoring capabilities (customization, personalization, adaptation, variation) regarding information dashboards **AND**
- **IC4.** The tailoring capabilities of the dashboard are related to its design, components or KPIs **AND**
- **IC5.** The papers were written in English or Spanish **AND**
- **IC6.** The papers were published in peer-reviewed Journals, Books or Conferences **AND**
- **IC7.** The publication is the most recent or complete of the set of related publications regarding the same study

The following items refer to the exclusion criteria applied:

- **EC1.** The paper did not describe a dashboard solution (proposal, architecture, software design, model, tool, etc.) **OR**
- **EC2.** The solution was not applied to information dashboards **OR**
- **EC3.** The solution did not support or addressed tailoring capabilities (customization, personalization, adaptation, variation) regarding information dashboards **OR**
- **EC4.** The tailoring capabilities of the dashboard are not related to its design, components or KPIs **OR**
- **EC5.** The papers were not written in English or Spanish **OR**
- **EC6.** The papers were not published in peer-reviewed Journals, Books or Conferences **OR**
- **EC7.** The publication is not the most recent or complete of the set of related publications regarding the same study

2.3 Search strategy

The first step taken to extract relevant works for this paper was the selection of the employed electronic databases. In this case, four electronic databases were selected: Scopus, Web of Science (WoS), IEEE Xplore and SpringerLink. These databases were chosen according to a set of requirements:

- It is a reference database in the research scope.

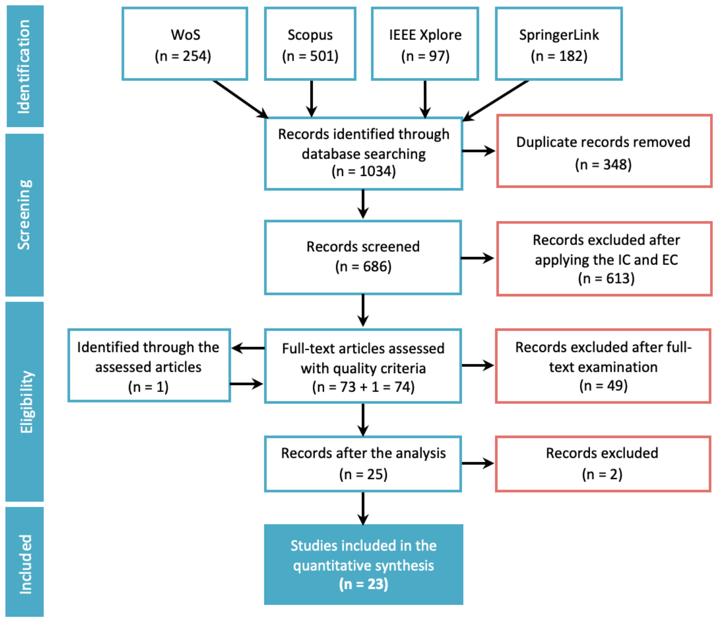


Figure 1: PRISMA flow. Adapted from [7].

- It is a relevant database in the research context of this mapping study.
- It allows using similar search strings to the rest of the selected databases as well as using Boolean operators.

Regarding the search terms, it was necessary to enclose any term related to customization, personalization, adaptation, context-awareness, etc. That is why, in addition to the terms above, concepts like flexible and configurable were included.

To include system-initiated processes that can address personalization, terms related to generation, template-based, composition or subject-driven processes were also incorporated. Finally, any work focused on dealing with heterogeneous, diverse, or dynamic stakeholders, users, requirements, tasks, etc. were also included as these circumstances could ask for customization or personalization.

The conjunction of these terms with the “dashboard” concept aims to collect any study addressing variability within the information dashboards’ domain. The term “meta-dashboard” was also incorporated to retrieve works seeking to define abstract dashboards that can be used to generate different types of concrete dashboards.

2.4 Search strings

The search strings for each chosen source were defined from the search terms connected by boolean AND / OR / NEAR operators. Moreover, the wildcard (*) was used in Scopus and Web of Science (WoS) to include both singular and plural of each term.

The NEAR operator enables the user to retrieve works where the terms joined by this operator are separated by a specified number of words at most. This operator is very useful for this research, as the terms “customizable”, “personalized”, “adaptive”, etc. should only refer to the dashboard term. However, it is necessary to explicitly select the number of words that can separate the target terms.

Specifically, the chosen number was 10 (i.e. the "dashboard" term and the rest of the terms will be within 10 number of words of each other). This number was selected after executing the same search with different proximity values (5, 7, 10 and 12). Examining the additional records found after incrementing this value, it was concluded that the ten value would retrieve relevant works without adding noise (i.e., irrelevant works).

The base structure of the search string (which was adapted to the specific syntax of the electronic databases afterward) was the following:

((meta-dashboard*) **OR**

((dashboard*) NEAR/10 (custom* OR personal* OR adapt* OR flexib* OR config* OR driven OR generat* OR compos* OR template* OR context-aware OR select*)) **OR**

((dashboard*) AND ((heterogeneous OR different OR diverse OR dynamic) NEAR/0 ("requirement*" OR "stakeholder*" OR "user*" OR "need*" OR "task*" OR “necess*”)))) **AND NOT**

(car OR vehicle OR automo*)

Some terms belonging to the automotive field were excluded to avoid the retrieval of car dashboards' studies, which are out of the scope of this research.

In the case of SpringerLink, the NEAR operator is not supported, so the AND operator was used. To limit the results, only the records that contained the term "dashboard" in their titles were retrieved.

3 DATA EXTRACTION

To describe the iterative data extraction process followed, the PRISMA statement [7] is used (Figure 1). To accomplish the first stage, the results obtained after applying the search strings were downloaded in CSV (comma-separated values) format, stored in a repository in GitHub [8], and organized in a spreadsheet in Google Sheets (http://bit.ly/2L8GRfY). Next, the title, the abstract and the keywords of each paper were analyzed, and the inclusion and exclusion criteria were applied. Finally, each candidate paper was fully read to decide if it fulfills a quality criterion (i.e., a set of characteristics to ensure that a paper fits in the context of this research). During the analysis, a series of quantitative questions were answered to perform subsequently the analysis.

After the full reading of the works, other relevant papers were identified through their references. Concretely, one paper was added and read in depth too. All the information regarding this stage was organized in a fourth sheet ("Third phase") of the spreadsheet (http://bit.ly/2L8GRfY).

To sum up, 1034 papers were collected once the search strings were applied, of which 254 from WoS, 501 from Scopus, 97 from IEEE Xplore and 182 from SpringerLink.

- After removing duplicates, there were 686 papers maintained.
- Once the criteria were applied to title, abstract and keywords, 73 papers moved into the next phase (10.79% of the unique papers retrieved).
- One paper was added after reading the selected ones, leaving 74 papers for the quality criteria application.
- After applying the quality criteria, 25 papers were selected and thoroughly analyzed. After the analysis, 2 papers were excluded as they did not comply with the IC7 criteria item.
- Finally, a total of 23 papers were analyzed (3.35% of the unique papers retrieved and 31.08% of the full-text assessed papers).

4 SYSTEMATIC MAPPING RESULTS

This section presents the mapping results of the collected records through the search above strategy. A Jupyter notebook (http://jupyter.org) was created to support the analysis process of the raw data [8]. The notebook is based on the work developed by Cruz-Benito http://bit.ly/2tS9JgF.

4.1 MQ1. How many studies were published over the years?

The number of selected papers per year were counted as can be seen in Figure 2.

The results cover from 2011 to 2018, with a work placed in 2007 [9]. A few records were published in 2011 [10; 11], 2012 [12], 2013 [13], 2014 [14-16] and 2016 [17; 18]. However, the majority of records are distributed between 2017 [19-24] and 2018 [25-31], six and seven papers respectively.

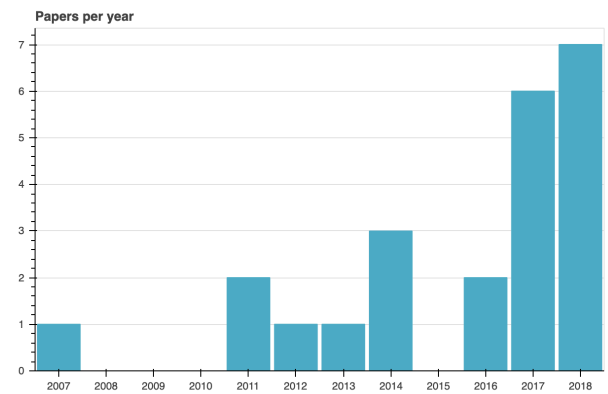


Figure 2: Distribution of papers per year.

4.2 MQ2. Who are the most active authors in the area?

Works published by each author were counted to answer this question. Only one author has more than one record. Kintz presents a model-driven solution for generating dashboards [12; 21], in one case it presents the semantic description language and in the other one it presents an extension to take into account user roles in the dashboards generation process. The rest of the authors appear only once in this mapping study. Table 1 shows all of the authors and their number of papers in the scope of this literature mapping. There are some authors that also had more than one paper related to tailored dashboards, however, they were omitted because of the exclusion criteria EC7, so the most recent and complete paper about their study made it to the final phase.

Table 1. Authors' addressing variability on dashboards.

Author	Total
Kintz M.	2
Arjun S.; Barros R.; Bederson B.B.; Belo O.; Bezerianos A.; Borges M. R. S.; Bose J.; Cardoso A.; Chowdhary P.; Collet P.; Correia H.; Danaisawat K.; Dantas V.; de Walle R .V.; Elias M.; Elmqvist N.; Filonik D.; Foth M.; Furtado V.; García-Peñalvo F. J.; George S.; Hruška T.; Huys C.; Hynek J.; Ines D.; Janssens O.; Jean-Marie G.; Ji M.; Karstens E.; Khunkornsiri T.; Kochanowski M.; Koetter F.; Kukolj S.; Kumar K.; Lavoue E.; Logre I.; Madeth M.; Magnoni L.; Majstorović B.; Mayer B.; McGuinness D. L.; Medland R.; Michel C.; Mihaila G.; Miotto G. L.; Mosser S.; Nascimento B. S.; Noonpakdee W.; Palpanas T.; Pastushenko O.; Petasis G.; Phothichai A.; Pinel F.; Pinheiro P.; Radovanović S.; Rittenbruch M.; Riveill M.; Rodrigues P.; Santos H.; Sebastien I.; Serge G.; Sloper J. E.; Soni S. K.; Sousa Pinto J.; Therón R.; Triantafillou A.; Van Hoecke S.; Vázquez-Ingelmo, A.; Verborgh R.; Vieira Teixeira C. J.; Vivacqua A. S.; Weinreich R.; Yalcin M. A.	1

4.3 MQ3. What type of papers has been published?

Each consulted electronic database provides the metadata to answer this mapping question. According to the inclusion and exclusion criteria, only papers involved in a peer review process (either in journals, conferences, books or workshops) are included. The complete list of types regarding the analyzed records can be consulted in Table 2.

Table 2. Papers grouped by type of publication

Type	Total	Papers
Article	4	[26] [29] [22] [9]
Conference paper	19	[13] [19] [17] [25] [20] [12] [21] [16] [27] [18] [11] [23] [28] [30] [14] [15] [10] [24] [31]

4.4 MQ4. To which contexts have been the variability processes applied?

Dashboards can be used in any domain; the only requirement is to have enough data to visualize. Regarding customizable and/or personalized dashboards, it can be seen that Business Intelligence (BI) is the most common application domain (Figure 3), followed by the Internet of Things (IoT), Learning Analytics (LA), services monitoring and social science domains.

Table 3. Papers grouped by target domain

Domain	Total	Papers
Business Intelligence	8	[9] [10] [12] [14] [21] [23] [25]
IoT	2	[16] [18]
Learning Analytics	2	[22] [24]
Services monitoring	2	[20] [26]
Disaster situations	1	[30]
Economics	1	[31]
Emergency management	1	[17]
Energy monitoring	1	[13]
Generic	1	[29]
Interface evaluation	1	[28]
Microservices monitoring	1	[19]
Physics	1	[11]
Sensor monitoring	1	[15]
Social sciences	1	[27]

4.5 MQ5. Which are the factors that condition the dashboards' variability process?

One of the first steps to perform a variability process is to determine the factors that will condition the dashboards' variation, i.e., the inputs of the customization and/or personalization stage. The majority of the included papers make use of the user preferences as input to modify the dashboard appearance and functionality (Table 4).

Table 4. Papers grouped by variability factors

Factor	Total	Papers
User preferences	15	[13] [19] [17] [25] [20] [26] [27] [11] [23] [28] [29] [15] [22] [10] [18]
Data structure	4	[23] [29] [24] [31]
Business process	3	[12] [9] [21]
User role	2	[21] [9]
Design guidelines	2	[25] [28]
Usage profiles	1	[14]
Data sources	2	[16] [18]
Goals	2	[12] [21]
User description	1	[24]
Analysis scenario	1	[24]
User abilities	1	[30]

4.6 MQ6. What is the target of the variability process?

Variability processes have a target that will change or be modified after the variation has been accomplished. In the case of dashboards, several elements could be the target of the variation: visualization types, layout, displayed data, visual design (i.e., color palettes, font sizes, etc.) and even interaction (pan, zoom, etc.) or functionalities (filters, exportation, etc.).

Table 5 lists the different variability targets identified in the included papers.

Table 5. Papers grouped by variability target

Target	Total	Papers
Displayed data	22	[13] [19] [17] [25] [20] [12] [26] [21] [16] [27] [18] [11] [23] [28] [9] [29] [14] [15] [22] [10] [24] [31]
Visualization types	21	[13] [19] [17] [25] [20] [12] [26] [16] [27] [18] [11] [23] [28] [9] [29] [14] [15] [22] [10] [24] [31]
Layout	20	[13] [19] [17] [25] [20] [12] [26] [16] [27] [18] [11] [23] [28] [9] [29] [14] [15] [22] [10] [24]
Functionalities	1	[27]
Visual design	2	[30] [26]
Interaction	2	[30] [12]

4.7 MQ7. At which development stage is the variability achieved?

The modification of dashboard features can be performed at different stages. In this case, four stages were identified: compile-time, run-time, pre-configuration time (i.e. a phase before the creation of the dashboard in which its configuration is defined by the end-user or any other stakeholder) and user-configuration time (i.e. at run-time, but the user is in charge of the configuration of its dashboard).

Pre-configuration and user-configuration seem to be the most preferred stages to customize or personalize the dashboards (Table 6).

Table 6. Papers grouped by variability methods

Stage	Total	Papers
Pre-configuration	9	[9] [12] [15] [21] [23] [25] [26] [27] [28]
User-configuration	8	[10] [11] [13] [17] [19] [20] [22] [31]
Run-time	6	[14] [16] [18] [24] [29] [30]
Compile-time	1	[16]

4.8 MQ8. Which methods have been used for enabling variability?

This mapping question aims at analyzing the methods, techniques, paradigms, etc. used for enabling customization and/or personalization within the dashboards' domain.

A set of methods have been identified through the included papers. The most repeated method consists of configuration wizards, to allow users to tailor their dashboards. Some solutions give extra support to these wizards with visual mapping to ease the selection of proper visualizations given the data structure to be visualized [11; 17; 29; 31]. Other common methods found involve configuration files, agents, software product lines (SPL) and model-driven development. The complete list of methods can be consulted in Table 7.

Table 7. Papers grouped by variability methods

Method	Total	Papers
Configuration wizard	8	[13] [19] [17] [11] [29] [22] [10] [31]
Visual mapping	4	[17] [11] [29] [31]
Configuration files	3	[20] [26] [28]
Model-driven	3	[12] [21] [9]
Agents	2	[16] [14]
SPL	2	[27] [15]
Pre-defined templates	2	[25] [9]
Semantic reasoner	1	[18]
Inclusive user modeling	1	[30]
Context-aware generator	1	[24]
Indicator ontology	1	[23]
Knowledge graphs	1	[23]

4.9 MQ9. How many studies have tested their proposed solutions in real environments?

The last mapping question is regarding the performed tests on the included dashboard solutions.

The majority (13) of the solutions have been tested in real-world scenarios, involving real data and real users, while 6 of the solutions have not been tested with real users or real data (Table 8). There are four solutions that have been partially tested in a real-world scenario, i.e., they have been tested with real data but not with real users, or vice versa.

Table 8. Papers grouped by testing maturity

Tested?	Total	Papers
Yes	13	[10] [11] [13] [15] [16] [17] [18] [21] [24] [25] [29] [30] [31]
No	6	[12] [14] [19] [20] [23] [26]
Partially	4	[9] [22] [27] [28]

5 DISCUSSION

In total, there are 23 papers that present dashboard solutions with tailoring features. However, as the systematic mapping results showed, these solutions are quite miscellaneous.

In [13], a customizable dashboard display for monitoring mobile energy is presented; users can build their dashboards by selecting pre-defined widgets and data streams from different sources (sensors, government agencies, social media and generic services). This kind of "manual approach" is also used in [10; 19; 22], in which the customizability capacity is based on the possibility of arrange the components of the dashboard through explicit interaction, and even the capacity of crafting custom indicators, as presented in [22]. The same "customizability principles" are present in other papers, with the difference of involving automatized approaches through configuration files [20; 26; 28], models [15; 27] or pre-defined templates [25].

Other solutions involve personalization; in [12], the methodology takes as input a model of the business process and goals to describe and generate a dashboard, so the authors use implicit data (goals) to build a concrete dashboard that would help to reach that input goals. The aforementioned work is extended in [21], where user-roles are taken into account to add more information to the dashboard personalization process. A similar solution is presented in [9], which also takes into account user-roles and business' KPIs to generate a dashboard that fits the business goals. Finally, in [30], the focus is on personalizing the display taking into account the user abilities through an initial questionnaire that ask users if they have eye diseases or any tremor in hands, making the dashboard accessible if necessary.

There are also solutions that can adapt themselves at run-time based on environmental changes. Belo et. al [14] present an adaptive dashboard that restructures itself given user profiles and behaviors extracted from the dashboards' analytical sessions. Another adaptive solution presented in [24] uses a dashboard generator fed with user, data and visualization models, thus generating information dashboards based on these models.

In [16], a device cloud platform dashboard is built based on the data model of the remote devices being monitored, but users can also customize it manually. On the other hand, Van Hoecke et al. [18] use a semantic reasoner to personalize indicators from available data sources.. Santos et al. [23] also proposes personalized dashboards based on knowledge graphs and indicator ontologies, but the solution allows the users to modify the dashboard recommendation to its own preferences.

Finally, there are four solutions that can assist and help the users to build their dashboards according to a series of factors. The papers identified in this category [11; 17; 29; 31] help users to identify the best visualization types for the data to be visualized while building and designing their dashboards.

Taking into account the factors that affect the variation of the dashboards' features and the development stage at which the tailoring process is performed, a high-level classification of the selected papers is presented in Table 9.

Customizable dashboards take as input explicit user requirements regarding their dashboards, while personalized dashboards use implicit user data (usage profiles, goals, business processes, etc.) at the moment of the dashboard creation to tailor its features. Adaptive solutions, on the other hand, use implicit user data to adapt the dashboards' components at run-time, taking into account that user requirements can evolve.

Finally, two more kinds of tailored solutions have been identified. Hybrid solutions are personalized solutions with customization support (i.e., the user can manually change the dashboard's personalized features), while customizable solutions with system support help users to configure their dashboards with (personalized) recommendations that can be optionally selected.

Table 9. Dashboard solutions classified by type of tailoring

Type	Total	Papers
Customizable	10	[13] [19] [25] [20] [26] [27] [28] [15] [22] [11]
Customizable w/ system support	4	[17] [10] [29] [31]
Personalized	4	[12] [21] [9] [30]
Hybrid	3	[16] [18] [23]
Adaptive	2	[14] [24]

Tailoring information dashboards has been identified as a relevant field and process, given the potential number of different requirements and user profiles that can employ these tools to support their decision-making processes.

Customizable solutions are the most common solutions, both in research and commercial areas. However, customizable solutions, although they do not require programming skills, still induce cognitive workload on users, because they need to determine their requirements and build their dashboards accordingly. Adaptive, hybrid and personalized solutions would potentially benefit users that don't have a clear set of requirements.

As shown in the first mapping question (MQ1), tailoring information dashboards is a current topic with very recent works addressing this issue. Regarding the different domains in which the dashboard solutions have been applied, Business Intelligence is the most common domain, given the relevance of dashboards for supporting business decision-making processes. However, dashboards are using in very diverse contexts, ranging from learning analytics to economics and energy monitoring. Using dashboard models to create generic solutions that can be adapted to any context would be highly useful, as mentioned in [27].

The majority of works take advantage of user-configuration and pre-configuration methods (through configuration wizards and configuration files) to tailor their solutions. These methods allow to easily build dashboards focusing on user preferences and decreasing the development time of specific solutions.

However, there is still room for automation, which is why other solutions use implicit user data, like usage profiles or business processes, to adapt the dashboards at run-time or personalize them before their delivery to the users.

Regarding the variability targets, only one work included the dashboards' functionalities [27], interaction possibilities [12; 30] and/or visual design [26]. Although the displayed data, visualization types, and layout of the dashboards are very relevant elements, these aforementioned features should not be ignored, as they also influence the user experience [1; 32].

The methods to achieve the variability of dashboard characteristics are very diverse. In general, configuration wizards support customizable solutions. Configuration files, model-driven development, and software product lines also enable variability by modeling the requirements in a structured format for a system to understand, and to generate concrete dashboards. Agents also provide support to manage evolving requirements and perform changes on the dashboards' configuration.

On the other hand, four solutions used visual mapping to assist the customization process. Visual mapping allows the selection of the best visualization type given a set of inputs (displayed data structure, user profiles, etc.), and can be very useful to recommend suitable configurations [33].

Finally, the majority of the solutions have been tested in real-world environments. On the other hand, 10 solutions have not completely validated their solutions. Users are the final beneficiaries of these tools, so they should be tested to verify their usefulness. However, having flexible solutions that don't require significant development time can ease the testing processes through A/B testing, for example [34].

This work aimed at identifying solutions to address tailoring on information dashboards. Performing a systematic review allows to identify valuable records in the literature in a replicable and traceable way. Information dashboards are increasingly being used in different domains, not only in business intelligence contexts, which has been the tendency in the past. But as stated in the introduction, dashboards need to be tailored to take into account individual requirements and needs, thus enhancing insight delivery. Knowing which solutions have been applied in the past within the context of tailoring dashboards helps to provide a basis for research opportunities.

Customizable dashboards are easier to implement, but the final user is still the responsible of building the dashboard. These users may not realize which configuration is the best for them, because they may not explicitly know what they want in their screens. Having customizable solutions with system support can mitigate this drawback, but in the end, the responsibility is still on the user. On the other hand, personalized solutions could be rigid if the requirements evolve over time. That is why adaptive or hybrid solutions seem to be better approaches, providing the user with a personalized solution with room for customization and supporting the requirements evolution.

There are different research opportunities found through the performance of this systematic mapping. The majority of the papers retrieved are between the 2016-2018 interval, meaning that tailored dashboards are a current concern. Only two solutions mention artificial intelligence approaches [14; 18]. Artificial intelligence paradigms could be extremely useful in this context, as an ideal solution might be an expert system that performs the same tasks that a visualization expert performs to design a dashboard based on client requirements. In the end, the user, domain and data requirements can be structured for a system to analyze and provide as an output a dashboard configuration.

The majority of the solutions are focused on a few factors to tailor dashboards (user preferences, goals, etc.), but these factors should be combined as they are all related. User preferences must be taken into account, but the data structure is crucial for delivering well designed visualizations, as well as the goals of the user; a visualization could fit into the user requirements and be compatible with the data structure, but it might not meet the user goals, thus having as a result an ineffective (or at least a not-that-effective) visualization.

6 CONCLUSIONS

There are several dashboard tools that allow customization, personalization and/or adaptation. A systematic mapping of the literature has been performed to understand the state-of-the-art of these solutions regarding their tailoring capabilities.

Dashboards are powerful tools that enable users to reach insights about certain topics, but the great number of potential end-users imply a great number of user profiles and requirements. Managing these requirements is not a trivial task, and several methods can be applied to address this issue.

The presented mapping of the literature aims at answering questions regarding tailored dashboards: their application context, targets and factors that the tailoring process, etc., providing a basis for identifying research opportunities in the area of tailored dashboards.

ACKNOWLEDGMENTS

This research work has been supported by the Spanish *Ministry of Education and Vocational Training* under an FPU fellowship (FPU17/03276). This work has been partially funded by the Spanish Government Ministry of Economy and Competitiveness throughout the DEFINES project (Ref. TIN2016-80172-R) and the Ministry of Education of the Junta de Castilla y León (Spain) throughout the T-CUIDA project (Ref. SA061P17).

REFERENCES

[1] SARIKAYA, A., CORRELL, M., BARTRAM, L., TORY, M., and FISHER, D., 2018. What Do We Talk About When We Talk About Dashboards? *IEEE Transactions on Visualization Computer Graphics*.

[2] BRESCIANI, S. and EPPLER, M.J., 2015. The pitfalls of visual representations: A review and classification of common errors made while designing and interpreting visualizations. *Sage Open 5*, 4, 2158244015611451.

[3] SUNDAR, S.S. and MARATHE, S.S., 2010. Personalization versus customization: The importance of agency, privacy, and power usage. *Human Communication Research 36*, 3, 298-322.

[4] KITCHENHAM, B.A., BUDGEN, D., and BRERETON, O.P., 2011. Using mapping studies as the basis for further research–a participant-observer case study. *Information Software Technology 53*, 6, 638-651.

[5] KITCHENHAM, B. and CHARTERS, S., 2007. *Guidelines for performing Systematic Literature Reviews in Software Engineering. Version 2.3.* Technical Report Report. School of Computer Science and Mathematics, Keele University.

[6] PETERSEN, K., VAKKALANKA, S., and KUZNIARZ, L., 2015. Guidelines for conducting systematic mapping studies in software engineering: An update. *Information Software Technology 64*, 1-18.

[7] MOHER, D., LIBERATI, A., TETZLAFF, J., ALTMAN, D.G., and THE PRISMA GROUP, 2009. Preferred reporting items for systematic reviews and meta-analyses: the PRISMA statement. *PLoS medicine 6*, 7, e1000097.

[8] VÁZQUEZ-INGELMO, A., 2019. Resources for the paper "Tailored information dashboards: A systematic mapping of the literature". DOI= http://dx.doi.org/10.5281/zenodo.2648377.

[9] PALPANAS, T., CHOWDHARY, P., MIHAILA, G., and PINEL, F., 2007. Integrated model-driven dashboard development. *Information Systems Frontiers 9*, 2-3, 195-208.

[10] MIOTTO, G.L., MAGNONI, L., and SLOPER, J.E., 2011. The TDAQ Analytics Dashboard: a real-time web application for the ATLAS TDAQ control infrastructure. In *Journal of Physics: Conference Series* IOP Publishing, 022019.

[11] ELIAS, M. and BEZERIANOS, A., 2011. Exploration views: understanding dashboard creation and customization for visualization novices. In *IFIP Conference on Human-Computer Interaction* Springer, 274-291.

[12] KINTZ, M., 2012. A semantic dashboard description language for a process-oriented dashboard design methodology. *Proceedings of MODIQUITOUS.*

[13] FILONIK, D., MEDLAND, R., FOTH, M., and RITTENBRUCH, M., 2013. A customisable dashboard display for environmental performance visualisations. In *International Conference on Persuasive Technology* Springer, 51-62.

[14] BELO, O., RODRIGUES, P., BARROS, R., and CORREIA, H., 2014. Restructuring Dynamically Analytical Dashboards Based on Usage Profiles. In *International Symposium on Methodologies for Intelligent Systems* Springer, 445-455.

[15] LOGRE, I., MOSSER, S., COLLET, P., and RIVEILL, M., 2014. Sensor data visualisation: a composition-based approach to support domain variability. In *European Conference on Modelling Foundations and Applications* Springer, 101-116.

[16] RADOVANOVIĆ, S., MAJSTOROVIĆ, B., KUKOLJ, S., and BJELICA, M.Z., 2014. Device Cloud platform with customizable Remote User Interfaces. In *2014 IEEE Fourth International Conference on Consumer Electronics Berlin (ICCE-Berlin)* IEEE, 202-204.

[17] NASCIMENTO, B.S., VIVACQUA, A.S., and BORGES, M.R., 2016. A flexible architecture for selection and visualization of information in emergency situations. In *2016 IEEE International Conference on Systems, Man, and Cybernetics (SMC)* IEEE, 003317-003322.

[18] VAN HOECKE, S., HUYS, C., JANSSENS, O., VERBORGH, R., and VAN DE WALLE, R., 2015. Dynamic Monitoring Dashboards Through Composition of Web and Visualization Services. In *International Internet of Things Summit* Springer, 465-474.

[19] MAYER, B. and WEINREICH, R., 2017. A dashboard for microservice monitoring and management. In *2017 IEEE International Conference on Software Architecture Workshops (ICSAW)* IEEE, 66-69.

[20] KUMAR, K., BOSE, J., and SONI, S.K., 2017. A Generic Visualization Framework based on a Data Driven Approach for the Analytics data. In *2017 14th IEEE India Council International Conference (INDICON)* IEEE, 1-6.

[21] KINTZ, M., KOCHANOWSKI, M., and KOETTER, F., 2017. Creating User-specific Business Process Monitoring Dashboards with a Model-driven Approach. In *MODELSWARD*, 353-361.

[22] MICHEL, C., LAVOUÉ, E., GEORGE, S., and JI, M., 2017. Supporting awareness and self-regulation in project-based learning through personalized dashboards. *International Journal of Technology Enhanced Learning 9*, 2/3, 204-226.

[23] SANTOS, H., DANTAS, V., FURTADO, V., PINHEIRO, P., and MCGUINNESS, D.L., 2017. From data to city indicators: a knowledge graph for supporting automatic generation of dashboards. In *European Semantic Web Conference* Springer, 94-108.

[24] DABBEBI, I., IKSAL, S., GILLIOT, J.-M., MAY, M., and GARLATTI, S., 2017. Towards Adaptive Dashboards for Learning Analytic: An Approach for Conceptual Design and implementation. In *9th International Conference on Computer Supported Education (CSEDU 2017)*, 120-131.

[25] NOONPAKDEE, W., KHUNKORNSIRI, T., PHOTHICHAI, A., and DANAISAWAT, K., 2018. A framework for analyzing and developing dashboard templates for small and medium enterprises. In *2018 5th International Conference on Industrial Engineering and Applications (ICIEA)* IEEE, 479-483.

[26] CARDOSO, A., VIEIRA TEIXEIRA, C.J., and SOUSA PINTO, J., 2018. Architecture for Highly Configurable Dashboards for Operations Monitoring and Support. *Studies in Informatics Control 27*, 3, 319-330.

[27] VÁZQUEZ-INGELMO, A., GARCÍA-PEÑALVO, F.J., and THERÓN, R., 2018. Domain engineering for generating dashboards to analyze employment and employability in the academic context. In *Proceedings of the 6th International Conference on Technological Ecosystems for Enhancing Multiculturality* (Salamanca, Spain2018).

[28] PASTUSHENKO, O., HYNEK, J., and HRUŠKA, T., 2018. Generation of test samples for construction of dashboard design guidelines: impact of color on layout balance. In *World Conference on Information Systems and Technologies* Springer, 980-990.

[29] YALÇIN, M.A., ELMQVIST, N., and BEDERSON, B.B., 2018. Keshif: Rapid and expressive tabular data exploration for novices. *IEEE Transactions on Visualization Computer Graphics 24*, 8, 2339-2352.

[30] ARJUN, S., 2018. Personalizing data visualization and interaction. In *Adjunct Publication of the 26th Conference on User Modeling, Adaptation and Personalization* ACM, 199-202.

[31] PETASIS, G., TRIANTAFILLOU, A., and KARSTENS, E., 2017. YourDataStories: Transparency and Corruption Fighting Through Data Interlinking and Visual Exploration. In *International Conference on Internet Science* Springer, 95-108.

[32] FEW, S., 2006. Information dashboard design.

[33] KAUR, P. and OWONIBI, M., 2017. A Review on Visualization Recommendation Strategies.

[34] KAKAS, A.C., 2017. A/B Testing.

7.8 Appendix H. Information dashboards and tailoring capabilities – A systematic literature review

Received July 28, 2019, accepted August 1, 2019, date of publication August 6, 2019, date of current version August 21, 2019.
Digital Object Identifier 10.1109/ACCESS.2019.2933472

Information Dashboards and Tailoring Capabilities - A Systematic Literature Review

ANDREA VÁZQUEZ-INGELMO[1], FRANCISCO J. GARCÍA-PEÑALVO[1], (Member, IEEE), AND ROBERTO THERÓN[1,2]
[1]GRIAL Research Group, Computer Science Department, Research Institute for Educational Sciences, University of Salamanca, 37008 Salamanca, Spain
[2]VisUSAL Research Group, University of Salamanca, 37008 Salamanca, Spain

Corresponding author: Andrea Vázquez-Ingelmo (andreavazquez@usal.es)

This work was supported in part by the Spanish *Ministry of Education and Vocational Training* through the FPU Fellowship under Grant FPU17/03276, in part by the Spanish Government Ministry of Economy and Competitiveness throughout the DEFINES Project under Grant TIN2016-80172-R, in part by the PROVIDEDH Project, funded within the CHIST-ERA Programme, MINECO, Spain, through the National Grant Agreement, under Grant PCIN-2017-064, and in part by the Ministry of Education of the Junta de Castilla y León, Spain, throughout the T-CUIDA Project under Grant SA061P17.

ABSTRACT The design and development of information dashboards are not trivial. Several factors must be accounted; from the data to be displayed to the audience that will use the dashboard. However, the increase in popularity of these tools has extended their use in several and very different contexts among very different user profiles. This popularization has increased the necessity of building tailored displays focused on specific requirements, goals, user roles, situations, domains, etc. Requirements are more sophisticated and varying; thus, dashboards need to match them to enhance knowledge generation and support more complex decision-making processes. This sophistication has led to the proposal of new approaches to address personal requirements and foster individualization regarding dashboards without involving high quantities of resources and long development processes. The goal of this work is to present a systematic review of the literature to analyze and classify the existing dashboard solutions that support tailoring capabilities and the methodologies used to achieve them. The methodology follows the guidelines proposed by Kitchenham and other authors in the field of software engineering. As results, 23 papers about tailored dashboards were retrieved. Three main approaches were identified regarding tailored solutions: customization, personalization, and adaptation. However, there is a wide variety of employed paradigms and features to develop tailored dashboards. The present systematic literature review analyzes challenges and issues regarding the existing solutions. It also identifies new research paths to enhance tailoring capabilities and thus, to improve user experience and insight delivery when it comes to visual analysis.

INDEX TERMS SLR, systematic literature review, tailoring, custom, personalized, adaptive, information dashboards.

I. INTRODUCTION

Information dashboards are nowadays key tools for understanding and extracting knowledge from large datasets, but they can take many forms. Information dashboards can be employed for different goals, to analyze different datasets (framed within different domains), to explain concepts, to generate knowledge, to confirm hypotheses, etc., [1]. The spread of dashboards and their use in different contexts makes their definition a complex task.

The associate editor coordinating the review of this manuscript and approving it for publication was Osama Sohaib.

Although identifying what is and what is not an information dashboard can be confusing in some cases, an information dashboard can be defined as a set of (visual) resources that enable its audience to understand and/or reach insights regarding the data being displayed [1]–[3].

Their capabilities not only try to cover the exploitation of datasets but also to provide a proper user experience to ease knowledge discovery. However, user experience, as the name suggests, depends on each user, there is no "one size fits all" in this domain. Although a "one size" dashboard, valid and useful for every possible user profile would be ideal, it is utopic; not every user is driven by the same goals, not every user is interested in the same data, not every user has the same visualization literacy, and so on. These aspects include not only personal preferences, but also social factors, like biases, beliefs, or past experiences [4], [5].

The support that technology provides to our everyday life has led to an exponential growth of data, making it necessary and crucial to take advantage of information to perform informed decision-making processes. Data are more accessible, and thus, not only specific profiles are in charge of visual analyses. Some users might need solutions that not only let them configure or develop dashboards given their requirements, but also that assist them in choosing a proper configuration if they don't have enough experience with visual analyses or enough visual literacy. Users should be provided with tailored dashboards that fulfill their requirements and foster insight delivery to enhance the outcomes of the decisions made.

Given these facts, it is essential to take into account final users when developing information dashboards, to improve the user experience and subsequently provide a dashboard that promotes knowledge generation. The user-centered design paradigm tries to address these issues by focusing on the user needs and requirements during all the development phases [6]. Involving the end-user into the design processes supports the development of better systems, which are useful for them and match their needs.

While necessary, this paradigm still lacks individualism when providing a solution, as not every potential user of the system can be involved in a development process. These potential users can present very different characteristics, mental schemas, and goals and therefore can demand very different features, especially in the dashboards domain that is faced in this work, given its complexity, so each person should be taken into account. However, is it efficient to develop an individual dashboard for each user? Should several quantities of resources be involved for the benefit of individualism? There exist any other approaches for designing and for building information dashboards for several and different user profiles?

Personalization and customization approaches try to address these individualization issues by tailoring products through different mechanisms. These mechanisms aim at supporting developers to configure products by reusing components and consequently, by decreasing the development time (even by assisting users in configuring their own products driven by their own needs). In the case of dashboards, there exist user-friendly tools that enable users to create and customize their dashboards without requiring any programming skills, like Tableau[1] or Grafana[2]. This kind of approaches give freedom to the users to configure their tools, but in such a complex domain that is visual analytics, some users might not exactly know which configuration is the best to accomplish their goals [7].

It is clear that dashboards are valuable but sophisticated tools, and their potential benefits when supporting decision-making processes has increased their popularity in several fields (business intelligence, learning analytics, services monitoring, etc.) and activities. Sarikaya *et al.* shown in a recent survey the relevance of researching on these tools and the relevance of users' goals, their characteristics, and context for designing useful dashboards [1]. However, before tackling how a tailored dashboard can be efficiently delivered to a specific user, it is necessary to understand and explore existing research lines and solutions regarding this domain. Laying a foundation on tailored dashboards can help to design better solutions based on case studies found in literature, analyzing their strengths and weaknesses.

A systematic literature review of existing tailoring methods regarding information dashboards has been carried out to clarify this matter. Through this review, the authors aim at providing a comprehensive view of this domain's solutions, to examine new research paths and opportunities for delivering effective tailored dashboards, and to learn about the trends and methods regarding the problem of finding a suitable dashboard configuration given a concrete user. Also, this systematic literature review can help to identify caveats or research opportunities to improve tailoring processes and to obtain more practical, usable, and individualized dashboards subsequently.

The term "tailored dashboard" is used throughout this work to enclose any dashboard solution that can vary its appearance and functionalities to match the users', data's and context's requirements, be them explicit requirements or implicit requirements. A general term is necessary, because using "customizable," "personalized" or "adaptive" indistinctly to refer to these solutions, could lead to misconceptions around these last terms, which, in the end, have different nuances.

As it will be exposed, tailored dashboards can be categorized taking into account a series of factors like the stage at which the tailoring process is performed, the driver of the tailoring process, the targets of the tailoring process, etc. This categorization shows that although the outcomes are "the same" (tailored dashboards), the methods to provide them can differ from each other (customization, personalization, adaptation, etc.).

The rest of this work is organized as follows. Section two (Methodology) describes the methodology, and the steps followed to perform the review. Section three (Review Planning) details the SLR planning phase. Section four (Review Process) presents the review and data extraction steps. Section five (Results) presents the results obtained from the analysis of the selected works to answer the research questions. Section six (Discussion) discusses the results, followed by section seven (Threats to Validity), in which the threats to the validity of the review are outlined. Finally, section eight

[1] https://www.tableau.com/
[2] https://grafana.com/

(Conclusions) includes some conclusions and future research lines.

II. METHODOLOGY

A systematic process has been followed to conduct the present literature review; specifically, the systematic literature review (SLR) methodology by Kitchenham [8] and Kitchenham and Charters [9]. The SLR has been complemented with a systematic mapping of the literature following the methodology proposed in [10]. The mapping results can be consulted in [11]. In this section, the protocol followed in carrying out the SLR is described, providing all the necessary information to trace the subsequent results. Following the [8], [9] guidelines, the SLR is composed of three main phases: planning, conducting, and reporting the study. These phases are detailed through the following sections.

Before planning the present SLR, a preliminary search was made to verify that no recent SLRs about tailored dashboards were carried out. If that were the case, there would not be any necessity to conduct a new one. This verification was performed by searching through different electronic databases (Scopus, Web of Science (WoS), IEEE Xplore and Springer) terms related to the methodology (''SLR'', ''systematic literature review'', etc.) and the target of the review (''tailored'', ''customizable'', ''personalized'', etc., along with the term ''dashboards''). The outcomes of these queries confirmed that currently, there are no previous systematic literature reviews about the thematic addressed in this work, justifying the execution of this SLR.

III. REVIEW PLANNING

The review planning process involves the identification and definition of different aspects to lay the foundations of the review execution, such as posing the questions to be answered, detailing the protocol followed, and any other relevant information to make the review traceable. These different aspects are described in this section.

A. RESEARCH QUESTIONS

First, a series of research questions have been raised. These questions can be classified into three main blocks: technical aspects (RQ1-RQ4), artificial intelligence (AI) application (RQ5), evaluation of the solutions (RQ6).

- **RQ1.** How have existing dashboard solutions tackled the necessity of tailoring capabilities?
- **RQ2.** Which methods have been applied to support tailoring capabilities within the dashboards' domain?
- **RQ3.** How the proposed solutions manage the dashboard's requirements?
- **RQ4.** Can the proposed solutions be transferred to different domains?
- **RQ5.** Has any artificial intelligence approach been applied to the dashboards' tailoring processes and, if applicable, how these approaches have been involved in the dashboards' tailoring processes?
- **RQ6.** How mature are tailored dashboards regarding their evaluation?

The first RQs block aims at answering questions regarding how tailoring capabilities have been materialized in tangible dashboard solutions (methods, requirements management, domain transferability). The goal of answering RQ5 is to identify research opportunities in terms of the application of AI mechanisms to support the dashboards' tailoring processes automatically. The last question's purpose is to understand if the solutions found have been tested with end-users and if the tailoring capabilities have been useful for enhancing insight delivery and knowledge generation.

As mentioned before, the SLR has been complemented with a literature mapping to perform a quantitative analysis of the domain and to obtain a broad view of the research area. The following mapping questions (MQs) were posed, but the outcomes of the mapping are out of the scope of this paper and can be consulted at [11]:

- **MQ1.** How many studies were published over the years?
- **MQ2.** Who are the most active authors in the area?
- **MQ3.** What type of papers are published?
- **MQ4.** To which contexts have been the variability processes applied? (BI, learning analytics, etc.)
- **MQ5.** Which are the factors that condition the dashboards' variability process?
- **MQ6.** What is the target of the variability process? (visual components, KPIs, interaction, the dashboard as a whole, etc.)
- **MQ7.** At which development stage is the variability achieved?
- **MQ8.** Which methods have been used for enabling variability?
- **MQ9.** How many studies have tested their proposed solutions in real environments?

The systematic mapping performed at [11] employs the same approach as in the present SLR. However, the mapping provides an overview of the research area by identifying and classifying the available evidence, while the following SLR results involve the analysis and interpretation of the evidence found [12] to answer the specific research questions posed at the beginning of this subsection.

Given the previous research questions, the PICOC method proposed by Petticrew and Roberts [13] has been followed to define the review scope.

- **Population (P):** Software solutions
- **Intervention (I):** Provide support to tailor (information) dashboards
- **Comparison (C):** No comparison intervention in this study, as the primary goal of the present SLR is to analyze existing approaches regarding tailoring capabilities and gain knowledge about them.
- **Outcomes (O):** Information dashboard proposals
- **Context (C):** Environments related to data visualization and (or) decision making (in the academia, industry, etc.)

B. INCLUSION AND EXCLUSION CRITERIA

Once the scope of the review has been established, a series of inclusion (IC) and exclusion criteria (EC) are defined to select relevant works for answering the identified research questions. If a work does not meet the whole set of inclusion criteria or does meet any exclusion criterion, it will be excluded from the review.

- **IC1.**The paper describes a dashboard solution (proposal, architecture, software design, model, tool, etc.) AND
- **IC2.** The solution is applied to information dashboards AND
- **IC3.** The solution supports or addresses tailoring capabilities (customization, personalization, adaptation, variation) regarding information dashboards AND
- **IC4.** The tailoring capabilities of the dashboard are related to its design, components or KPIs AND
- **IC5.** The papers are written in English or Spanish AND
- **IC6.** The papers are published in peer-reviewed Journals, Books or Conferences AND
- **IC7.** The publication is the most recent or complete of the set of related publications regarding the same study

The exclusion criteria are derived from the inclusion criteria as their opposite.

- **EC1.**The paper does not describe a dashboard solution (proposal, architecture, software design, model, tool, etc.) OR
- **EC2.** The solution is not applied to information dashboards OR
- **EC3.** The solution does not support or address tailoring capabilities (customization, personalization, adaptation, variation) regarding information dashboards OR
- **EC4.** The tailoring capabilities of the dashboard are not related to its design, components or KPIs OR
- **EC5.** The papers are not written in English or Spanish OR
- **EC6.** The papers are not published in peer-reviewed Journals, Books or Conferences OR
- **EC7.** The publication is not the most recent or complete of the set of related publications regarding the same study

The IC5 includes the Spanish language, because the main research terms, as it will be seen in the next subsection, are compatible with their Spanish equivalent terms (dashboard* along with custom*, personal*, adapt*, flexib* and config*). As a consequence, works written in Spanish could be retrieved through the search string and could be potentially included in the review given the authors' comprehension of this language.

C. SEARCH STRATEGY

It is necessary to identify the most important databases regarding the research context in which the queries will be performed to obtain relevant outcomes from the search. In this case, four electronic databases were selected: Scopus, Web of Science (WoS), IEEE Xplore, and SpringerLink. These databases were chosen according to the following criteria:

- It is a reference database in the research scope
- It is a relevant database in the research context of this literature review
- It allows using similar search strings to the rest of the selected databases as well as using Boolean operators to enhance the outcomes of the retrieval process

Regarding the search concepts employed to build the search query, the following terms were included:

- The "meta-dashboard" concept to search for solutions that employ a meta-modeling approach to extract common and abstract features from dashboards that can be applied for tailoring processes.
- Related terms to tailoring capabilities: tailored, customized, personalized, adaptive, flexible, configurable, context-aware, etc., along with the word "dashboard," which is the main target of the review.
- Other terms like "selection," "composition," or "generation" to search for generative solutions that provide dashboards as a result of a generation, composition or selection process of suitable visualizations and features.
- The term "template" to retrieve works that use dashboard templates that can be configured to fit specific requirements (this term can also be related to generative processes)
- The term "driven" to enclose works that use context-driven, data-driven, user-driven, etc., approaches, thus being necessary to take into account these factors to develop the dashboards
- Additional terms related to heterogeneous requirements and diverse necessities to retrieve works that do not mention directly any of the above terms, but do implicitly refer to them by calling upon the heterogeneity of dashboards requirements and the involved user profiles, thus potentially addressing these issues by tailoring mechanisms.

Finally, given the fact that the word "dashboard" is also employed for referring to cars' control panels, words related to the automotive area ("car," "vehicle," "automotive") were excluded to avoid irrelevant papers outside the scope of information dashboards.

D. QUERY STRINGS

The search strings for each chosen source were built using relevant search terms derived from the PICOC methodology outcomes, connected by Boolean AND / OR / NEAR operators. Moreover, the wildcard (*) was used to enclose both the singular and plural of each term.

The NEAR operator enables the user to retrieve works where the terms joined by this operator are separated by an interval of words explicitly specified. This operator is handy in the context of the present research, as the terms "customizable," "personalized," "adaptive," etc. should only refer to the dashboard term, to avoid works that are not explicitly focused on the tailoring capabilities of dashboards. However, the drawback of using this operator is the necessity to explicitly define the maximum number of words that can separate the target terms.

In this case, the chosen number was 10 (i.e., the "dashboard" term and the rest of the terms will be within 10 number of words of each other). This number was selected after performing a "simulation" by executing the same search with different proximity values (5, 7, 10, and 12). Examining the titles, abstracts and keywords of the additional records found after incrementing this value, it was concluded that the ten value would retrieve relevant works without adding noise (i.e., irrelevant works), meaning that the terms affected by the NEAR operator are potentially in the same sentence, given average sentence length guidelines and evidence [14], [15].

Once the NEAR operator value was selected, the specific query strings for each chosen database were specified using their query syntax.

1) SCOPUS

TITLE-ABS-KEY ((meta-dashboard) OR ((dashboard*) W/10 (custom* OR personal* OR adapt* OR flexib* OR config* OR tailor* OR context-aware OR generat* OR compos* OR select* OR template* OR driven)) OR ((dashboard*) AND ((heterogeneous OR different OR diverse OR dynamic) W/0 (requirement* OR stakeholder* OR user* OR need* OR task* OR necess*)))) AND NOT TITLE-ABS-KEY (car OR vehicle OR automo*) AND NOT DOCTYPE(cr)*

2) WEB OF SCIENCE

TS=((meta-dashboard) OR ((dashboard*) NEAR/10 (custom* OR personal* OR adapt* OR flexib* OR config* OR tailor* OR context-aware OR generat* OR compos* OR select* OR template* OR driven)) OR ((dashboard*) AND ((heterogeneous OR different OR diverse OR dynamic) NEAR/0 (requirement* OR stakeholder* OR user* OR need* OR task* OR necess*)))) NOT TS= (car OR vehicle OR automo*)*

3) IEEE XPLORE

(((meta-dashboard) OR ((dashboard) NEAR/10 (custom OR personal* OR adapt* OR flexib* OR tailor OR tailored OR configurable OR context-aware OR generation OR generated OR generative OR composed OR composition OR selection OR selecting OR template OR driven)) OR ((dashboard) AND ((heterogeneous OR different OR diverse OR dynamic) NEAR/0 (requirement OR stakeholder OR user OR need OR task OR necessities)))) AND NOT (car OR vehicle OR automo*))*

4) SPRINGERLINK

((meta-dashboard) OR ((dashboard*) NEAR/10 (custom* OR personal* OR adapt* OR flexib* OR config* OR tailor* OR context-aware OR generat* OR compos* OR select* OR template* OR driven)) OR ((dashboard∗) AND ((heterogeneous OR different OR diverse OR dynamic) NEAR/0 (requirement* OR stakeholder* OR user* OR need* OR task* OR necess*))))*

In case of SpringerLink, this query string was complemented with an additional restriction, given SpringerLink policy of searching the query terms along with the papers' full-text (which includes huge amounts of noise to the review process). Through the advanced search tool, the query results were limited to those that have the term "dashboard*" in their titles, additionally to the search string terms in their full-texts to ensure that the main focus of the retrieved works is information dashboards.

E. QUALITY CRITERIA

Although the inclusion and exclusion criteria are useful for including in the review relevant works in terms of the scope of the literature review, they don't address the quality of the retrieved papers regarding their capacity to answer the posed research questions. A new set of criteria has been defined to check the works' quality before including them into the final literature review. Each criterion can be scored with three values: 1 (the paper meets the criterion), 0.5 (the paper partially meets the criterion) and 0 (the paper does not meet the criterion).

1. The research goals of the work are focused on addressing the variability, adaptability, customization or personalization of an information dashboard to improve individual user experience (UX)
   - *Partial: not every research goal tries to address UX through tailoring capabilities*
2. A software solution that supports the variability of the dashboard components is presented
   - *Partial: the software supports customization of the dashboard but is not the focus*
3. A model, framework, architecture or any software engineering artifact that address the variation of the dashboard components and interaction methods are properly exposed
   - *Partial: a model, framework, architecture or any software engineering artifact is exposed but not detailed, i.e., the nature of the referred elements is mentioned, but their internal structures and details are not further explained.*
4. The employed methods or paradigms to achieve tailoring capabilities are properly described
   - *Partial: the employed methods or paradigms to achieve tailoring capabilities are partially described, i.e., the methodology is mentioned, but how the methodology has been particularly used within the application context is not detailed.*
5. The context or domain of application of the dashboard is described
   - *Partial: the context or domain of application is mentioned but not detailed.*

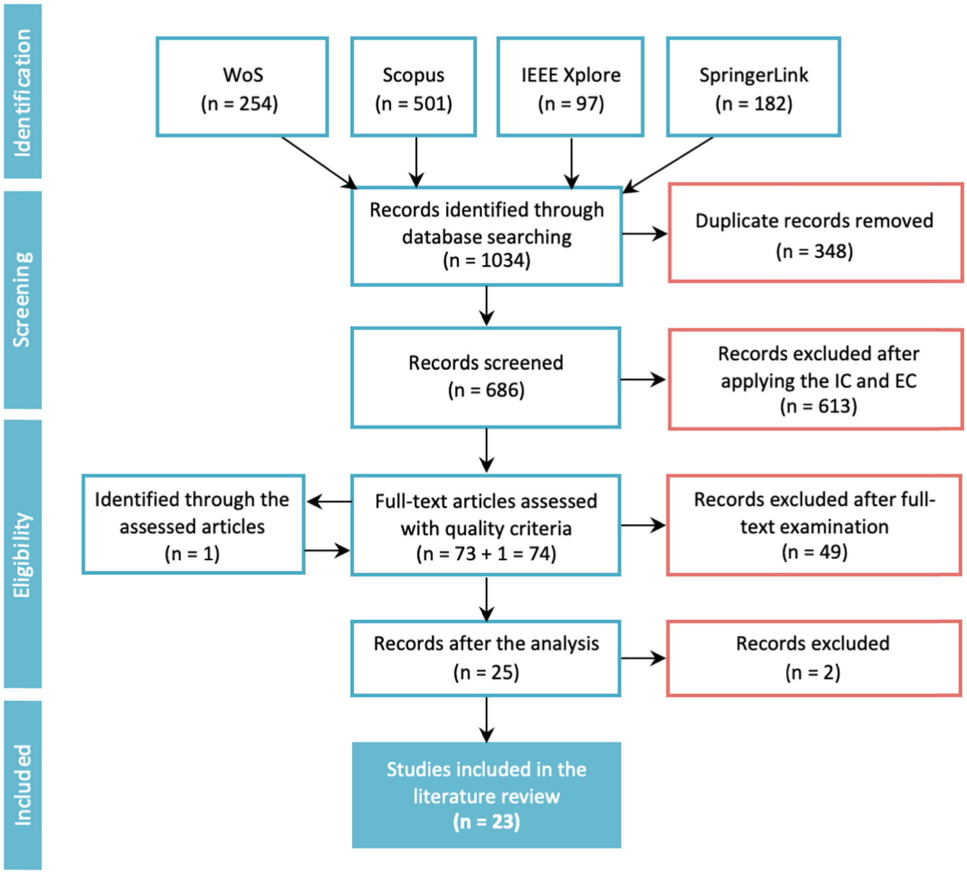


FIGURE 1. **Phases and outcomes of the review process using the PRISMA flow diagram.**

6. The proposed solution has been tested with real users
   - *Partial: real users have used it and tested its functionality, but no further testing has been performed*
7. Issues or limitations regarding the proposed solution are identified
   - *Partial: issues or limitations are mentioned but not detailed*

Each paper can obtain a maximum of 7 points regarding its quality following this methodology. This 0-to-7 score was transformed into a 0-to-10 scale, and the seven value was chosen as the threshold for including a paper into the final synthesis. If in a 0-to-10 scale, a paper obtains a score of fewer than seven points, it will be dismissed from the review as it did not meet a minimum quality to answer the stated research questions.

The chosen threshold ensures that the works have obtained the maximum score in some criteria, without neglecting the rest of the quality statements. With this threshold, a paper is limited to a maximum of two criteria with a 0 score to reach the next phase, ensuring that the majority of the criteria is always fully or partially met.

IV. REVIEW PROCESS

The data gathering process to conduct the present SLR has been divided into different phases in which various activities are carried out. The PRISMA flow diagram [16] has been employed to detail the actions performed during the data extraction (Figure 1).

Once the search was performed (on January 22, 2019), the paper selection process was carried out through the following process:

1. The raw results (i.e., the records obtained from each selected database) were gathered in a GIT repository[3] [17] and arranged into a spreadsheet[4]. A total of 1034 papers were retrieved: 254 from Web of Science, 501 from Scopus, 97 from IEEE Xplore and 183 from SpringerLink.
2. After organizing the records, duplicate works were removed. Specifically, 348 records were removed, retaining 686 works (66.34% of the raw records) for the next phase.

[3]https://github.com/AndVazquez/slr-tailored-dashboards
[4] http://bit.ly/2wRCU5w

3. The maintained papers were analyzed by reading their titles, abstracts, and keywords and by applying the inclusion and exclusion criteria. A total of 613 papers were discarded as they didn't meet the criteria, retaining 73 papers (10.79% of the unique papers retrieved) for the next phase.
4. The selected 73 papers were read in detail and further analyzed. The papers were scored regarding their quality to answer the research questions using the quality assessment checklist described in the previous section. One paper was added after checking the references of the assessed works, leaving 74 records for this quality assessment phase.
5. After applying the quality criteria, a total of 23 papers (3.35% of the unique papers retrieved and 31.08% of the full-text assessed papers) were selected for the present review.

Two records were finally discarded. The reason for this exclusion was that the two works were previous versions of other studies found within the retrieved records. The decision was to keep the more complete and/or more recent work.

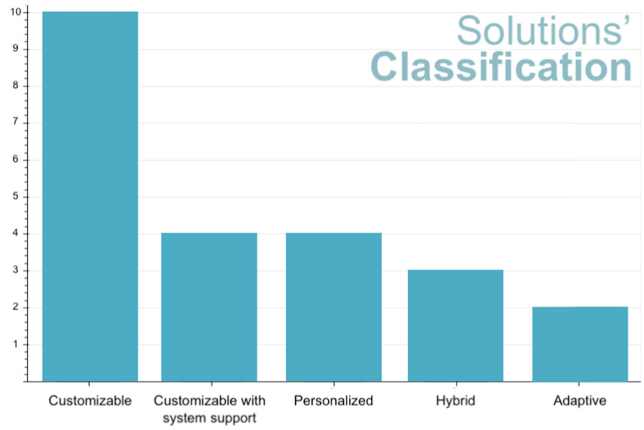


FIGURE 2. Classification of the retrieved solutions in terms of their tailoring method. Source: [11], elaborated by the authors.

V. RESULTS

A. HOW HAVE EXISTING DASHBOARD SOLUTIONS TACKLED THE NECESSITY OF TAILORING CAPABILITIES?

The first research question tries to answer, which are the trends when it comes to tailoring an information dashboard. As stated in the introduction of this work, some terms are misleading or not being appropriately used, given their formal meaning. ''Custom'' and 'personalized'' are often used as interchangeable terms with the same connotations. It is important to make distinctions among these terms, as they have entirely different meanings regarding their mechanisms.

The selected works were categorized in terms of their tailoring process. Each paper was analyzed to answer the questions that would frame the tailoring process employed (i.e., at which stage is the tailoring process performed? Who performs the tailoring process?). As shown in Figure 2, the majority of the selected works are framed in the category of ''customizable,'' meaning that the tailoring process of the dashboard is driven by explicit user requirements [18]–[27].

Most customizable solutions identified involve manual approaches (which will be detailed in RQ2), meaning that users need to perform a set of explicit actions to tailor their dashboard according to their needs.

In [18], a customizable dashboard display for monitoring mobile energy is presented; users can build their dashboards by selecting pre-defined widgets and data streams from different sources (sensors, government agencies, social media, and generic services). Manual approaches like those above are also used in [19], [26], [27], in which the customizability capacity is based on the possibility of arranging the components of the dashboard through end-user interaction, and even the ability to craft custom indicators, as described in [26].

However, not only manual user interactions are employed for arranging the tool, some of these customizable dashboards involve generative or automatized approaches through the specification of configuration files [21], [22], [24], models [23], [25] or pre-defined templates [20]. Although technically the tailoring process is indeed made by the system (not involving direct user actions to modify the dashboard appearance), which is a characteristic of personalization approaches, the data contained within the configuration files or model instances does involve explicit user requirements, so the dashboard is tailored according to the users by means of their requirements. In the end, these generative approaches add an abstraction layer which helps users to configure their dashboards without requiring programming skills. For these reasons, these solutions are also classified as customizable dashboards.

Despite the previous distinction about customizable dashboards, personalized solutions have also been identified. In this case, personalized solutions infer a suitable configuration based on implicit data about users, tasks, or goals [28]–[31]. In [28], the methodology takes as input a model of the business process and goals to describe and generate a dashboard, so the authors use implicit data (goals) to build a concrete dashboard that would help to reach the input goals. User-roles are also added to this methodology in [29], to include more information to the dashboard personalization process. A similar solution is presented in [30], which also takes into account user-roles and business' KPIs to generate a dashboard that fits the business goals. Finally, in [31], the focus is on personalizing the display taking into account the user abilities through an initial questionnaire that ask users if they have eye diseases or any tremor in hands, making the dashboard accessible if necessary. Once generated, these dashboards cannot be adapted at run-time, being essential to re-generate them.

Adaptive solutions, on the other side, can adapt themselves at run-time based on environmental changes. Belo *et al.* [32] present an adaptive dashboard that restructures itself given user-profiles and behaviors extracted from the dashboards' analytical sessions (i.e., through the analysis of the user queries). Another adaptive solution presented in [33] uses a dashboard generator fed with user, data and visualization models, thus generating information dashboards according to different contexts, and, in theory (as the proof of concept is not fully adaptive at the time of publishing the paper), adapting themselves given their users' interaction history.

Other two kinds of tailored solutions have been identified, as they cannot be framed on the last categories (customizable, personalized, or adaptive). On the one hand, solutions identified as "hybrid" are mainly personalized or adaptive dashboards that allow the user to have the last word regarding the dashboard configuration, or need user actions to complete the tailoring process. In [34], a device cloud platform dashboard is built based on the data model of the remote devices being monitored, but users can also customize it manually. Van Hoecke *et al.* [35] use a semantic reasoner to personalize indicators from available data sources, but the dashboard construction is still a user task. Santos *et al.* [36] also proposes personalized dashboards based on knowledge graphs and indicator ontologies, but allow the users to modify the dashboard recommendation to her or his preferences.

On the other hand, there are customizable solutions that can assist and help the users to build their dashboards according to a series of factors. The four papers identified in this category [37]–[40] use visual mapping to help users to determine the best visualization types for the data to be visualized while building and designing their dashboards. These solutions are mainly customizable dashboards with mechanisms that help users with the selection of a suitable dashboard configuration.

Classifying these tools regarding their tailoring capabilities is complex, as the selected papers present too many different solutions implemented through various methods with different goals, so this classification of tailored dashboards should be seen as a spectrum, allowing the existence of dashboards that mix features of different approaches. However, framing them in distinct categories, allow better understanding regarding existing solutions as well as regarding the current state of the present field.

B. WHICH METHODS HAVE BEEN APPLIED TO SUPPORT TAILORING CAPABILITIES WITHIN THE DASHBOARDS' DOMAIN?

This research question is tightly related to RQ1. In the end, the selected tailoring process narrows the potential methods to accomplish it. As shown in the first research question, the most common type of tailored dashboards are customizable dashboards, within the scope of this systematic literature review. Regarding these mechanisms, the preferred method for customizing dashboards is the use of configuration wizards that supports the users' decisions when building her or his customized dashboards without requiring programming skills. For example, [18], [19], [26], [27] use graphical user interfaces that ease the selection of widgets and the data to be displayed, following the workflow shown in Figure 3.

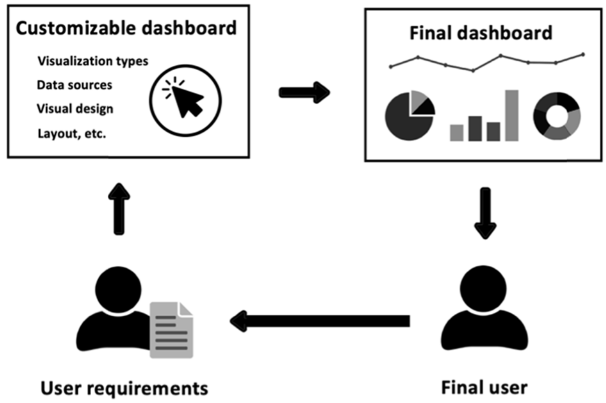


FIGURE 3. Customizable dashboard workflow.

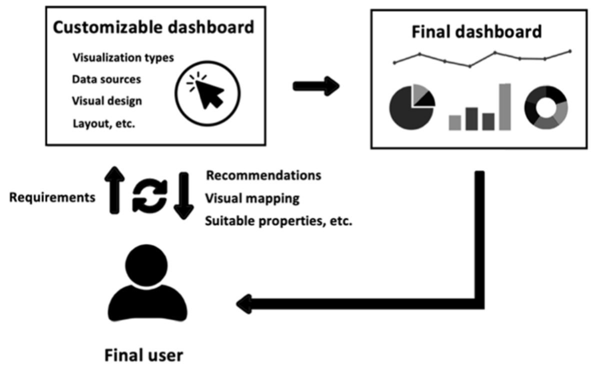


FIGURE 4. Customizable dashboard with system assistance workflow.

Configuration wizards are also the preferred method for customizable dashboards with system assistance, in conjunction with visual mapping methods that ease the selection of visualization types given the data types or structure [37]–[40]. Visual mapping is a transformation that matches data properties with visual marks or visual elements to obtain a suitable visualization for the selected data [41]. Figure 4 shows a generic workflow of how this approach work; users configure their dashboards based on their needs, and the system provides feedback to support the customization process and to obtain more effective dashboards potentially.

Another common method to customize dashboards is to configure them by using structured configuration files [21], [22], [24], which also allow users to tailor their dashboards with a higher level of abstraction (through JSON files, XML files, etc.) through richer and more domain-specific syntaxes than programming languages. Figure 5 shows the workflow of these configuration processes using configuration files, where a series of parameters are set to render a concrete and functional dashboard.

Some works also take advantage of the Software Product Line (SPL) paradigm [23], [25] or Model-Driven Development (MDD) [28]–[30]. In the case of SPL approaches applied to dashboards, they are based on the conception that dashboards are sets of components with optional, alternative,

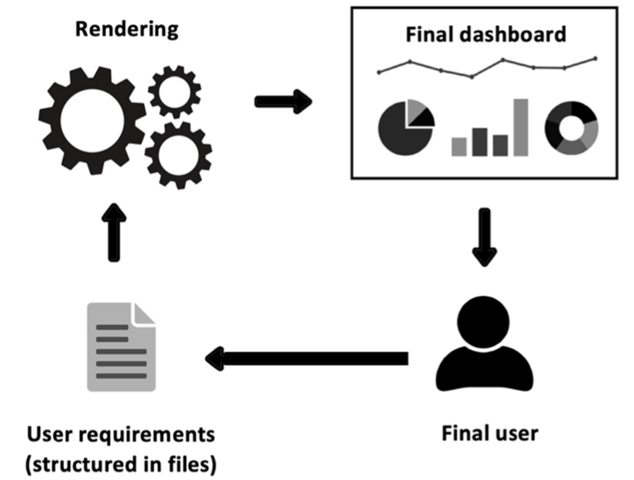


FIGURE 5. Dashboard configuration process involving files.

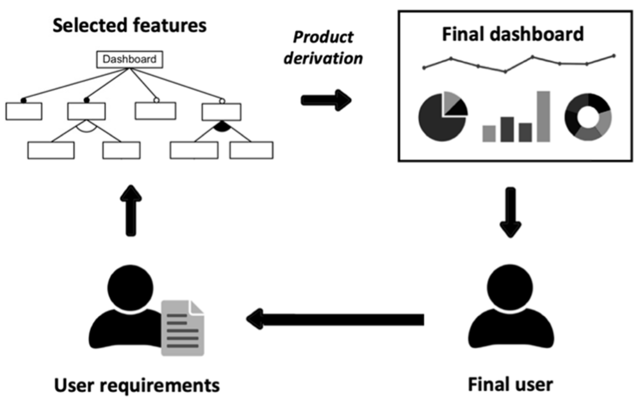


FIGURE 6. The software product line paradigm applied to dashboards.

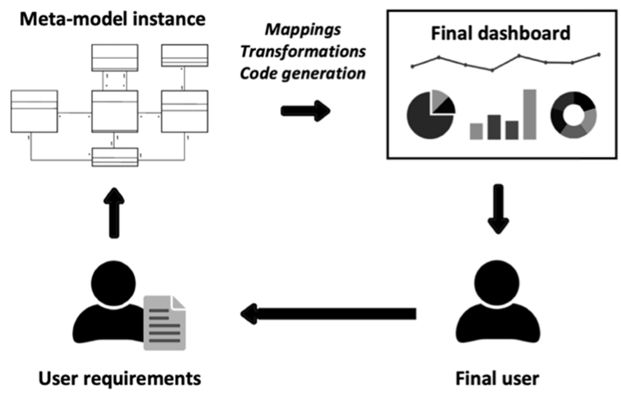


FIGURE 7. The MDD workflow applied to dashboard development.

or mandatory features. These paradigms are used to finally generate a dashboard that fits the previously defined feature model, as shown in Figure 6. Regarding the solution presented in [23], it is worth to mention that an extended version of this work can be found at [42]. This last work did not appear within the selected papers because it was published after the execution of the present SLR, but in subsequent updates, it would replace the previous paper, keeping the most complete and recent version of the study.

In the case of MDD approaches, the logic is similar; code generators are fed with a series of models that describe the dashboard at high-level, for example, as described in [28]. With a set of transformations and mappings, high-level models are transformed into concrete dashboards, through specific description files [28] or by using pre-defined or custom-made templates [30]. Figure 7 illustrates this approach.

In [30], the authors point out the necessity of having pre-defined templates in conjunction with the models, to materialize and generate the dashboards. The approach of using pre-defined templates is also present in [20], where authors propose a framework for creating different templates with different KPIs and goals for small and medium enterprises.

A similar MDD approach is followed in [33], although authors don't explicitly indicate that they followed this paradigm. In this case, to generate the dashboard, a context-aware generator with user, data and visualization models as inputs is in charge of generating the dashboard instances, but the internal features of the dashboard generator are not detailed.

Regarding adaptive solutions, agents are a common method for managing changing requirements [32], [34]. In [34], device cloud platform dashboards are adjusted through cloud agents that adapt themselves to the devices' data models, thus generating remote user interfaces based on the characteristics of the monitored devices. In [32], an analytical system is guided by agents that are present in five communities (gatherers, conciliators, providers, visualizers, and restructurers) to log user interactions with the system and reconfigure the dashboard accordingly.

Other methods found in the selected papers enclose inclusive user modeling for adapting the dashboard interface to the user abilities [31], semantic reasoners for selecting appropriate data sources and compositions [35] and knowledge graphs and ontologies to adapt the dashboards to the target data domain [36].

C. HOW THE PROPOSED SOLUTIONS MANAGE THE DASHBOARD'S REQUIREMENTS?

As introduced before, the necessity of tailored dashboards lies in a large number of existing user profiles that could potentially use these tools. Generic or "one size fits all" dashboards are relatively easier to implement than a specific dashboard for each end-user, because the latter approach is not scalable at all, as the number of users could increment and their requirements evolve. However, "one size fits all" dashboards lack of flexibility, and would only be effective and efficient for specific user profiles [43], because data that is relevant for one user could be irrelevant for another user, and vice versa, and could play different roles in their decision-making processes.

These are the reasons why tailored dashboards should be considered; to fulfill the requirements and necessities of each user profile simultaneously. But managing this high volume of requirements at once (that can even evolve) is not a trivial task.

That is why this question is to be answered; to learn how these solutions manage the requirements associated with each user and how they provide a tailored dashboard accordingly.

The second research question shown that configuration wizards are popular methods to manage these requirements by giving the user the responsibility of building their own dashboard based on their necessities. These solutions allow users to customize their displays while using their dashboards freely, thus performing the tailoring process at user-configuration time (i.e., at run-time, but with the intervention of the user through explicit actions). All solutions found that use a configuration wizard approach [18], [19], [26], [27], [37]–[40] manage individual user requirements by implementing authentication and account management services, associating each user to his/her dashboard configuration persistently. Even some solutions let users build visualizations without logging in to the system [40], in case the users do not need or do not want to save their own configuration for the future. This user management approach is also applied to other solutions found, like in [31], where a user creates an account and fills a questionnaire about her or his abilities to finally access her or his personalized view based on the previous information.

However, these works do no further discuss the storage method nor the possibility of storing different versions of a user dashboard over time, which could be very useful to collect the evolution of the preferences or user behavior.

On the other hand, 10 of the selected works take advantage of structured files or models to hold individual dashboard requirements that finally serve as inputs of generators that provide the configured dashboard instance meeting the original specifications. In this category fall those solutions based on configuration files [21], [22], [24], context models [33], software product lines [23], [25] or model-driven development [28]–[30]. In this case, user requirements are managed ''outside'' the dashboard systems, before their exploitation, and stored within individual files or models.

In the case of [20], no requirement management is explicitly performed, as the pre-defined templates enclose general requirements collected from the gathering and analysis phase, and subsequently, users select the template that fits better their needs. This management method allows better requirement traceability, as requirements are parsed and mapped from the specification to the concrete system features. Also, it allows an easier version control of each file or model, keeping the evolution of individual dashboard requirements.

Also, in the agent-based solution found [34], the system's cloud agent adapts to each device data-model and adds additional cloud agent information markers which act as a user interface description language. These markers are initially provided by the devices' data model but can be modified through the solution's web application. As the paper exposes, the devices' data models are held in XML format, so the requirements management is made through these device models, and therefore falling in the same category as these previous works based in models and structured files.

The other agent-based solution presented in [32] stores and modifies settings according to the users' behavior and their events, thus needing also authentication and account management services to work correctly, as discussed at the beginning of this research question. In this case system's agents are the drivers of the dashboard modifications.

The remaining solutions proposed, on the one hand, a semantic reasoner to infer potentially interesting compositions of data streams in the context of the Internet of Things (IoT) [35]. This solution personalizes the presented information by composing semantically annotated data and visualization services. However, as specified in the research question RQ1, the solution is classified as hybrid, as the paper states that ''sensor and data compositions need to be dynamically visualized, thereby limiting the user input to selecting the preferred visualization method from a system-generated list of meaningful options, taking into account the preferences and characteristics of the current user profile'' [35].

So the dashboard is personalized given the available sensor data (the dashboards' requirements are managed through reasoning processes and knowledge bases), but in the end, users need to select the widgets that will compose their dashboard, although this aspect of the system is no further discussed in the paper.

On the other hand, Santos *et al.* presented in [36] a knowledge graph and indicator ontology approach to automatically generate dashboards in the context of smart cities. The proposed dashboard generator takes as input serialized knowledge graphs and offers different dashboard configurations accordingly. The application allows the customization of the automatically proposed dashboard, given the user freedom to change the configuration before generating it. The dashboard information requirements are managed through knowledge graphs, but the user requirements management when manually customizing the dashboard is not further discussed.

D. CAN THE PROPOSED SOLUTIONS BE TRANSFERRED TO DIFFERENT DOMAINS?

Dashboards are used to exploit datasets that are usually large, but also these datasets come from different domains. This research question tries to answer the flexibility of the solution found regarding their transfer capabilities to another domain. In other words, can the solutions fully support the visualization of data from other domains without significant changes in the original code?

When a solution is focused on a particular data domain, it could be challenging to reuse that same solution for other data domains if the source code is coupled to the original goals, allowing tailoring capabilities, but only within the domain's frontiers.

Some solutions based on configuration wizards, like [38], [39], support the exploitation of datasets from different domains by allowing the users to upload or specify their concrete data sources. These solutions are robust as they can be reutilized for different goals depending on the data domain. The rest of solutions using configuration wizards allow freedom when configuring the dashboards, but only within the original domain (environmental performance [18], micro-services monitoring [19], emergency situations [37], learning analytics [26], physics [27] or economics [40]).

On the other hand, although the solution presented in [22] can be adapted to different monitoring scenarios, it cannot be transferred to other data domains as it relies on API endpoints to monitor resources. In the case of [21], the configuration files allow the specification of the data sources, which can be local as well as remote, and their associated elements, meaning that the solution can be applied to other data domains besides web analytics (which is the domain of the presented prototype). The dashboard generator detailed in [24] also allows the specification of the dataset to be represented, but this approach is mainly employed to develop studies regarding usability guidelines, so the data domain's transferability is not significantly relevant in this case.

Other solutions that take part in the configuration files approach, as discussed in the second research question, are the ones using MDD or SPL approaches. These solutions, which are based on meta-modeling [28]–[30], take advantage of high-abstraction levels and commonalities among the potential products [23], [25] to address the generation of different dashboards. Meta-modelling and domain engineering allow the abstraction of the dashboards' features, improving reusability of core assets and thus, making it possible to transfer the solutions to other data domains without significant efforts.

The dashboard presented in [31] is also tightly coupled to its original domain, as the adaptation of the dashboard is focused on the user physical abilities. The same is true for [20]; although alternative templates can be chosen to visualize different data aspects, they are always related to business intelligence (sales, human resources, overall equipment effectiveness, etc.). The template approach can be taken, of course, to address other data domains, but, new templates should be developed for each target domain to accomplish this "domain transfer." In the case of [36], the solution employs a knowledge graph and indicator ontologies to generate personalized dashboards; the ontologies used are related to the Smart City context, so, to transfer this methodology to other domains, ontologies related to them should be employed.

Some of the works are focused on sensor monitoring [35] and device clouds [34]. The methodologies employed in the papers mentioned above (semantic reasoners and multi-agent systems, respectively) could be reused for other domains, but in the end, the dashboard solutions would need to be built from zero to adapt them to new domains.

The remaining works, on the one hand, use agents to restructure dynamically a dashboard based on user behavior. This adaptive process is not coupled to the data domain, as there are a specific community of agents (called gatherers) that are responsible for collecting the data from different sources [32], suggesting that their task is to gather data no matter its domain. Finally, in [33], a dashboard generator fed with the user, visualization, and analysis scenario models is presented in the context of learning analytics. The user and visualization models are more generic and focused on the users' preferences, experience, goals, visualization purposes, etc., allowing their reuse on other data domains. However, the analysis scenario model is more coupled to the learning analytics domain, mentioning learning objectives, pedagogical context, fields of education, etc., so it could not be reused for domains outside the learning analytics context.

As a clarification, it is worth to state that every methodology employed in the selected papers could be applied to develop dashboards in different data domains. However, the purpose of this research question is to identify the most flexible and powerful solutions regarding their abstraction and, therefore, their potential reuse to other domains in an automatized manner (i.e., avoiding to develop the same solution for new domains manually).

E. HAS ANY ARTIFICIAL INTELLIGENCE APPROACH BEEN APPLIED TO THE DASHBOARDS' TAILORING PROCESSES AND, IF APPLICABLE, HOW THESE APPROACHES HAVE BEEN INVOLVED IN THE DASHBOARDS' TAILORING PROCESSES?

Again, dashboards deal with lots of data and requirements, and even generative approaches based on configuration files or generators still need from manual configuration through high-level languages or domain-specific languages. A similar issue arises from configuration wizards; in the end, users need, through actions, to specify requirements that are not always clear for themselves.

Using methods that involve artificial intelligence (AI) algorithms to manage the dashboards requirements could lead to more accurate dashboard configurations and decrease the consumed resources during the requirement elicitation phases, as requirements could be automatically inferred by the AI algorithm. With AI, systems can use algorithms to learn patterns from data and apply inference to predict future values. This approach would be potentially beneficial in the domain of tailored dashboards because user preferences could be inferred from behavioral data, context, or any other factor.

Only a few works have applied or mentioned AI when presenting their dashboard solutions. In [32], the Apriori algorithm [44] is used to compute association rules, which is a technique from the data mining field. This solution takes advantage of "pairs of events that have happened in sequence" that fed the Apriori algorithm to obtain a set of if-then rules that will be used to restructure the dashboard in terms of the presented data and visualization types employed. In a study referencing those mentioned above [45], the same authors specify that their solution also supports the restructuration of the dashboards through other methods, like Markov chains or top-k queries, but they don't detail these processes.

Also, in [35], a semantic EYE reasoner is employed to discover potentially interesting data compositions through a knowledge base and semantically annotated visualization and data services. The use of semantic reasoners allows the inference of consequences from facts, enabling in this case, ''the detection of complex events that previously would have remained undetected'' [35]. However, no details about the implementation of the reasoner are addressed in this work.

Other papers mention the possibility of introducing AI techniques, like [24], to rate the generated dashboards through classification algorithms, but authors state that is out of the scope of the paper and refer to [46] as an inspiration. There is also a work that mentions inference [33] to provide a suitable dashboard given the context, user description, and analysis scenario, although no further details are given nor the inference method named

F. HOW MATURE ARE TAILORED DASHBOARDS REGARDING THEIR EVALUATION?

The proposed dashboard solutions are functional regarding their tailoring capabilities, but the maturity of these dashboards regarding their usability is essential to demonstrate if tailoring the dashboards is beneficial for the final users.

In [18], focus groups, pre- and post-study interviews were employed to test the perceived usability and impact of the customizable dashboards. Five experts were in charge of testing the prototype, and subsequently, 13 participants tested the final prototype. Issues regarding the solution involved the configuration process, the interface, and the diversity of available widgets to include in each dashboard, meaning that users needed more components to satisfy their concrete requirements.

In the case of [19], a combined survey and interview was performed to obtain requirements regarding micro-service monitoring. A total of 15 participants were involved, and the gathered information was used to define the main requirements of the customizable dashboard, but no further usability testing was performed regarding the finally implemented dashboard solution.

In [37], it is mentioned that ''first evaluations with users from the domain have already shown that this solution could successfully address the problem of information overload,'' but no details about the evaluation methods nor detailed results are given.

The pre-defined templates-based solution presented in [20] was evaluated during 6-9 months in 40 different small and medium enterprises (SMEs) and evaluated the implemented dashboards' capabilities through [47], obtaining good results regarding dashboard layout, design, presentation, alerting, analysis, KPIs, etc. Also, ''25% of SMEs suggested to change some types of graph or chart and changing some layouts and colors'', possibly needing more customizable elements.

The two model-driven solutions regarding semantic approaches [28], [29] also mentioned user testing and claiming that the feedback shown the relevance of the semantic description language to adapt their dashboards easily. On the other hand, in [29], two users were considered for the assessment to test the role-based dashboard generation and provided their KPIs requirements; however, no further details about this evaluations were given.

In [34], the unique evaluation mentioned that addresses user experience shows that it takes less time for a user to find device commands through the custom user interfaces described in that work. However, no details about the evaluation sample nor methodologies are provided.

The dashboard solution focused on novices presented in [38] is complemented with a detailed usability study to validate their approach and to examine how novice users create dashboards. Fifteen users (7 novice users and 8 BI dashboards experts) participated in the study at a usability lab, where they were interviewed, recorded and asked to complete a series of tasks with the dashboard solution following the think-aloud protocol. The findings shown that novices ranked better the dashboard regarding utility, functionality, ease of use, and overall satisfaction, and expressed the intuitiveness of the dashboard. Experts, on the other hand, requested extra functionalities. The authors then provide a series of additional guidelines when designing visualization systems for novice users based on the results, which can be found in [38].

In [39], an insight-based evaluation is employed to test the validity of the presented model. Six participants not skilled in visual data exploration were asked to do an unguided exploration of a dataset. Participants were asked to complete a survey focusing on insight-based metrics [48] complemented with a demographic survey. The study shown that ''non-default, less familiar settings for expressive richness are more likely to lead to incorrect statements.''

The solution described in [26] performed a usability test through the SUS questionnaire [49] and other open questions. Twelve participants were asked to complete a series of tasks using the DDART system (7 users performed the experiment remotely and 5 with the presence of a researcher). The general usability using the SUS scale was 53.93 for the remote group and 54.50 for the assisted group. They also tested the ease of use of the dashboard by analyzing the success ratio, average time, efficiency and average invalid operations ratio when crafting indicators to gain insights about the difficulty of this feature; results shown that the assisted group performed better than the remote group.

Finally, the dashboard presented in [40] was also tested to obtain information about the usefulness of the solution through users' feedback. In this case, more than 60 users tested the solution through half-open scenarios, where users can ask questions and directions are given only under their demand. Satisfaction scores were collected regarding usability, information output, and functionalities like search, detail view of the results, visualizations, and maps; obtaining high percentages of satisfied users.

The solutions presented in [21]–[25], [27], [30]–[33], [35], [36], did not mention any formal testing regarding end-users' perceptions about the dashboard solutions, mentioning these evaluations as future work. Some of these proposed tools were tested in real-world scenarios to prove their applicability or functionalities, but this research question is focused on user perceptions and experiences on the solutions

VI. DISCUSSION

A variety of works have been retrieved regarding tailoring capabilities within the dashboards' context. Through this literature review, it has been possible to answer questions about the capacities and approaches taken to build tailored dashboards through the existing solutions found in the literature.

It is clear that dashboards can be extremely powerful tools if leveraged; they can support decision-making processes, motivate, persuade, and even make data memorable if properly designed [50]. However, as introduced, a large number of potential users and their large number of individual requirements makes ''one size fits all'' approaches only effective for some profiles, primarily because one size does not fit all for tasks involving cognitive processes; users are influenced by their experiences, their biases, their individual preferences, etc., [43]. Different people could see the same dashboard with the same data and reach different insights as they could be driven by different goals. This cannot be overlooked because people could be missing relevant data for their decision-making processes if the dashboard is not correctly configured for them. What also comes into play are users' abilities: a colorful dashboard could be a pleasant, aesthetic and effective dashboard for one person, but could be a nightmare for a person with eye diseases [51], [52].

For all these reasons, dashboards should include mechanisms to allow tailor-made solutions for individual users without requiring large amounts of resources (making a dashboard from scratch for every potential user is not an efficient option). The existing literature has been analyzed to find how these tailoring processes have been addressed before and to understand the current research context in this area.

Accurately, 23 solutions that addressed the necessity of customizing, personalizing, adapting, etc. dashboards were retrieved. The retrieved solutions address this challenge through very different approaches. The first three research questions had the goal of identifying the technical features of the retrieved solutions. These research questions allowed to distinguish between tailored dashboards by classifying them through technical dimensions. While some solutions let users customize their displays manually [18]–[27] or with assistance [37]–[40], other personalized dashboards through implicit requirements like goals, roles, target data, etc., [28]–[31], in some cases letting users customize the personalized display on demand [34]–[36], the remaining adapted the dashboards in real-time based on user behavior [32], [33].

Customizable solutions address individual requirements by directly asking the user to design their own dashboard without requiring programming skills; by either using graphical user interfaces [18], [19], [26], [27], [37]–[40] or high-level configuration files [21]–[25], [28]–[30] that can abstract the technical and complex details of the dashboard implementation. This approach partially delegates the dashboard design and composition responsibility to the users, which lead to a decrease in development time. But these solutions come with a significant disadvantage: users do not always know what is good for them [7], so they can build ineffective dashboard solutions unwillingly.

This issue can be addressed through assisted customization processes, as found in [37]–[40]. These approaches give freedom to the users regarding their dashboards composition, but also help them with design decisions and charts selection through methods like visual mapping. Another approach is to personalize dashboards by extracting dashboard requirements implicitly from the users [28]–[31].

Hybrid solutions also seem to address these caveats by taking into account that users do not always know what the best for them or their goals is, thus requiring to add a degree of personalization that can materialize implicit requirements. But forcing users to stick to a ''personalized'' solution that makes them uncomfortable is counterproductive, so customizability options should be available if a user feels that she needs to change something. Also, it is interesting to take into account an adaptive dimension to consider the users' behavior evolution because what could seem the best configuration at some point of time could become ineffective over time, as users are involved in new experiences that could change their goals and even improve their visualization literacy and knowledge.

However, relevant questions regarding self-adaptive solutions involve how often should be the dashboard updated to display a new configuration or when it is considered that a requirement has evolved enough and, therefore, the dashboard needs a new configuration. Works addressing adaptive solutions have almost no allusions regarding this concern [32], [33]. In [33], authors state that the adaptation is made on-demand on their proof of concept dashboard, but they don't mention this issue. In [32], [45], the authors suggest that the restructuring period is previously set in each restructurer agents' agendas, but they don't identify or test the implications of these restructuration periods. Adaptation time or adaptation triggers are relevant factors to address when using this approach, because continually changing the user interface could annoy users [53] and be counterproductive, despite the potential benefits of the adaptation.

One of the limitations of the retrieved solutions is that they are very specific to the domain to which they were designed for, as exposed in RQ4. An ideal solution would be valid for every data domain and context, but the vast number of varying features these tools can have increased the complexity of creating a suitable dashboard for every domain and context. There are interesting approaches that tackle tailored dashboards from an abstract point of view, proposing generic solutions that are instantiated to fit into concrete requirements. For instance, model-driven development [28]–[30] and domain engineering paradigms [23], [25] aim at extracting commonalities and shared properties from instances of dashboard systems to foster reusability and decrease the development time. Dealing with domain transferability seems to be easier when a generic point of view of the problem has been established, as it can be seen in [23], [25], [28]–[30], where instantiated dashboards can be adapted to fit into other requirements, domains and contexts without modifying high-level, abstract models.

The answer to RQ5 revealed that artificial intelligence (AI) mechanisms are not leveraged, which could be useful to infer user requirements. Only one work exposed the application of a data mining technique to their dashboard proposal. However, although potentially beneficial, AI approaches present challenges, like the gathering process of significant data to build models. It is crucial to implement collecting mechanisms to store user behavior and their implicit/explicit preferences. But before implementing these mechanisms, relevant users' aspects and factors that can influence their user experience must be identified and backed up by previous studies about perception, to avoid an arbitrary selection of factors that could skew the AI models' outcomes.

Regarding this application, solutions based in structured files to describe a dashboard's features, like in [21]–[25], [28]–[30], [33], provide a good basis for implementing AI solutions because AI algorithms are easier to handle if their inputs are already structured. AI models could be useful to detect patterns or clusters of users regarding dashboard configurations [54]. Artificial intelligence is currently identified as a potentially beneficial method to recommend suitable settings of single information visualizations given different factors, like the target data, user behavior, etc., [55], [56]; but dashboards, where several information visualizations can be displayed and can even hold linked views, are not mentioned in these works.

Finally, there is a lack of user testing regarding the developed solutions. As exposed in RQ6, the solutions were tested regarding their functionality, but few works also included testing regarding user experience and insight delivery. Formal testing is essential in this kind of solutions because it can expose the actual usefulness of tailoring capabilities in terms of usability and knowledge generation, thus helping to improve the tailoring mechanisms and enhance decision-making processes.

This systematic literature review exposed a wide range of available approaches to tackle tailored dashboards. Choosing a proper approach depends, of course, on the application context, the audience, the data, and the available resources for the development of tailored dashboards. Experts could demand fine-grained features, but they can build their own dashboards without assistance given their visualization literacy. Novices could require system support for composing a dashboard, or even a personalized display already designed to match their requirements. Some users could request contextual information about the presented data if their knowledge about the data domain is limited. And so on. What is clear after conducting the present SLR is that one size does not fit all when talking about dashboards, but pursuing generic solutions that can be derived to match different contexts might be a proper path to follow.

VII. THREATS TO VALIDITY

This kind of reviews can be influenced by a series of limitations. One of these limitations is the authors' bias regarding the whole data extraction. As exposed in previous sections, quality criteria were employed to reduce the effects of bias in the inclusion phase of the SLR. The three authors were involved in the review planning to identify and avoid any early issues regarding the study design. Moreover, the first author was the primary reviewer, while the last two authors reproduced each SLR phase to ensure the validity of the results taking into account different perspectives.

Also, different resources with the outcomes of each step are provided to make the whole process reproducible.

Although following a systematic, well-defined protocol, it is not guaranteed that all the relevant works about this field are retrieved. Regarding the search medium, the most relevant electronic databases in the field of computer science were included. The exclusion of Google Scholar from this review is justified by the necessity of considering only databases that index quality contrasted contents. Also, to include the maximum quantity of representative terms about the tailored dashboards, synonyms, and related terms were identified, and the results of preliminary search strings were evaluated to analyze if the retrieved data were relevant for the scope of this literature review. Through this iterative process, the query string was refined to ensure useful and precise data extraction.

One of the main limitations regarding the field in which this SLR is framed is that several dashboard solutions are commercial solutions. Thus no public research works regarding their technical features or the methodologies used are available. Despite this issue, enough relevant works for answering the considered research questions were retrieved.

Lastly, to ensure that the whole process is traceable and reproducible, all the materials, partial results, checklists, etc., have been made available through public repositories.

VIII. CONCLUSION

A systematic literature review (SLR) has been conducted to analyze previous works that address tailoring capabilities and mechanisms regarding information dashboards. This SLR addresses relevant aspects regarding these solutions, such as the applied methodologies, dashboard requirements management, domain transferability, artificial intelligence applications, and user experience testing, to identify current issues and challenges, as well as new research paths to enhance tailoring capabilities and consequently, knowledge generation through individualized dashboards.

By performing an SLR, research questions about these dashboard solutions have been answered, providing a comprehensive view of this research field's current state. During the review process, 1034 papers were retrieved from 4 different electronic databases. The number of papers was reduced to 23 after applying an inclusion and exclusion criteria, and a quality assessment to keep only relevant works for the scope of the research. The analysis of the selected papers exposed that tailored dashboards have been tackled through diverse methodologies and mechanisms that enable support for different dashboard configurations without consuming loads of resources and without requiring long development processes as compared with the design and implementation of individual dashboards from scratch.

This SLR provides a foundation in terms of existing approaches for tailoring information dashboards. Information dashboards have become key tools when dealing with data, and there are a lot of challenges regarding their development and design; one of these challenges is their adaptation to different contexts, domains and users [1]. This review can support researchers and developers in choosing a proper mechanism to develop tailored dashboards. Also, the identified challenges can open new research paths. Moreover, the obtained results can be applied to improve dashboard solutions lacking flexibility.

Future work will involve the application of the gained knowledge to propose new tailored dashboard solutions that address the challenges and issues found through this review. Based on the presented analysis, one of the most promising research avenues is the application of AI paradigms to automatize the design and development of dashboards. This application will not only involve the selection and development of tailoring mechanisms but also the study and classification of users to make these automatic tailoring processes useful and effective.

REFERENCES

[1] A. Sarikaya, M. Correll, L. Bartram, M. Tory, and D. Fisher, "What do we talk about when we talk about dashboards?" *IEEE Trans. Vis. Comput. Graph.*, vol. 25, no. 1, pp. 682–692, Jan. 2019.

[2] S. Few, *Information Dashboard Design*. Sebastopol, CA, USA: O'Reilly Media, 2006.

[3] S. Wexler, J. Shaffer, and A. Cotgreave, *The Big Book of Dashboards: Visualizing Your Data Using Real-World Business Scenarios*. Hoboken, NJ, USA: Wiley, 2017.

[4] J. Hullman, E. Adar, and P. Shah, "The impact of social information on visual judgments," in *Proc. ACM SIGCHI Conf. Hum. Factors Comput. Syst.*, 2011, pp. 1461–1470.

[5] Y.-S. Kim, K. Reinecke, and J. Hullman, "Data through others 'eyes: The impact of visualizing others' expectations on visualization interpretation," *IEEE Trans. Vis. Comput. Graphics*, vol. 24, no. 1, pp. 760–769, Jan. 2018.

[6] D. A. Norman and S. W. Draper, *User Centered System Design: New Perspectives on Human-computer Interaction*. Boca Raton, FL, USA: CRC Press, 1986.

[7] L. Padilla. (2018). *How do we Know When a Visualization is Good? Perspectives From a Cognitive Scientist*. [Online]. Available: https://medium.com/multiple-views-visualization-research-explained/how-do-we-know-when-a-visualization-is-good-c894b5194b62

[8] B. Kitchenham, "Procedures for performing systematic reviews," Dept. Comput. Sci., Keele Univ., Keele, U.K., Tech. Rep. 0400011T.1, 2004, pp. 1–26, vol. 33.

[9] B. Kitchenham and S. Charters, "Guidelines for performing systematic literature reviews in software engineering," School Comput. Sci. Math., Keele Univ., Keele, U.K., Tech. Rep. EBSE-2007-01, 2007. [Online]. Available: https://goo.gl/L1VHcw.

[10] B. A. Kitchenham, D. Budgen, and O. P. Brereton, "Using mapping studies as the basis for further research—A participant-observer case study," *Inf. Softw. Technol.*, vol. 53, no. 6, pp. 638–651, 2011.

[11] A. Vázquez-Ingelmo, F. J. García-Peñalvo, and R. Therón, "Tailored information dashboards: A systematic mapping of the literature," presented at the Interacción, Donostia, Spain, 2019.

[12] B. Napoleão, K. R. Felizardo, É. F. de Souza, and N. L. Vijaykumar, "Practical similarities and differences between systematic literature reviews and systematic mappings: A tertiary study," in *Proc. SEKE*, 2017, pp. 85–90.

[13] M. Petticrew and H. Roberts, *Systematic Reviews in the Social Sciences: A Practical Guide*. Hoboken, NJ, USA: Wiley, 2008.

[14] M. Cutts, *Oxford Guide to Plain English*. Oxford, U.K.: OUP Oxford, 2013.

[15] R. R. Prasath, "Learning age and gender using co-occurrence of non-dictionary words from stylistic variations," in *Proc. Int. Conf. Rough Sets Current Trends Comput.* Berlin, Germany: Springer, 2010, pp. 544–550.

[16] D. Moher, A. Liberati, J. Tetzlaff, D. G. Altman, and T. P. Group, "Preferred reporting items for systematic reviews and meta-analyses: The PRISMA statement," *PLoS Med.*, vol. 6, no. 7, 2009, Art. no. e1000097.

[17] A. V. Ingelmo. (2019). *Resources for the Paper, Information Dashboards and Tailoring Capabilities—A Systematic Literature Review*. [Online]. Available: https://zenodo.org/record/3246043-.XTl15ZMzZ24

[18] D. Filonik, R. Medland, M. Foth, and M. Rittenbruch, "A customisable dashboard display for environmental performance visualisations," in *Persuasive Technology*. Berlin, Germany: Springer, 2013, pp. 51–62.

[19] B. Mayer and R. Weinreich, "A dashboard for microservice monitoring and management," in *Proc. IEEE Int. Conf. Softw. Archit. Workshops (ICSAW)*, Gothenburg, Sweden, Apr. 2017, pp. 66–69.

[20] W. Noonpakdee, T. Khunkornsiri, A. Phothichai, and K. Danaisawat, "A framework for analyzing and developing dashboard templates for small and medium enterprises," in *Proc. IEEE 5th Int. Conf. Ind. Eng. Appl. (ICIEA)*, Apr. 2018, pp. 479–483.

[21] K. Kumar, J. Bose, and S. K. Soni, "A generic visualization framework based on a data driven approach for the analytics data," in *Proc. 14th IEEE India Council Int. Conf. (INDICON)*, Dec. 2017, pp. 1–6.

[22] A. Cardoso, C. J. V. Teixeira, and J. S. Pinto, "Architecture for highly configurable dashboards for operations monitoring and support," *Stud. Inform. Control*, vol. 27, no. 3, pp. 319–330, 2018.

[23] A. Vázquez-Ingelmo, F. J. García-Peñalvo, and R. Therón, "Domain engineering for generating dashboards to analyze employment and employability in the academic context," presented at the 6th Int. Conf. Technol. Ecosyst. Enhancing Multiculturality, Salamanca, Spain, 2018.

[24] O. Pastushenko, J. Hynek, and T. Hruška, "Generation of test samples for construction of dashboard design guidelines: Impact of color on layout balance," in *Proc. World Conf. Inf. Syst. Technol.* Cham, Switzerland: Springer, 2018, pp. 980–990.

[25] I. Logre, S. Mosser, P. Collet, and M. Riveill, "Sensor data visualisation: A composition-based approach to support domain variability," in *Proc. Eur. Conf. Model. Found. Appl.* Cham, Switzerland: Springer, 2014, pp. 101–116.

[26] C. Michel, E. Lavoué, S. George, and M. Ji, "Supporting awareness and self-regulation in project-based learning through personalized dashboards," *Int. J. Technol. Enhanced Learn.*, vol. 9, nos. 2–3, pp. 204–226, 2017.

[27] G. L. Miotto, L. Magnoni, and J. E. Sloper, "The TDAQ Analytics Dashboard: A real-time Web application for the ATLAS TDAQ control infrastructure," *J. Phys., Conf. Ser.*, vol. 331, no. 2, 2011, Art. no. 022019.

[28] M. Kintz, "A semantic dashboard description language for a process-oriented dashboard design methodology," in *Proc. 2nd Int. Workshop Model-Based Interact. Ubiquitous Syst. (MODIQUITOUS)*, Copenhagen, Denmark, vol. 947, 2012, pp. 31–36.

[29] M. Kintz, M. Kochanowski, and F. Koetter, "Creating user-specific business process monitoring dashboards with a model-driven approach," in *Proc. MODELSWARD*, 2017, pp. 353–361.

[30] T. Palpanas, P. Chowdhary, G. Mihaila, and F. Pinel, "Integrated model-driven dashboard development," *Inf. Syst. Frontiers*, vol. 9, nos. 2–3, pp. 195–208, 2007.

[31] S. Arjun, "Personalizing data visualization and interaction," in *Proc. ACM Adjunct Publication 26th Conf. User Modeling, Adaptation Personalization*, 2018, pp. 199–202.
[32] O. Belo, P. Rodrigues, R. Barros, and H. Correia, "Restructuring dynamically analytical dashboards based on usage profiles," in *Proc. Int. Symp. Methodol. Intell. Syst.* Cham, Switzerland: Springer, 2014, pp. 445–455.
[33] I. Dabbebi, S. Iksal, J.-M. Gilliot, M. May, and S. Garlatti, "Towards adaptive dashboards for learning analytic: An approach for conceptual design and implementation," in *Proc. 9th Int. Conf. Comput. Supported Edu. (CSEDU)*, Porto, Portugal, vol. 1, 2017, pp. 120–131.
[34] S. Radovanović, B. Majstorović, S. Kukolj, and M. Z. Bjelica, "Device cloud platform with customizable remote user interfaces," in *Proc. IEEE 4th Int. Conf. Consum. Electron. Berlin (ICCE-Berlin)*, Sep. 2014, pp. 202–204.
[35] S. Van Hoecke, C. Huys, O. Janssens, R. Verborgh, and R. Van de Walle, "Dynamic monitoring dashboards through composition of Web and visualization services," in *Internet of Things. IoT Infrastructures*. Cham, Switzerland: Springer, 2015, pp. 465–474.
[36] H. Santos, V. Dantas, V. Furtado, P. Pinheiro, and D. L. McGuinness, "From data to city indicators: A knowledge graph for supporting automatic generation of dashboards," in *Proc. Eur. Semantic Web Conf.* Cham, Switzerland: Springer, 2017, pp. 94–108.
[37] B. S. Nascimento, A. S. Vivacqua, and M. R. S. Borges, "A flexible architecture for selection and visualization of information in emergency situations," in *Proc. IEEE Int. Conf. Syst., Man, Cybern. (SMC)*, Oct. 2016, pp. 3317–3322.
[38] M. Elias and A. Bezerianos, "Exploration views: Understanding dashboard creation and customization for visualization novices," in *Proc. IFIP Conf. Hum.-Comput. Interact.* Berlin, Germany: Springer, 2011, pp. 274–291.
[39] M. A. Yalçın, N. Elmqvist, and B. B. Bederson, "Keshif: Rapid and expressive tabular data exploration for novices," *IEEE Trans. Vis. Comput. Graphics*, vol. 24, no. 8, pp. 2339–2352, Aug. 2018.
[40] G. Petasis, A. Triantafillou, and E. Karstens, "YourDataStories: Transparency and corruption fighting through data interlinking and visual exploration," in *Proc. Int. Conf. Internet Sci.* Cham, Switzerland: Springer, 2017, pp. 95–108.
[41] E. H.-H. Chi, "A taxonomy of visualization techniques using the data state reference model," in *Proc. IEEE Symp. Inf. Vis. (INFOVIS)*, Oct. 2000, pp. 69–75.
[42] A. Vázquez-Ingelmo, F. J. García-Peñalvo, and R. Therón, "Taking advantage of the software product line paradigm to generate customized user interfaces for decision-making processes: A case study on university employability," *PeerJ Comput. Sci.*, vol. 5, p. e203, Jul. 2019.
[43] S. D. Teasley, "Student facing dashboards: One size fits all?" *Technol., Knowl. Learn.*, vol. 22, no. 3, pp. 377–384, 2017.
[44] R. Agrawal and R. Srikant, "Fast algorithms for mining association rules," in *Proc. 20th Int. Conf. Very Large Data Bases, VLDB*, vol. 1215, 1994, pp. 487–499.
[45] O. Belo, H. Correia, P. Rodrigues, and R. Barros, "A personalization system for data visualization platforms," in *Proc. IEEE 6th Int. Conf. Innov. Comput. Technol. (INTECH)*, Aug. 2016, pp. 162–167.
[46] K. Reinecke, T. Yeh, L. Miratrix, R. Mardiko, Y. Zhao, J. Liu, and K. Z. Gajos, "Predicting users' first impressions of website aesthetics with a quantification of perceived visual complexity and colorfulness," in *Proc. ACM SIGCHI Conf. Hum. Factors Comput. Syst.*, 2013, pp. 2049–2058.
[47] M. Dyczkowski, J. Korczak, and H. Dudycz, "Multi-criteria evaluation of the intelligent dashboard for SME managers based on scorecard framework," in *Proc. IEEE Federated Conf. Comput. Sci. Inf. Syst.*, Sep. 2014, pp. 1147–1155.
[48] P. Saraiya, C. North, and K. Duca, "An insight-based methodology for evaluating bioinformatics visualizations," *IEEE Trans. Vis. Comput. Graphics*, vol. 11, no. 4, pp. 443–456, Jul. 2005.
[49] J. Sauro, *A Practical Guide to the System Usability Scale: Background, Benchmarks & Best Practices*. Scotts Valley, CA, USA: Createspace Independent, 2011.
[50] S. Berinato, *Good Charts: The HBR Guide to Making Smarter, More Persuasive Data Visualizations*. Brighton, MA, USA: Harvard Business Review, 2016.
[51] S. Silva, B. S. Santos, and J. Madeira, "Using color in visualization: A survey," *Comput. Graph.*, vol. 35, no. 2, pp. 320–333, 2011.
[52] T. Zuk, L. Schlesier, P. Neumann, M. S. Hancock, and S. Carpendale, "Heuristics for information visualization evaluation," in *Proc. ACM AVI Workshop BEyond Time Errors, Novel Eval. Methods Inf. Vis.*, 2006, pp. 1–6.
[53] M. Zavyiboroda. (2018). *Why User Interface Changes Annoy People & How to Handle it*. [Online]. Available: https://speckyboy.com/annoying-ui-changes/
[54] J. Cruz-Benito, A. Vázquez-Ingelmo, J. C. Sánchez-Prieto, R. Therón, F. J. García-Peñalvo, and M. Martín-González, "Enabling adaptability in Web forms based on user characteristics detection through A/B testing and machine learning," *IEEE Access*, vol. 6, pp. 2251–2265, 2018.
[55] M. Vartak, S. Huang, T. Siddiqui, S. Madden, and A. Parameswaran, "Towards visualization recommendation systems," *ACM SIGMOD Rec.*, vol. 45, no. 4, pp. 34–39, 2017.
[56] P. Kaur and M. Owonibi, "A review on visualization recommendation strategies," in *Proc. 12th Int. Joint Conf. Comput. Vis., Imag. Comput. Graph. Theory Appl. (VISIGRAPP)*, vol. 3, 2017, pp. 266–273.

ANDREA VÁZQUEZ-INGELMO received the bachelor's and master's degrees in computer engineering from the University of Salamanca, Salamanca, in 2016 and 2018, respectively. She is currently pursuing the Ph.D. degree in computer sciences with the Research Group of Interaction and eLearning (GRIAL), where she is also a member. Her research interests include related to human–computer interaction, software engineering, information visualization, and machine learning applications.

FRANCISCO J. GARCÍA-PEÑALVO received the degrees in computing from the University of Salamanca and the University of Valladolid, and the Ph.D. degree from the University of Salamanca. He was the Vice Chancellor of innovation with the University of Salamanca, from 2007 to 2009. He is currently the Head of the Research Group Interaction and eLearning and a Full Professor with the Computer Science Department, University of Salamanca. He has led and participated in over 50 research and innovation projects. He has authored over 300 articles in international journals and conferences. His main research interests include eLearning, computers and education, adaptive systems, web engineering, semantic web, and software reuse. He is also a member of the Program Committee of several international conferences and a Reviewer of several international journals. He was a Guest Editor of several special issues of international journals, such as the *Online Information Review*, the *Computers in Human Behavior*, and the *Interactive Learning Environments*. He is currently the Editor-in-Chief of the *International Journal of Information Technology Research* and the *Education in the Knowledge Society Journal*. Besides, he is the Coordinator of the multidisciplinary Ph.D. Programme on Education in the Knowledge Society.

ROBERTO THERÓN received the Diploma degree in computer science from the University of Salamanca, the B.A. degree from the Universidade da Coruña, the bachelor's degree in communication studies and humanities from the University of Salamanca, and the Ph.D. degree from the Research Group Robotics, University of Salamanca. His Ph.D. thesis was on parallel calculation configuration space for redundant robots. He is currently the Manager of the VisUSAL Group (within the Recognized Research Group GRIAL), University of Salamanca, which focuses on the combination of approaches from computer science, statistics, graphic design, and information visualization to obtain an adequate understanding of complex data sets. He has authored over 100 articles in international journals and conferences. In recent years, he has been involved in developing advanced visualization tools for multidimensional data, such as genetics or paleo-climate data. In the field of visual analytics, he develops productive collaborations with groups and institutions internationally recognized as the Laboratory of Climate Sciences and the Environment, France, or the Austrian Academy of Sciences, Austria. He received the Extraordinary Doctoral Award for his Ph.D. thesis.

7.9 Appendix I. Automatic generation of software interfaces for supporting decision-making processes. An application of domain engineering and machine learning

A. Vázquez-Ingelmo, F. J. García-Peñalvo and R. Therón, "Automatic generation of software interfaces for supporting decision-making processes. An application of domain engineering and machine learning," in TEEM'19 Proceedings of the Seventh International Conference on Technological Ecosystems for Enhancing Multiculturality (Leon, Spain, October 16th-18th, 2019), M. Á. Conde-González, F. J. Rodríguez-Sedano, C. Fernández-Llamas and F. J. García-Peñalvo, Eds. ICPS: ACM International Conference Proceedings Series, pp. 1007-1011, New York, NY, USA: ACM, 2019. doi: 10.1145/3362789.3362923.

Automatic generation of software interfaces for supporting decision-making processes. An application of domain engineering and machine learning.

Andrea Vázquez-Ingelmo
GRIAL Research Group, Department of Computer Science, Research Institute for Educational Sciences. University of Salamanca
Paseo de Canalejas 169, 37008 Salamanca, Spain
andreavazquez@usal.es

Francisco J. García-Peñalvo
GRIAL Research Group, Department of Computer Science, Research Institute for Educational Sciences. University of Salamanca
Paseo de Canalejas 169, 37008 Salamanca, Spain
fgarcia@usal.es

Roberto Therón
GRIAL Research Group, Department of Computer Science, Research Institute for Educational Sciences. University of Salamanca
Paseo de Canalejas 169, 37008 Salamanca, Spain
theron@usal.es

ABSTRACT

Information dashboards are sophisticated tools. Although they enable users to reach useful insights and support their decision-making challenges, a good design process is essential to obtain powerful tools. Users need to be part of these design processes, as they will be the consumers of the information displayed. But users are very diverse and can have different goals, beliefs, preferences, etc., and creating a new dashboard for each potential user is not viable. There exist several tools that allow users to configure their displays without requiring programming skills. However, users might not exactly know what they want to visualize or explore, also becoming the configuration process a tedious task. This research project aims to explore the automatic generation of user interfaces for supporting these decision-making processes. To tackle these challenges, a domain engineering, and machine learning approach is taken. The main goal is to automatize the design process of dashboards by learning from the context, including the end-users and the target data to be displayed.

KEYWORDS

Automatic generation; Domain engineering; Meta-modeling; Information Dashboards; High-level requirements.

1 Introduction

The introduction of information systems to support and automatize the great diversity of activities has led an increment regarding the volume of generated data. However, the possession of significant quantities of data is not valuable if they are not analyzed and leveraged to create knowledge [1].

Data analysis has gained relevance over the years in different sectors [2] with the vision to accomplish robust information bases to support strategic decision-making processes [3].

Different people and profiles can participate during the process of making decisions, especially in interdisciplinary contexts. Given this situation, communication gaps could raise among the existing profiles, as they have different knowledge levels regarding different domains [4].

Each person involved must understand which implications the collected and analyzed data have to get the most out of decision-making processes. However, the gathered variables that might be relevant for supporting decisions could be complex, and users could need assistance to explore patterns and relationships among them.

One of the most powerful software tools for generating knowledge and exploiting datasets are information dashboards [5, 6]. Information dashboards are displays composed of graphical resources [7] and metrics to present information understandably [8], allowing pattern recognition or relevant indicators for decision-making processes.

However, designing a dashboard is not a trivial task [6]; it is necessary to take into account the users' necessities to assist them in reaching their goals. But this search process of needs process is usually complex, given the impossibility of knowing right from the beginning the exact kind of visualizations or metrics that would be beneficial for each individual user.

Current studies have identified different challenges regarding the design process of these tools, like functional flexibility, the influence of a user's knowledge or literacy about certain domains, the social impact, etc. [5]. Indeed, it is necessary to bear in mind that knowledge generation processes can vary depending on the person. Also, each person's goals regarding the same data can be completely different. However, building individual dashboards from scratch for every potentially involved profile would be unfeasible, requiring great quantities of time and resources.

The main focus of this research project is to tackle the automatic generation of personalized dashboards to raise the effectiveness of these tools as well as the productivity regarding their development, establishing relations among the users' concrete goals, preferences, abilities, etc. with the set of potential features and elements that would finally compose the personalized dashboard.

As it will be detailed, two paradigms are considered to address this challenge: the software product line (SPL) paradigm [9] and the model-driven engineering approach [10]. These two approaches are a good strategy to generate flexible and maintainable dashboards with different features.

On the other hand, current artificial intelligence (AI) methods can be leveraged and applied to the dashboards generation process. An AI model can be fed with users' characteristics [11] to offer the potentially best and most beneficial dashboard configurations for their contexts.

The rest of the paper is organized through the following sections: the second section introduces the hypotheses and objectives of this research project, followed by section 3, where the used methodology is detailed. Section 4 presents the current status of the research, finalizing with section 5, where the conclusions derived from this work are presented.

2 Hypotheses and objectives

The main hypothesis of this work is the following:

H1. The tailoring of user interfaces for supporting decision-making processes increment the efficiency and efficacy when extracting information and generating knowledge from the displayed data.

The goal of the research is to obtain a generative framework for the automatic and systematic development of information dashboards, where the tailoring task involves not only variability at the layout and visual design levels, but also at data and interaction mechanisms levels to foster individualization, usability, and flexibility. This goal also involves the application of AI technology to provide beneficial dashboard configurations automatically.

A series of sub-objectives are posed to reach the mentioned main goal.

- Identify common characteristics of information dashboards at a meta-level (i.e., abstract level)
- Identify connection mechanisms to enable a model-driven approach to build concrete products of the SPL
- Implement mechanisms that foster interoperability to allow the connection of different data sources
- Define and implement reusable and configurable core assets to generate concrete products of the SPL
- Evaluate the SPL at a generative and functional level
- Evaluate the generated dashboards in terms of usability and tailoring capabilities
- Study the automatic adaptation of the dashboards depending on the users' characteristics an behavior through AI mechanisms
- Study the integration of the dashboards SPL within different technological ecosystems and case studies

The outcomes of the different phases of the research will support the test of posed hypothesis.

3 Methodology

3.1 Action-research methodology

This research project is approached as an iterative process where the knowledge gained through past experiences and the outcomes of the different cycles is crucial for the following stages. The Action-Research methodological framework [12] will be followed to accomplish this process.

Kemmis posed Action-Research [13] as an inquiry method carried out by the participants in social situations with the aim of improving and understanding their own social practices and their contexts.

Later, McTaggart & Kemmis described the characteristics of this methodology. The Action-Research methodology is based on a cyclic spiral of research and actions composed of a series of phases and sequences [14].

Therefore, Action-Research is an iterative process where each cycle provides an output that will be the input for the next cycle.

The iterative nature of the methodology enables the researcher to address previously identified problems, thus obtaining more refined solutions.

However, to be able to start the Action-Research cycles, it is necessary to formalize the problem to be addressed. Similar problems and previously developed solutions have been studied to understand the context and the current state of the field. The methodology used for this step (a Systematic Literature Review) is detailed in the next section.

Once the problem is formalized, two Action-Research cycles are proposed to develop a proposal for generating dashboards and evaluate them in real contexts. Evaluation is necessary to obtain feedback to improve the proposal.

The chosen framework for software development is an agile approach based on SCRUM [15]. This framework provides the necessary processes, rules, practices, roles, and artifacts to increase the productivity of development teams through an iterative and incremental software development cycle [16].

A mixed methods research approach will be employed to evaluate the dashboards. The research will be conducted using both quantitative and qualitative methods [17], leveraging the two perspectives to obtain a wider view of the results to face the next Action-Research cycles.

3.2 Systematic literature review

As introduced above, a systematic literature review (SLR) is a powerful method to gain knowledge about previous solutions and similar problems. The SLR helps in the contextualization of the problem to be solved and provides new research lines by identifying weaknesses and strengths in previous solutions.

The SLR is conducted under the guidelines proposed by Kitchenham [18]. Following the [18, 19] guidelines, the SLR is composed of three main phases: planning, conducting, and reporting the study.

However, before planning the review, a preliminary search was performed to verify that no recent reviews about the target topic. If any recent SLR were found, there would not be any necessity to conduct a new one.

This preliminary search was performed using different electronic databases (Scopus, Web of Science (WoS), IEEE Xplorer and Springer) and using terms related to literature reviews ("SLR", "systematic literature review", etc.), as well as terms related to the target of the review ("dashboards").

The result of the previous search confirmed that, at the time of performing the queries, there were not any previous SLR about tailored dashboards, so the necessity of performing a literature review was justified.

3.3 Meta-modeling and domain engineering

Two methods are selected to tackle the design of tailored dashboards with automation possibilities.

Given the complexity of the dashboards' design processes, it is necessary to understand their domain deeply. Dashboards can present different features, different visual designs, different purposes, etc. However, dashboards also share common features that are always present.

These common features can be abstracted to obtain generic schemas or models that can help with the domain understanding and systematic reuse of software components. The technique for identifying shared properties and variabilities within a specific domain is called domain engineering [20].

Domain engineering is based on knowledge reuse regarding some specific domain. This approach is an essential phase of the software product line (SPL) paradigm [21, 22]. This methodology allows the reuse of software components and their configuration to match certain requirements; that is why identifying common features and variabilities is an essential step.

Once the domain has been studied, it is possible to develop a generic model (a meta-model) that captures every abstract property of dashboards, as well as the relationships among the identified entities.

Meta-models are crucial artifacts in model-driven paradigms [10, 23, 24], as they allow mapping entities from high-abstraction levels to more concrete entities and even code through transformations.

These two related methodologies increase not only productivity regarding software development, but also knowledge reuse, and are suitable methods to address several requirements from different profiles and contexts.

4 Dissertation status

The presented research project is currently in a conceptualization stage.

As previously introduced, the dashboards domain is a complex field of study, as several elements and disciplines are involved. That is why the contextualization and research of previous solutions are crucial for the development of the thesis.

A systematic literature review (SLR) has been performed to gain knowledge about this domain, as detailed before. The SLR is focused on how existing approaches and solutions have tackled tailoring capabilities of dashboards.

During the planning phase, the scope of the review was defined: research questions, inclusion and exclusion criteria, search strategy, query strings, and quality criteria. The research questions that the systematic review aims to answer are the following:

- **RQ1.** How have existing dashboard solutions tackled the necessity of tailoring capabilities?
- **RQ2.** Which methods have been applied to support tailoring capabilities within the dashboards' domain?
- **RQ3.** How the proposed solutions manage the dashboard's requirements?
- **RQ4.** Can the proposed solutions be transferred to different domains?
- **RQ5.** Has any artificial intelligence approach been applied to the dashboards' tailoring processes and, if applicable, how these approaches have been involved in the dashboards' tailoring processes?
- **RQ6.** How mature are tailored dashboards regarding their evaluation?

The data extraction process to conduct the present SLR has been divided into different phases in which various activities are performed. The PRISMA statement [25] has been used to detail the performed tasks during the whole review process, and it can be consulted in Figure 1.

In the end, 23 papers about tailoring capabilities within the dashboards domain were selected, and the different research questions were answered through them, obtaining a wide-view of how this tailoring challenge has been addressed before.

The SLR [26] was complemented with a systematic mapping of the literature that can be consulted at [27]. The gained knowledge about technical features (RQ1-RQ4), AI applications (RQ5), and evaluation (RQ6) will be applied to improve tailoring capabilities. Especially, the answer to RQ5 provided clarity on the lack of AI applications on this domain, which could be potentially useful to automate the configuration processes of dashboards to match concrete user requirements and contexts.

A dashboard meta-model has been developed using the gained knowledge and employing the methodologies described in section 3.3. The meta-model is composed of three main sections: user, layout, and components.

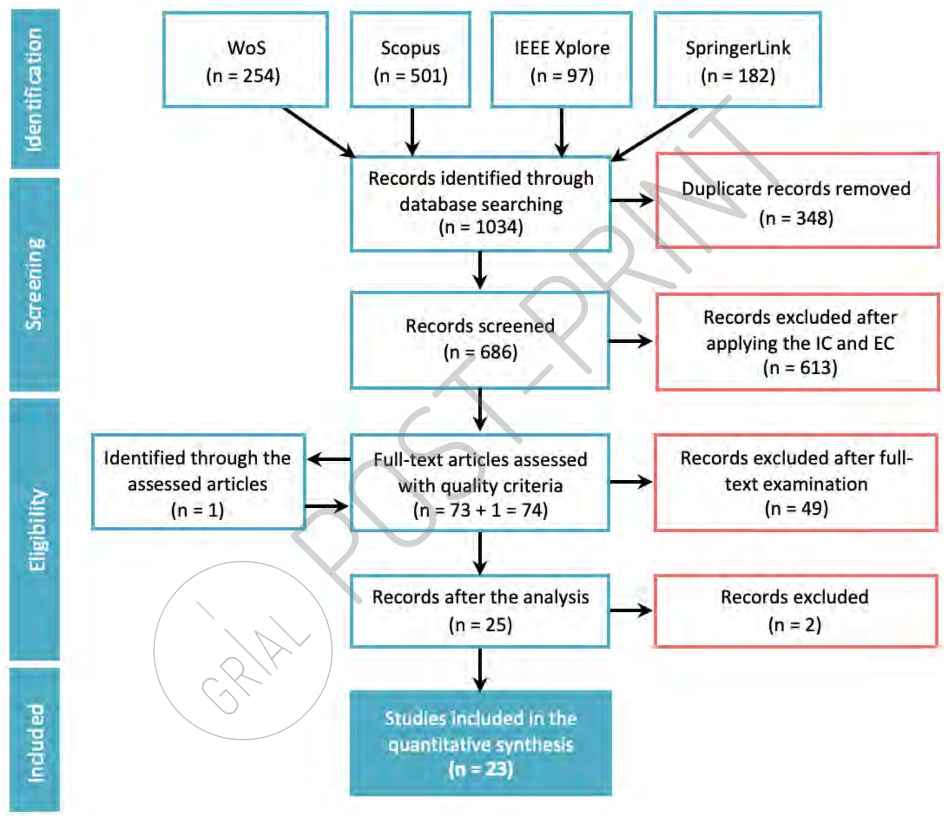


Figure 1: PRISMA flow. Adapted from [25]. Elaborated by the authors [27].

Each section is related to each other, and the whole meta-model describes a high-level view of dashboards with their common features and properties.

The user and layout meta-models can be consulted at [28, 29]. The next steps will involve the refinement (addition of constraints, rules, etc.) and instantiation of this meta-model to obtain concrete models and mapping them to real code through AI paradigms.

5 Conclusions

The proposed research project is focused on developing tailored solutions to support decision-making processes. This work outlines the methodologies and current status of the dissertation.

A systematic literature review has been performed to gain knowledge about the domain, and a dashboard meta-model has been developed to capture high-level features and properties.

The next steps will employ the meta-model to instantiate concrete dashboard models to generate their code subsequently.

The automatic generation of dashboards could improve the effectiveness of these tools by adapting them to specific users' needs without consuming significant quantities of resources and time.

ACKNOWLEDGMENTS

This research work has been supported by the Spanish *Ministry of Education and Vocational Training* under an FPU fellowship (FPU17/03276). This work has been partially funded by the Spanish Government Ministry of Economy and Competitiveness throughout the DEFINES project (Ref. TIN2016-80172-R) and the Ministry of Education of the Junta de Castilla y León (Spain) throughout the T-CUIDA project (Ref. SA061P17).

REFERENCES

[1] S Christian Albright, Wayne Winston, and Christopher Zappe. 2010. Data analysis and decision making. Cengage Learning, Mason, OH, USA.
[2] Dj Patil and Hilary Mason. 2015. Data Driven. " O'Reilly Media, Inc.".
[3] Ramesh Sharda, Dursun Delen, and Efraim Turban. 2013. Business intelligence: a managerial perspective on analytics. Prentice Hall Press.
[4] Martin J Eppler. 2007. Knowledge communication problems between experts and decision makers: An overview and classification. Electronic Journal of Knowledge Management 5, 3.
[5] Alper Sarikaya, Michael Correll, Lyn Bartram, Melanie Tory, and Danyel Fisher. 2018. What Do We Talk About When We Talk About Dashboards? IEEE Transactions on Visualization Computer Graphics 25, 1, 682 - 692.
[6] Stephen Few. 2006. Information dashboard design.
[7] Daniel Keim, Gennady Andrienko, Jean-Daniel Fekete, Carsten Görg, Jörn Kohlhammer, and Guy Melançon. 2008. Visual analytics: Definition, process, and challenges. In Information Visualization Springer, 154-175.
[8] Edward Tufte and P Graves-Morris. 2014. The visual display of quantitative information.; 1983 Graphics Press, Cheshire, CT, USA.
[9] Klaus Pohl, Gunter Böckle, and Frank J. Van Der Linden. 2005. Software Product Line Engineering: Foundations, Principles and Techniques. Springer-Verlag New York, Inc., New York, NY, USA.
[10] Anneke G Kleppe, Jos Warmer, and Wim Bast. 2003. MDA Explained. The Model Driven Architecture: Practice and Promise Addison-Wesley Longman Publishing Co., Inc., Boston, MA.
[11] J. Cruz-Benito, A. Vázquez-Ingelmo, J. C. Sánchez-Prieto, R. Therón, F. J. García-Peñalvo, and M. Martín-González. 2018. Enabling adaptability in web forms based on user characteristics detection through A/B testing and machine learning. IEEE Access 6, 2251-2265. DOI:http://dx.doi.org/10.1109/ACCESS.2017.2782678.
[12] Kurt Lewin. 1946. Action research and minority problems. Journal of social issues 2, 4, 34-46.
[13] Stephen Kemmis. 1984. Point-by-point guide to action research. Victoria: Deakin University.
[14] Robin Mctaggart and Stephen Kemmis. 1988. The action research planner. Deakin university.
[15] Ken Schwaber. 1997. Scrum development process. In Business object design and implementation Springer, 117-134.
[16] Ken Schwaber and Mike Beedle. 2002. Agile software development with Scrum. Prentice Hall Upper Saddle River.
[17] John W Creswell and J David Creswell. 2017. Research design: Qualitative, quantitative, and mixed methods approaches. Sage publications.
[18] B. Kitchenham and S. Charters. 2007. Guidelines for performing Systematic Literature Reviews in Software Engineering. Version 2.3. Technical Report Report. School of Computer Science and Mathematics, Keele University.
[19] Barbara Kitchenham. 2004. Procedures for performing systematic reviews. Keele, UK, Keele University 33, 2004, 1-26.
[20] Maarit Harsu. 2002. A survey on domain engineering. Citeseer.
[21] Markus Voelter and Eelco Visser. 2011. Product line engineering using domain-specific languages. In Software Product Line Conference (SPLC), 2011 15th International IEEE, 70-79.
[22] Paul Clements and Linda Northrop. 2002. Software product lines. Addison-Wesley, Boston, MA, USA.
[23] Nicolas Anquetil, Birgit Grammel, Ismênia Galvão, Joost Noppen, Safoora Shakil Khan, Hugo Arboleda, Awais Rashid, and Alessandro Garcia. 2008. Traceability for model driven, software product line engineering. In ECMDA Traceability Workshop Proceedings SINTEF, 77-86.
[24] Salvador Trujillo, Don Batory, and Oscar Diaz. 2007. Feature oriented model driven development: A case study for portlets. In Proceedings of the 29th international conference on Software Engineering IEEE Computer Society, 44-53.
[25] David Moher, Alessandro Liberati, Jennifer Tetzlaff, Douglas G Altman, and The Prisma Group. 2009. Preferred reporting items for systematic reviews and meta-analyses: the PRISMA statement. PLoS medicine 6, 7, e1000097.
[26] A. Vázquez-Ingelmo, F. J. Garcia-Peñalvo, and R. Therón. 2019. Information Dashboards and Tailoring Capabilities - A Systematic Literature Review. IEEE Access 7, 109673-109688. DOI:http://dx.doi.org/10.1109/ACCESS.2019.2933472.
[27] Andrea Vázquez Ingelmo, Francisco J. García-Peñalvo, and Roberto Therón. 2019. Tailored information dashboards: A systematic mapping of the literature. In Proceedings of the Interacción2019 (Donostia, Spain2019). ACM.
[28] Andrea Vázquez Ingelmo, Francisco J. García-Peñalvo, Roberto Therón, and Miguel Ángel Conde González. 2019. Extending a dashboard meta-model to account for users' characteristics and goals for enhancing personalization. In Proceedings of the Learning Analytics Summer Institute (LASI) Spain 2019 (Vigo, Spain2019).
[29] Andrea Vázquez-Ingelmo, Francisco J. García-Peñalvo, and Roberto Therón. 2019. Taking advantage of the software product line paradigm to generate customized user interfaces for decision-making processes: a case study on university employability. PeerJ Computer Science 5(2019/07/01), e203. DOI:http://dx.doi.org/10.7717/peerj-cs.203.

7.10 Appendix J. Capturing high-level requirements of information dashboards components through meta-modeling

Capturing high-level requirements of information dashboards' components through meta-modeling

Andrea Vázquez-Ingelmo
GRIAL Research Group, Department of Computer Science, Research Institute for Educational Sciences. University of Salamanca
Paseo de Canalejas 169, 37008 Salamanca, Spain
andreavazquez@usal.es

Francisco J. García-Peñalvo
GRIAL Research Group, Department of Computer Science, Research Institute for Educational Sciences. University of Salamanca
Paseo de Canalejas 169, 37008 Salamanca, Spain
fgarcia@usal.es

Roberto Therón
GRIAL Research Group, Department of Computer Science, Research Institute for Educational Sciences. University of Salamanca
Paseo de Canalejas 169, 37008 Salamanca, Spain
theron@usal.es

ABSTRACT

Information dashboards are increasing their sophistication to match new necessities and adapt to the high quantities of generated data nowadays. These tools support visual analysis, knowledge generation, and thus, are crucial systems to assist decision-making processes. However, the design and development processes are complex, because several perspectives and components can be involved. Tailoring capabilities are focused on providing individualized dashboards without affecting the time-to-market through the decrease of the development processes' time. Among the methods used to configure these tools, the software product lines paradigm and model-driven development can be found. These paradigms benefit from the study of the target domain and the abstraction of features, obtaining high-level models that can be instantiated into concrete models. This paper presents a dashboard meta-model that aims to be applicable to any dashboard. Through domain engineering, different features of these tools are identified and arranged into abstract structures and relationships to gain a better understanding of the domain. The goal of the meta-model is to obtain a framework for instantiating any dashboard to adapt them to different contexts and user profiles. One of the contexts in which dashboards are gaining relevance is Learning Analytics, as learning dashboards are powerful tools for assisting teachers and students in their learning activities. To illustrate the instantiation process of the presented meta-model, a small example within this relevant context (Learning Analytics) is also provided.

TEEM'19, October 16–18, 2019, León, Spain

ACM ISBN 978-1-4503-7191-9/19/10...$15.00
https://doi.org/10.1145/3362789.3362837

CCS CONCEPTS

• Software and its engineering → Reusability • Human-centered computing → Visualization toolkits

KEYWORDS

Domain engineering; Meta-modeling; Information Dashboards; High-level requirements.

ACM Reference format:

A. Vázquez-Ingelmo, F. J. García-Peñalvo, and R. Therón. 2019. Capturing high-level requirements of information dashboards' components through meta-modeling. In *Proceedings of the Seventh International Conference on Technological Ecosystems for Enhancing Multiculturality (TEEM 2019) (León, Spain, October 16-18, 2019), ACM, New York, NY, USA*, 7 pages. https://doi.org/10.1145/3362789.3362837

1 Introduction

In a recent survey about information dashboards [1], Sarikaya et al. stated the following: "Visualization dashboards are ubiquitous." It is clear that these tools have become essential in nearly every process that involves decision-making. Dashboards support knowledge generation by composing different views and perspectives of data, allowing users to extract patterns and reach insights about a target domain visually.

But in the previously mentioned survey, another characteristic about dashboards arose: the fact that they are very different tools, both in terms of their visual perspective and their functional perspective [1]. This means that dashboards can appear with varying designs and features, depending on the users, datasets, and contexts they need to give support to.

It is this number of potential contexts in which a dashboard can be framed, and the number of possible user profiles that could use a dashboard what makes them complex tools that need extensive design phases.

One specific context that will be addressed in this paper is Learning Analytics (LA). Learning Analytics dashboards are increasing their popularity, given the potential benefits that they can provide to their stakeholders (teachers, students, etc.) [2]. But, as well as in other contexts, several users can present very different motivations and can be driven by different goals. Teachers could be more interested in summaries or how their students interact with each other, while students can be more interested in their individual results and contrasting them with the rest of the class.

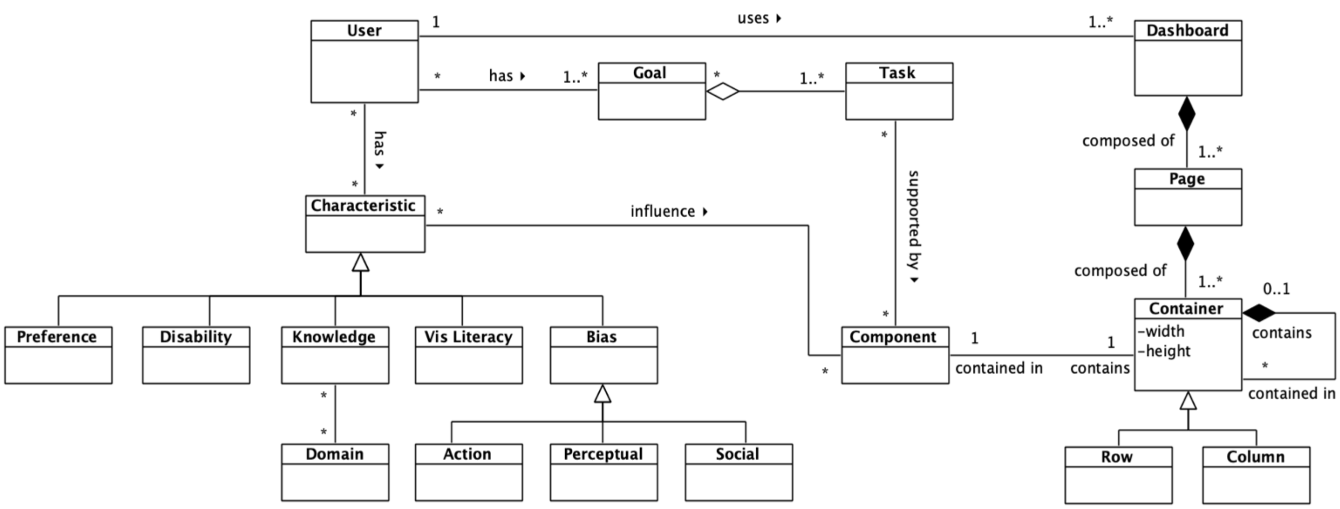


Figure 1: The user's section of the meta-model.

The variety of contexts and users is one of the main reasons to search for (and employ) paradigms that accelerate the development of dashboards without neglecting individual user experience and specific requirements.

However, it is not trivial to build a one-size dashboard that can adapt itself to evolving necessities and contexts. Customization, personalization, or adaptation are some of the most employed approaches to manage individualization regarding user interfaces. These approaches use generic solutions that can be explicitly configured to fit into new contexts and match new requirements [3].

Among the methods used to configure these tools, the software product lines (SPL) paradigm [4, 5] and model-driven development (MDD) [6, 7] can be found [8-12].

By abstracting common and high-level components of information dashboards, it is possible to boost reusability and decrease development times.

But another benefit of these approaches is the possibility of mapping features and abstract classes into concrete code, thus being possible to generate information dashboards automatically.

Identifying abstract features and arranging them into a robust meta-model is the first step to accomplish the benefits above. This paper describes the application of meta-modeling to obtain a set of high-level characteristics that can be applied to any dashboard.

Through the analysis of the domain using domain engineering, commonalities among dashboards are identified and connected to provide a generic dashboard structure that can be instantiated to match specific contexts.

The rest of this work is organized as follows. Section 2 describes the methodology followed to develop the dashboard meta-model. Section 3 presents the information dashboard meta-model. Section 4 discusses the meta-model elements, followed by section 5, where an instantiation example in the LA context is shown. The final sections present the discussion and conclusions derived from this work.

2 Methodology

2.1 Meta-modeling

The methodology employed makes use of meta-modeling. Meta-models are artifacts from the model-driven architecture paradigm [6, 7].

These artifacts allow capturing high level and abstract concepts, enabling a better understanding of the problem's domain. Meta-models are also useful to document and represent in a structured manner these concepts. This methodology fosters the development of general rules, constraints, structures, etc., for a set of related problems by abstracting commonalities and associations found in particular domain's instances.

The dashboard domain is complex, because several elements and technical properties can be involved, and these specific properties could be crucial to improve user experience and knowledge generation.

There is a huge diversity of dashboards in terms of features, design, interaction, or tasks that they can support. However, as it will be explained in the next sub-section, by using a domain engineering approach, it can be possible to extract commonalities among these tools and arrange them into abstract models.

These abstract models can be mapped to concrete products, according to the OMG four-layer meta-model architecture [13]: meta-meta-model layer (M3), meta-model layer (M2), user model layer (M1) and user object layer (M0).

In this work, the presented dashboard meta-model is an M2 model (an instantiation of the M3 layer, using MOF language), which, in turn, can be instantiated to obtain dashboard instances.

2.2 Domain engineering

Domain engineering [14] is based on knowledge reuse regarding some specific domain. This approach is an essential phase of the software product line (SPL) paradigm because it allows planning how different core assets and components will be built to boost development processes and decrease the time-to-market of different products from the same domain.

Dashboards can present several features, as mentioned before, but among all the types of dashboards that exist, there are shared and common properties that can be identified.

After studying the problem's domain, all these shared properties can be abstracted into a set of conceptual classes and relations among them, obtaining a simplified representation of the problem's domain.

3 The meta-model

The following section presents the proposed meta-model. This model can be divided into three main elements: the user, the layout, and the components. The most complex part is the abstraction of the components because they can present information, static resources, interactivity, etc. By analyzing different dashboards and visualization taxonomy studies, a decomposition of potential dashboard's components is presented.

3.1 The user

First of all, the user is the driver of the dashboard, and the whole individualization process is based on his or her characteristics and goals. These elements will influence the dashboard's components to match the user's needs.

The section of the meta-model that addresses the user is presented in Fig.1. As can be seen, goals involve a series of specific tasks, and characteristics can be preferences, disabilities, knowledge about certain domains, visualization literacy, and different kind of biases. These elements have been previously discussed; for more detail, refer to [15].

3.2 The layout

The layout section (Fig. 2) addresses the structure of a dashboard. Dashboards can be composed of different pages, with different containers (rows or columns) that hold different components.

3.3 Components

The dashboards' components are complex elements because several parts and possibilities are involved.

First of all, a component in a dashboard display does not have to be necessarily a visualization. It can also be a control that can handle several visualizations at once (e.g., global filters).

On the other hand, some of the dashboard's containers can hold graphical resources (e.g., images or illustrations) or text, to provide a context to the displayed information or instructions about how to employ the tool. But in the end, the main components of dashboards are the information visualizations that present the domain's data.

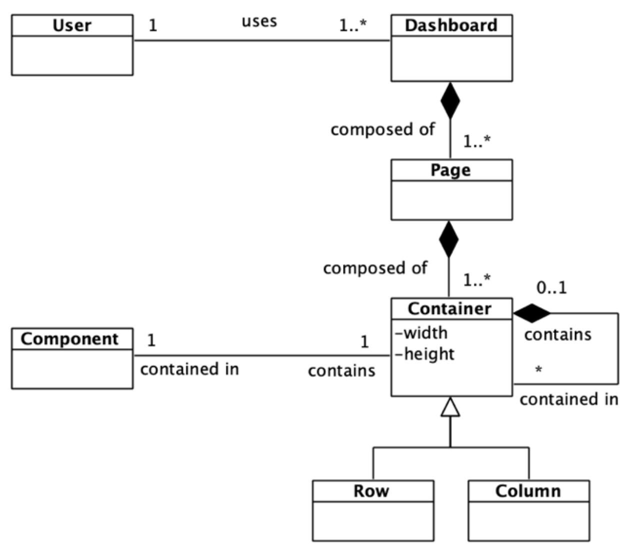


Figure 2: The layout's section of the meta-model.

A visualization can be affected by global controls, and also by "local" controls (i.e., controls that only affect that specific visualization). This distinction allows having control of visualizations both on global and local levels, thus letting users explore data more freely. In this case, a control is understood as any explicit handler that allows modifications on visualizations at any dimension: displayed data, design, visual encoding, etc.

Moreover, a visualization can be decomposed in lower-level elements that are shared among all the potential instances. That is why the meta-model reflects that a visualization is composed of one or more primitives. The "Primitive" class is a high-level class that encompasses different elements.

These elements can be axes, annotations, marks, and resources (images, text, etc.). But before detailing the mentioned low-level components, it is important to clarify that the visualizations' local controls can affect these primitives; as introduced, a control allows the modification of the visualizations, that is, their primitives, which are who hold the actual information.

In addition, a visualization's primitives can also be modified by the available interaction methods. For example, a visualization that allows zooming will change the primitives (specifically, the axes, the scales, and, therefore, the visual marks' encoding values) when the interaction method is employed. Once these classes and associations have been clarified, each primitive will be detailed.

First, one of the most important primitives regarding visualizations is axes. Axes contain information about the scales and thus, about some channels that can influence a visual mark, as it will be explained. Axes can take different forms, which are encoded as a meta-class attribute (*type*); for example, an axis can be linear or radial, presenting curvature in its presentation.

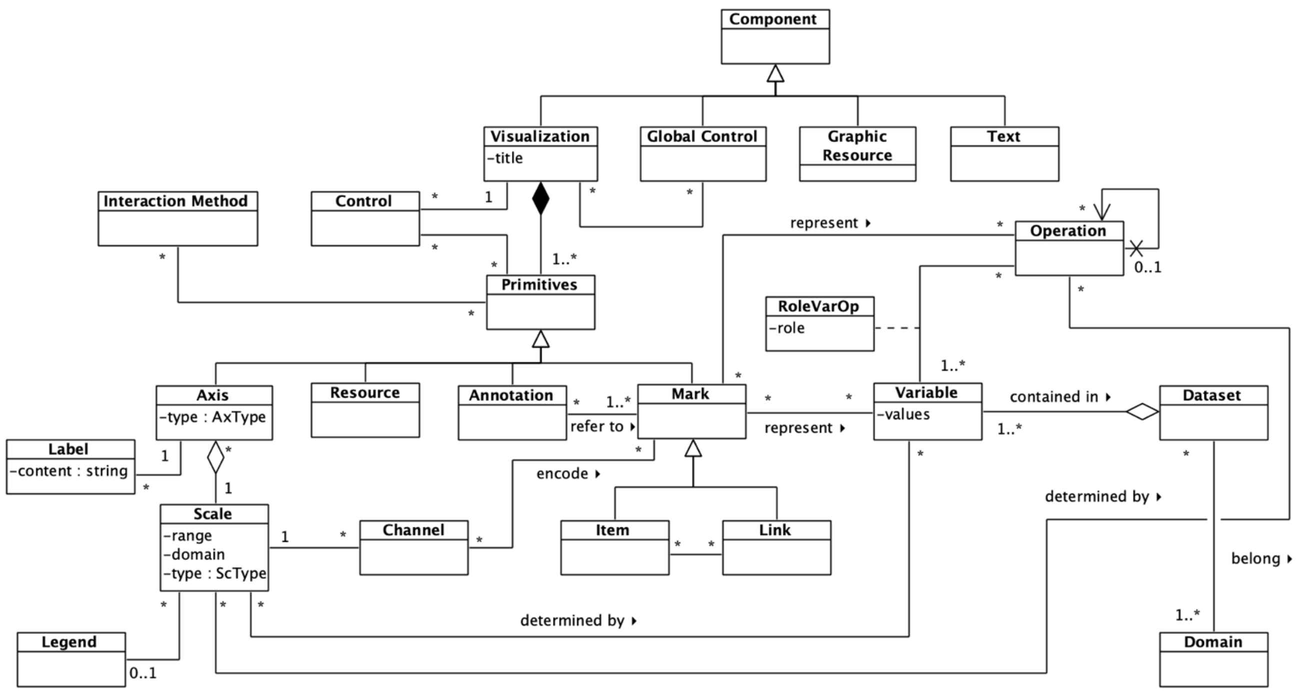


Figure 3: The components' section of the meta-model.

Axes can be labeled to clarify their role or the variable being represented. A meta-class *Label* is included in the diagram to reflect this fact.

On the other hand, an axis is always associated with a unique scale. A unique axis cannot represent more than one scale at once; however, a scale can be represented in several axes, providing redundant information, for instance.

Scales have a domain, a range and a type, the last referring to the nature of the data. Given the data properties, associated scales can be linear, ordinal, nominal, logarithmic, etc.; the selection of a proper scale is essential in the information visualization field, so the mentioned attributes are necessary for the meta-model. In addition, these attributes are common to any scale, so they are worth to include in an abstract representation of an information visualization.

Scales can be associated with a legend to improve the understandability of the visualization.

Relevant visualization elements have been explained so far, but their backbone is the visual encodings of data, that is, the marks that contain actual information about different data variables.

There are popular terms to refer to data elements, but the most used among the literature are marks and visual channels or visual encodings [16-19].

Marks can be items or links. Items represent nodes, points, series, zones, etc., and links represent connections, containments, etc., among items [19].

Marks represent data variables contained in a dataset or the outcomes of operations (arithmetic operations, aggregations, etc.), based on the PTAH meta-model presented in [10]. To visually represent the values, these can be encoded through different channels: position, size, length, color, opacity, angle, curvature, etc. The same channel can be employed to encode different marks. Channels are associated with a scale, which will map a variable's or operations' values to specific channel values.

Annotations are also taken into account in the meta-model, as they can be crucial elements in declarative visualizations, where the main focus is on explaining values rather than on exploring them [20]. Annotations can refer to different marks, and a mark can be affected by zero or more annotations.

Regarding the previously mentioned operations, it is also important to bear in mind the role of the different variables that might take part in an operation. For example, an aggregation would have groups and a target. For this reason, an association class (*RoleVarOp*) that models the role of a variable within an operation has been included. Moreover, a recursive association in the *Operation* class has been modeled to support chained operations between variables.

Finally, datasets can be associated with different domains. The *Domain* class is, in turn, associated with the meta-class *Knowledge* presented on the user's section of the meta-model in Fig. 1. This relation captures the familiarity that a user can have in terms of the data being represented.

The presented meta-model aims to represent at an abstract level the skeleton of any kind of information dashboard.

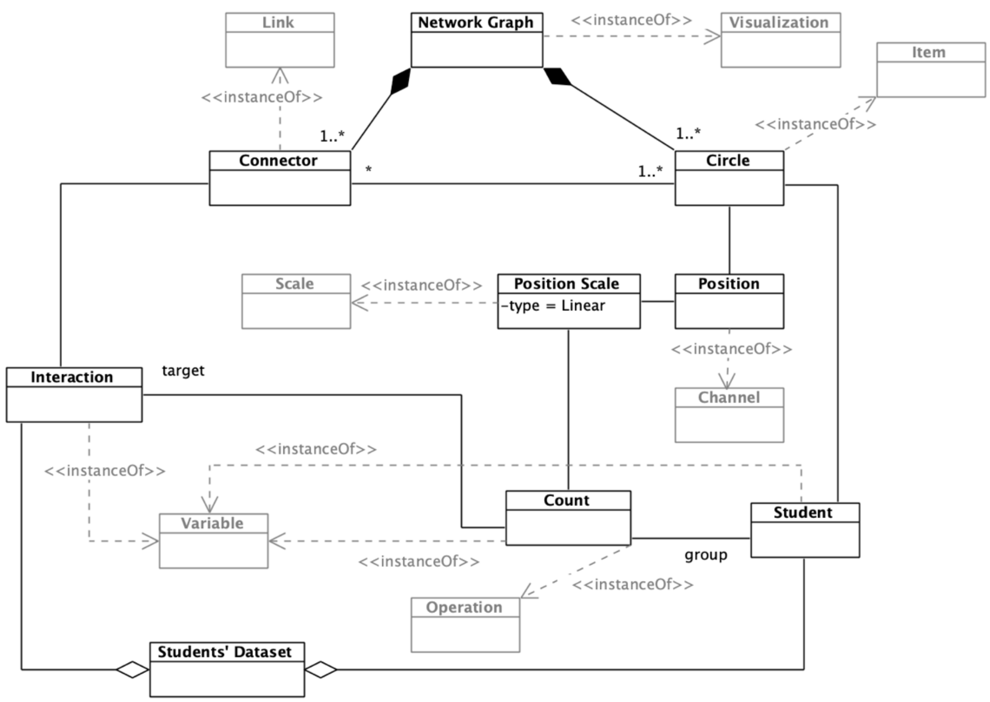


Figure 4: Possible instantiation of a network diagram to present students' interactions among them.

4 Meta-model instantiation

As mentioned before, the main advantage of developing a meta-model is the subsequent instantiations that can be derived from it. These instantiations enable the possibility of obtaining concrete dashboard instantiations for any context and user profile.

A small example in the context of LA dashboards is provided in this section to illustrate the instantiation process. LA dashboards are gaining relevance because of their potential benefits in learning processes, supporting not only teachers but also students in reaching insights about learning experiences.

A very recurrent visualization on this context is network diagrams to gain insights about how students interact with each other in discussion forums, assignments, or different kind of activities. Interaction data has been used mainly as a social-related indicator [21].

This kind of visualization can be useful for teachers to identify clusters and understand how his or her students work.

The main goal could be to "explore patterns regarding students' interactions."

A simplified network chart can be instantiated as in Figure 4 using the meta-model.

The instantiated network graph presents students' interactions through connections between circles, in this case. Each circle would represent a student who is positioned based on the number of interactions with other students. Students that interact a lot with each other would be closer, and students that don't interact at all would be isolated.

This is one possible instantiation of a network chart for LA, but other components could be added, like a new channel for each circle that encodes the total number of interactions of each students using a color or size channel. Also, the connectors could have other channels to encode each interaction or any other kind of relevant data.

5 Discussion

A meta-modeling approach is applicable to the dashboards domain, given its complexity and the necessity of extracting shared features to improve design and development mechanisms, also improving the quality of these tools.

Regarding the development of the meta-model, several factors have been taken into account to ensure the identification of common properties and elements — first, the user. Users are the drivers of dashboards; these tools are designed to reach their goals and assist their decision-making processes, so they are an unquestionable part of dashboards. To include the user in the meta-model is crucial to take them into account during the development phases explicitly.

Users have different attributes that might entirely influence a dashboard's configuration. As exposed in [1], "the intended use of a dashboard drives the choices in its visual design and functional affordances." This statement reflects the relevance of the users' goals in the process of developing dashboards. On the other hand, users' characteristics should not be underestimated. Currently, biases, visual literacy, and domain expertise are being treated as crucial factors that influence dashboards' and visualizations' effectiveness [22-31].

User preferences must also be included to give users the freedom to tailor their own dashboards, and any disability (eye diseases, hand tremors, etc.) that might influence the user experience must be a relevant factor for adapting the components to improve accessibility [32, 33].

Regarding the elements of a dashboard's components, several elements have been identified and detailed in section 3.3.

Representing the channels is crucial because several studies reflect the relevance of choosing a proper encoding in specific contexts to enhance perception [18, 19, 34]. Axes and scales are also necessary to ensure an appropriate representation of data through visual marks [35, 36]. Other resources, like labels, legends, and annotations, have been included, given their contextual role in information visualizations [37].

The purpose of having a meta-model is not only to understand such a complex domain but to developing an artifact that can be instantiated into concrete dashboards. Concrete dashboards would be useful to create a generation pipeline of these tools, thus automatizing the process of developing tailored dashboards based on the user, domain, tasks, etc.

Other solutions based on the model-driven paradigm or the software product line paradigm also made use of dashboard meta-models [6, 8, 10]. However, these meta-models are more focused on data properties and user roles than on the dashboard's elements and their relationships. Of course, data properties and user roles are crucial to choose a right visualization method, but adding common elements regarding dashboards' features allow a fine-grained understanding on how dashboards can be developed through the composition of generic pieces.

By abstracting dashboards' components, it has been possible to identify high-level features that are shared among any type of dashboard, obtaining a generic dashboard that could assist the automatic generation of tailored products.

The instantiation example helps illustrate the approach taken. As it has been shown, a complex domain such as Learning Analytics can benefit from the meta-model to account for stakeholders' characteristics and for the nature of the data.

The main strength comes from the generic structure and properties of the meta-model, which enables its application in a variety of distinct domains.

6 Conclusions

A dashboard meta-model has been developed to capture high-level requirements and elements found in the complex domain of data visualization.

The dashboards domain and its related literature have been studied to extract shared elements present in these tools. The identified elements have been arranged in a meta-model that can be instantiated to obtain concrete dashboards' models.

Future research lines will involve the refinement of the meta-model through the addition of constraints, rules, and design guidelines. These additions seek the support of an automatic generation of concrete dashboards by instantiating the meta-model. Finally, the meta-model must be validated through case studies.

ACKNOWLEDGMENTS

This research work has been supported by the Spanish *Ministry of Education and Vocational Training* under an FPU fellowship (FPU17/03276). This work has been partially funded by the Spanish Government Ministry of Economy and Competitiveness throughout the DEFINES project (Ref. TIN2016-80172-R) and the Ministry of Education of the Junta de Castilla y León (Spain) throughout the T-CUIDA project (Ref. SA061P17).

REFERENCES

[1] Alper Sarikaya, Michael Correll, Lyn Bartram, Melanie Tory, and Danyel Fisher. 2018. What Do We Talk About When We Talk About Dashboards? IEEE Transactions on Visualization Computer Graphics 25, 1, 682 - 692.
[2] Jeonghyun Kim, Il-Hyun Jo, and Yeonjeong Park. 2016. Effects of learning analytics dashboard: analyzing the relations among dashboard utilization, satisfaction, and learning achievement. Asia Pacific Education Review 17, 1, 13-24.
[3] Andrea Vázquez Ingelmo, Francisco J. García-Peñalvo, and Roberto Therón. 2019. Tailored information dashboards: A systematic mapping of the literature. In Proceedings of the Interacción2019 (Donostia, Spain2019). ACM.
[4] Paul Clements and Linda Northrop. 2002. Software product lines. Addison-Wesley, Boston, MA, USA.
[5] Klaus Pohl, Gunter Böckle, and Frank J. Van Der Linden. 2005. Software Product Line Engineering: Foundations, Principles and Techniques. Springer-Verlag New York, Inc., New York, NY, USA.
[6] Andreas Pleuss, Stefan Wollny, and Goetz Botterweck. 2013. Model-driven development and evolution of customized user interfaces. In Proceedings of the 5th ACM SIGCHI symposium on Engineering interactive computing systems ACM, 13-22.
[7] Anneke G Kleppe, Jos Warmer, and Wim Bast. 2003. MDA Explained. The Model Driven Architecture: Practice and Promise Addison-Wesley Longman Publishing Co., Inc., Boston, MA.
[8] Themis Palpanas, Pawan Chowdhary, George Mihaila, and Florian Pinel. 2007. Integrated model-driven dashboard development. Information Systems Frontiers 9, 2-3, 195-208.
[9] Maximilien Kintz, Monika Kochanowski, and Falko Koetter. 2017. Creating User-specific Business Process Monitoring Dashboards with a Model-driven Approach. In MODELSWARD, 353-361.
[10] Ivan Logre, Sébastien Mosser, Philippe Collet, and Michel Riveill. 2014. Sensor data visualisation: a composition-based approach to support domain variability. In European Conference on Modelling Foundations and Applications Springer, 101-116.
[11] Andrea Vázquez-Ingelmo, Francisco José García-Peñalvo, and Roberto Therón. 2018. Domain engineering for generating dashboards to analyze employment and employability in the academic context. In Proceedings of the 6th International Conference on Technological Ecosystems for Enhancing Multiculturality (Salamanca, Spain2018).
[12] Andrea Vázquez-Ingelmo, Francisco J. García-Peñalvo, and Roberto Therón. 2019. Taking advantage of the software product line paradigm to generate customized user interfaces for decision-making processes: a case study on university employability. PeerJ Computer Science 5(2019/07/01), e203. DOI:http://dx.doi.org/10.7717/peerj-cs.203.
[13] José M Álvarez, Andy Evans, and Paul Sammut. 2001. Mapping between levels in the metamodel architecture. In International Conference on the Unified Modeling Language Springer, 34-46.
[14] Maarit Harsu. 2002. A survey on domain engineering. Citeseer.
[15] Andrea Vázquez Ingelmo, Francisco J. García-Peñalvo, Roberto Therón, and Miguel Ángel Conde González. 2019. Extending a dashboard meta-model to account for users' characteristics and goals for enhancing personalization. In Proceedings of the Learning Analytics Summer Institute (LASI) Spain 2019 (Vigo, Spain2019).
[16] Jock Mackinlay, Pat Hanrahan, and Chris Stolte. 2007. Show me: Automatic presentation for visual analysis. IEEE Transactions on Visualization Computer Graphics 13, 6.
[17] Jock Mackinlay. 1986. Automating the design of graphical presentations of relational information. Acm Transactions On Graphics (Tog) 5, 2, 110-141.
[18] Noah Iliinsky and Julie Steele. 2011. Designing data visualizations: Representing informational Relationships. O'Reilly Media, Inc., Sebastopol, CA, USA.
[19] Tamara Munzner. 2014. Visualization analysis and design. AK Peters/CRC Press, Boca Raton, FL, USA.
[20] Scott Berinato. 2016. Good charts: The HBR guide to making smarter, more persuasive data visualizations. Harvard Business Review Press, Brighton, MA, US.
[21] Beat A Schwendimann, Maria Jesus Rodriguez-Triana, Andrii Vozniuk, Luis P Prieto, Mina Shirvani Boroujeni, Adrian Holzer, Denis Gillet, and Pierre Dillenbourg. 2017. Perceiving learning at a glance: A systematic literature review of learning dashboard research. IEEE Transactions on Learning Technologies 10, 1, 30-41.
[22] Jessica Hullman, Eytan Adar, and Priti Shah. 2011. The impact of social information on visual judgments. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems ACM, 1461-1470.
[23] Yea-Seul Kim, Katharina Reinecke, and Jessica Hullman. 2018. Data through others' eyes: The impact of visualizing others' expectations on visualization interpretation. IEEE transactions on visualization and computer graphics 24, 1, 760-769.
[24] Michael A Bedek, Alexander Nussbaumer, Luca Huszar, and Dietrich Albert. 2018. Methods for Discovering Cognitive Biases in a Visual Analytics Environment. In Cognitive Biases in Visualizations Springer, 61-73.
[25] André Caldero Valdez, Martina Ziefle, and Michael Sedlmair. 2017. A framework for studying biases in visualization research.
[26] Sukwon Lee, Sung-Hee Kim, and Bum Chul Kwon. 2017. Vlat: Development of a visualization literacy assessment test. IEEE transactions on visualization and computer graphics 23, 1, 551-560.
[27] Jeremy Boy, Ronald A Rensink, Enrico Bertini, and Jean-Daniel Fekete. 2014. A principled way of assessing visualization literacy. IEEE transactions on visualization and computer graphics 20, 12, 1963-1972.
[28] Adam V Maltese, Joseph A Harsh, and Dubravka Svetina. 2015. Data visualization literacy: Investigating data interpretation along the novice—expert continuum. Journal of College Science Teaching 45, 1, 84-90.
[29] Dereck Toker, Cristina Conati, Giuseppe Carenini, and Mona Haraty. 2012. Towards adaptive information visualization: on the influence of user characteristics. In International Conference on User Modeling, Adaptation, and Personalization Springer, Berlin, Germany, 274-285.
[30] Andrew Dillon. 2000. Spatial-semantics: How users derive shape from information space. Journal of the American Society for Information Science 51, 6, 521-528.
[31] Cristina Conati, Giuseppe Carenini, Enamul Hoque, Ben Steichen, and Dereck Toker. 2014. Evaluating the impact of user characteristics and different layouts on an interactive visualization for decision making. In Computer Graphics Forum Wiley Online Library, 371-380.
[32] Lisa Pappas and Lisa Whitman. 2011. Riding the technology wave: Effective dashboard data visualization. In Symposium on Human Interface Springer, 249-258.
[33] Somnath Arjun. 2018. Personalizing data visualization and interaction. In Adjunct Publication of the 26th Conference on User Modeling, Adaptation and Personalization ACM, 199-202.
[34] Jeffrey Heer and Michael Bostock. 2010. Crowdsourcing graphical perception: using mechanical turk to assess visualization design. In Proceedings of the SIGCHI conference on human factors in computing systems ACM, 203-212.
[35] Samuel Silva, Beatriz Sousa Santos, and Joaquim Madeira. 2011. Using color in visualization: A survey. Computers & Graphics 35, 2, 320-333.
[36] William S. Cleveland. 1985. The Elements of Graphing Data Hobart Press.
[37] Carl Edwards. 2017. One easy-to-forget step you need to enhance your data visualizations Yellowfin.

7.11 Appendix K. Extending a dashboard meta-model to account for users' characteristics and goals for enhancing personalization

Extending a dashboard meta-model to account for users' characteristics and goals for enhancing personalization

Andrea Vázquez-Ingelmo[1[0000-0002-7284-5593]], Francisco José García-Peñalvo[1[0000-0001-9987-5584]], Roberto Therón[1,2[0000-0001-6739-8875]] and Miguel Ángel Conde[3[0000-0001-5881-7775]]

[1] GRIAL Research Group, Computer Sciences Department, Research Institute for Educational Sciences, University of Salamanca, Salamanca, Spain
[2]VisUSAL Research Group. University of Salamanca. Salamanca, Spain
[3]Department of Mechanics, Computer Science and Aerospace Engineering. University of León. León, Spain.

{andreavazquez,fgarcia,theron}@usal.es
miguel.conde@unileon.es

Abstract. Information dashboards are useful tools for exploiting datasets and support decision-making processes. However, these tools are not trivial to design and build. Information dashboards not only involve a set of visualizations and handlers to manage the presented data, but also a set of users that will potentially benefit from the knowledge generated by interacting with the data. It is important to know and understand the requirements of the final users of a dashboard because they will influence the design processes. But several user profiles can be involved, making these processes even more complicated. This paper identifies and discusses why it is essential to include the final users when modeling a dashboard. Through meta-modeling, different characteristics of potential users are structured, thus obtaining a meta-model that dissects not only technical and functional features of a dashboard (from an abstract point of view) but also the different aspects of the final users that will make use of it. By identifying these user characteristics and by arranging them into a meta-model, software engineering paradigms such as model-driven development or software product lines can employ it as an input for generating concrete dashboard products. This approach could be useful for generating Learning Analytics dashboards that take into account the users' motivations, beliefs, and knowledge.

Keywords: Information Dashboards, Meta-model, Information Visualization, User Model, MDA.

1 Introduction

Information dashboards are compelling tools for generating knowledge and for supporting data-driven decisions. These tools allow users to visually understand and extract patterns from their datasets, fostering informed decision-making processes.

However, dashboards are also sophisticated tools, both in terms of development and use. First, the development of an information dashboard is not trivial; developers need to detail and understand the goal of the dashboard, the domain in which it will be framed, the information that will be presented, and, last but not least, the users that will use the dashboard.

Users, from an abstract point of view, are complex entities, with different characteristics from one to the other, with different behaviors, beliefs, and goals [1, 2]. This fact means that a specific dashboard configuration could be extremely beneficial for one individual, but entirely useless for another, as it could not match his or her goals, domain knowledge, visual literacy, and of course, his or her individual preferences.

In existing literature about the process of designing a dashboard, several authors point out the necessity of taking into account the problem to be solved through the visual presentation of data [3-5]. However, the problem definition is tightly related to the data domain and the user goals [6], thus needing to address the problem particularly in the target domain's context, spoiling the opportunity of reusing components, hence consuming time and resources.

Generalizing these user dimensions can be useful to understand the problem's domain better, to improve the dashboards' development processes, and to provide personalized products that take into account individual requirements. That is the reason why it is crucial to extract commonalities in user tasks and interactions, no matter the data context or domain. In the end, the user behavior is based in primitive tasks (pan, zoom, click, hover, etc.) that will provide them with outputs to reach their goals and to improve insights delivery processes.

Some software engineering paradigms can benefit from the abstraction of the elements that compose a dashboard, users included. Such paradigms, like model-driven development (MDD) [7] or software product lines (SPL) [8, 9] aim at decreasing development time by leveraging the reuse of software components or by mapping high-level models to concrete models or code.

In this paper, an extension of a previously developed dashboard meta-model [10-12] is presented. This extension takes into account different user dimensions that can influence dashboard components, to establish a framework for generating personalized dashboards that foster better user experience and insights delivery.

Characterizing the user could lead to benefits in fields like Learning Analytics (LA), where dashboards showing the users' learning data could be counterproductive if individual aspects are not addressed [13, 14].

The remainder of this paper is organized as follows. Section 2 describes the methodology followed to model the user from an abstract point of view using meta-modeling. In Section 3, the obtained meta-model is provided and explained. In Section 4, the meta-model is discussed, to finally close with Section 6, where conclusions and future research lines are presented.

2 Methodology

The followed methodology employs a meta-model, an artifact from the model-driven architecture paradigm [7, 15]. Meta-models are useful for capturing high level and abstract concepts, and not only for understanding the problem's domain but also to document and represent in a structured manner these concepts. Thus meta-modeling fosters the development of general rules, constraints, structures, etc., for a set of related problems by abstracting shared features and relations found in particular domain's instances.

But why applying meta-modeling to the dashboards' domain? As introduced, this domain is extraordinarily complex, because not only the technical features of a dashboard should be identified and detailed, but also the final users' characteristics that can influence their experience with the dashboard. Through domain engineering [16] processes, all these properties can be abstracted into a set of conceptual classes and relations among them, obtaining a simplified representation of the problem's domain.

These abstract models can be mapped to concrete products, according to the OMG four-layer meta-model architecture [17]: meta-meta-model layer (M3), meta-model layer (M2), user model layer (M1) and user object layer (M0). In this work, the presented dashboard meta-model is an M2 model (an instantiation of the M3 layer, using MOF language), which, in turn, can be instantiated to obtain dashboard instances.

3 The meta-model

In this section, the designed meta-model is presented. As introduced in the previous section, the level of abstraction of the meta-model is high, to capture generic commonalities among the potential objects. The main benefit from these levels of abstraction is the achievement of a general model from which concrete models can be instantiated.

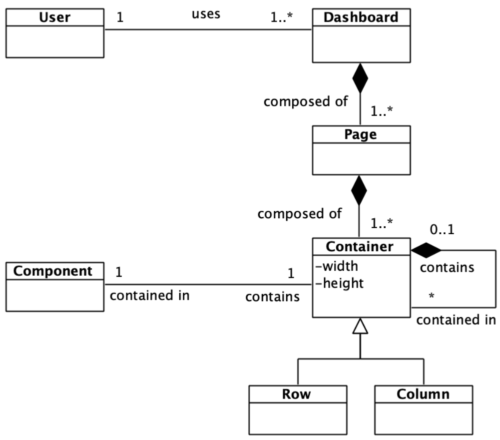


Fig. 1. The initial dashboard meta-model [10].

The initial dashboard meta-model to be extended consist of five main classes, and two specializations (showed in Fig. 1). This meta-model captures at a very high level the different components of an information dashboard, as well as its generic layout, which, in the end, is based on containers that can be either rows and columns. A user could employ one or more dashboards, and a dashboard, through this approach, belong to one user, because involving more users would introduce noise in the personalization process of a dashboard.

As can be seen in this simple meta-model, details are omitted. The *User* class represents a high-level user, but none of his or her characteristics are represented nor detailed. However, the user should be defined in terms of different significant and influential aspects to support a personalized dashboard design, thus being necessary to extend this meta-model with more elements regarding the users' characteristics and goals, as well as defining the relations of these aspects with the dashboard's components.

Given that, the extended dashboard meta-model is presented in Fig. 2. The diagram represents the same dashboard structure as in Fig. 1, but in this case, the user has been decomposed in terms of his or her goals and his or her characteristics.

Firstly, a new concept arises; *Goal*. A user employing a dashboard must have at least one goal, however implicit. Even users that want to explore data casually have a goal (that is, exploring data itself). That is the reason for the "one or more" (1..*) multiplicity. In turn, a goal can belong to any user, and users can share common and general purposes, explaining the "zero or more" (*) multiplicity on that side of the relation.

On the other hand, a goal can be broken down into individual and more specific tasks. Simple goals can be accomplished by performing one task, e.g. if a concrete goal is "to know which USA city has the largest number of inhabitants," a straightforward yet necessary task could be "to sort USA cities by population number," meaning that the dashboard components must support sorting capabilities.

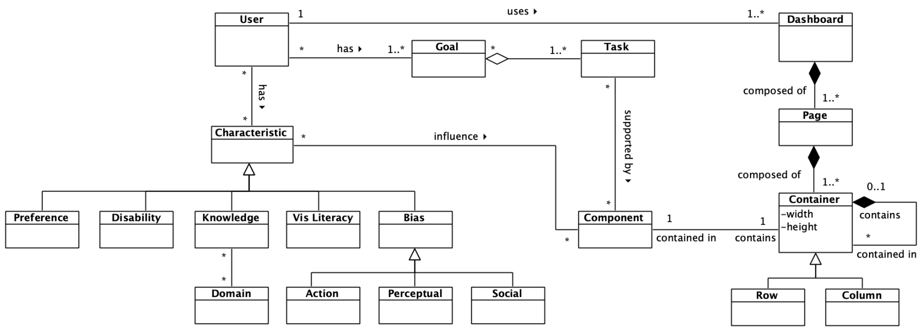


Fig. 2. The dashboard meta-model extension, including the user decomposition in terms of his or her characteristics and goals.

However, more complex goals might involve several specific and chained tasks such as "to understand why there has been a business income loss within the last six months," which could involve applying different tasks to different dimensions of the data to reach insights about the stated problem. That is the reason why the dashboard's components need to support the identified responsibilities to enable them.

Finally, a user can have zero or more identified characteristics, given the fact that, at a certain point, there could be no user data available of the possible dimensions. These characteristics can belong to zero or more users, as different users can share general characteristics. Characteristics can be of a different kind; preferences, disabilities, knowledge about different domains, visualization literacy, and bias (action, perceptual, or social bias). These characteristics can influence the dashboard's components to adapt them and, therefore, to match the identified user aspects.

4 Discussion

Including the user as an extremely significant element within the dashboard, domain is crucial. The development processes of a visualizations and dashboards start with the user (requirement elicitation) and end with the user (product refining) [5, 18], so not only the technical features of a dashboard should be taken into account when meta-modeling these tools, as these features arise from the users' requirements and are influenced by them [19].

The developed meta-model defines the users of a dashboard in terms of their goals and their characteristics. The users' goals drive the whole dashboard design processes, as it will influence user behavior when interacting with the dashboard's components [20]. However, goals are not enough to define a dashboard's configuration, it is necessary to decompose these goals into primitive tasks that can be directly supported by the dashboard's features (e.g., sort data, highlight data, annotate data, zoom, etc.) [21-23].

High-level user goals and user characteristics would be mapped low-level interactions in particular dashboard views presenting specific data dimensions to provide the user with a dashboard that could fulfill their information needs.

Once goals are addressed at high-level, the next phase is to take into account user preferences (implicitly exposed in its purposes, like, for example, the data that the user is interested in) as well as other characteristics, like the user's knowledge level about the data domain, the user's visual literacy and the user's potential biases. This process would provide the most suitable view type by configuring recognizable visual marks or visual metaphors, proper axes domains, preferred visual design, etc. Finally, user disabilities, such as color blindness, hand tremors, etc., would refine the dashboards' visual design and interaction methods by choosing right color palettes, mouse sensibility, etc.

The listed characteristics are hugely significant as they play an essential role when interpreting visualizations and reaching insights from them. For example, not being familiar with a type a visualization can lead to confusion and could be error prone when trying to reach insights [24, 25]. For these reasons, assessing visualization literacy is currently an important research field [26, 27], to address beforehand the users' visualization skills, delivering an understandable yet useful set of visualizations for them. Also, the users' knowledge level about the data domain should be addressed in the same manner; by providing views with right data dimensions and contextual information to mitigate unawareness about the domain [19].

On the other hand, user bias is not only influenced by past visualization experiences, but also by gender, age, race, etc. Why is it important to take this information into account? It could be seen as irrelevant factors, but the truth is that, unconsciously, bias could lead to valuable information loss [1, 2], that not only could undermine people but could also lead to financial losses by not addressing final users' bias when analyzing data [28]. But not only social biases (beliefs, expectations, etc.) are relevant within this context; action and perceptual biases can be harmful as well [29]. It is crucial to model dashboards taking into account these factors, because unintentionally, and from the user’s point of view, he or she could ignore data that could lead to beneficial decisions, thus being the insights reached half-truths.

Using generalization for modeling the above characteristics support the inclusion of new factors that might arise, allowing the meta-model’s evolution. These identified factors can influence dashboards to match both explicit and implicit characteristics, obtaining an effective and tailored visualization tool. However, there should also be room for customizing the dashboard, as the user should also have the freedom to craft their dashboards or to modify certain features. The main drawback of this approach is the retrieval of all the presented user dimensions, not only because several factors are involved, but because the information must be precise to map these characteristics into proper dashboard components successfully. Questionnaires about the different dimensions could be employed, like [26] for visual literacy, or even automatic approaches that measure these aspects through the analysis of users’ behavior [30].

Understanding user necessities is essential in the dashboard domain, but especially in some subdomains, such as LA dashboards. LA dashboards aim at visually assisting users (teachers, students, etc.) through a “single display that aggregates different indicators about learner(s), learning process(es) and learning context(s) into one or multiple visualizations," as stated in [31]. Personalizing these displays can foster self-regulated learning and academic achievement [13]. The presented meta-model can support personalization processes to achieve the mentioned benefits. Also, using this abstract meta-model can leverage reusability not only at a component-development level but also at design-level, by reusing knowledge.

5 Conclusions

In this paper, a dashboards meta-model extension is presented. The extension involves the inclusion of the final users as the main element of a dashboard, given their influence in the different design processes regarding the development of these tools. Different perspectives of the user are identified and discussed, such as the user goals, preferences, bias, disabilities, etc., to include them in the meta-model through high-level classes.

The purpose of having a dashboards meta-model is to provide a framework for instantiating any possible dashboard product, enabling personalization of individual dashboards. This approach could be useful for tailoring LA dashboards, where the necessities of each user can depend on their learning processes and motivations.

Future research lines would involve refining the meta-model through the addition of more specific properties, constraints, rules and the inclusion of design guidelines to support the automatic generation of concrete dashboards by instantiating the meta-model, and also designing questionnaires and methods to retrieve the presented user characteristics to finally implement the meta-model and validate it through case studies.

Acknowledgements

This research work has been supported by the Spanish *Ministry of Education and Vocational Training* under a FPU fellowship (FPU17/03276). This work has been partially funded by the Spanish Government Ministry of Economy and Competitiveness throughout the DEFINES project (Ref. TIN2016-80172-R), T-CUIDA project (Ref. SA061P17), and the PROVIDEDH project, funded within the CHIST-ERA Programme under the national grant agreement: PCIN-2017-064 (MINECO, Spain).

References

1. Hullman, J., Adar, E., Shah, P.: The impact of social information on visual judgments. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, pp. 1461-1470. ACM, (2011)
2. Kim, Y.-S., Reinecke, K., Hullman, J.: Data through others' eyes: The impact of visualizing others' expectations on visualization interpretation. IEEE transactions on visualization and computer graphics 24, 760-769 (2018)
3. Mazza, R.: Introduction to information visualization. Springer Science & Business Media (2009)
4. Few, S.: Information dashboard design. (2006)
5. Berinato, S.: Good charts: The HBR guide to making smarter, more persuasive data visualizations. Harvard Business Review Press (2016)
6. Colomo-Palacios, R., Casado-Lumbreras, C., Soto-Acosta, P., Misra, S., García-Peñalvo, F.J.: Analyzing human resource management practices within the GSD context. Journal of Global Information Technology Management 15, 30-54 (2012)
7. Kleppe, A.G., Warmer, J., Bast, W.: MDA Explained. The Model Driven Architecture: Practice and Promise. Addison-Wesley Longman Publishing Co., Inc., Boston, MA (2003)
8. Clements, P., Northrop, L.: Software product lines. Addison-Wesley (2002)
9. Gomaa, H.: Designing Software Product Lines with UML: From Use Cases to Pattern-Based Software Architectures. Addison Wesley Longman Publishing Co., Inc. (2004)
10. Vázquez Ingelmo, A.: Aplicación de Ingeniería de Dominio para la generación de dashboards personalizados. Grupo GRIAL (2018)
11. Vázquez-Ingelmo, A., García-Peñalvo, F.J., Therón, R.: Addressing Fine-Grained Variability in User-Centered Software Product Lines: A Case Study on Dashboards. In: World Conference on Information Systems and Technologies, pp. 855-864. Springer, (2019)
12. Vázquez-Ingelmo, A., García-Peñalvo, F.J., Therón, R.: Taking advantage of the software product line paradigm to generate customized user interfaces for decision-making processes: a case study on university employability. PeerJ Computer Science (Article In Press)
13. Roberts, L.D., Howell, J.A., Seaman, K.: Give me a customizable dashboard: Personalized learning analytics dashboards in higher education. Technology, Knowledge and Learning 22, 317-333 (2017)

14. Jivet, I., Scheffel, M., Drachsler, H., Specht, M.: Awareness is not enough: pitfalls of learning analytics dashboards in the educational practice. In: European Conference on Technology Enhanced Learning, pp. 82-96. Springer, (2017)
15. Pleuss, A., Wollny, S., Botterweck, G.: Model-driven development and evolution of customized user interfaces. In: Proceedings of the 5th ACM SIGCHI symposium on Engineering interactive computing systems, pp. 13-22. ACM, (2013)
16. Harsu, M.: A survey on domain engineering. Citeseer (2002)
17. Álvarez, J.M., Evans, A., Sammut, P.: Mapping between levels in the metamodel architecture. In: International Conference on the Unified Modeling Language, pp. 34-46. Springer, (2001)
18. Plaisant, C.: The challenge of information visualization evaluation. In: Proceedings of the working conference on Advanced visual interfaces, pp. 109-116. ACM, (2004)
19. Sarikaya, A., Correll, M., Bartram, L., Tory, M., Fisher, D.: What Do We Talk About When We Talk About Dashboards? IEEE Transactions on Visualization Computer Graphics (2018)
20. Gotz, D., Wen, Z.: Behavior-driven visualization recommendation. In: Proceedings of the 14th international conference on Intelligent user interfaces, pp. 315-324. ACM, (2009)
21. Amar, R., Eagan, J., Stasko, J.: Low-level components of analytic activity in information visualization. In: IEEE Symposium on Information Visualization, 2005. INFOVIS 2005., pp. 111-117. IEEE, (2005)
22. Gotz, D., Zhou, M.X.: Characterizing users' visual analytic activity for insight provenance. Information Visualization 8, 42-55 (2009)
23. Saket, B., Endert, A., Demiralp, C.: Task-based effectiveness of basic visualizations. IEEE transactions on visualization and computer graphics (2018)
24. Börner, K., Maltese, A., Balliet, R.N., Heimlich, J.: Investigating aspects of data visualization literacy using 20 information visualizations and 273 science museum visitors. Information Visualization 15, 198-213 (2016)
25. Maltese, A.V., Harsh, J.A., Svetina, D.: Data visualization literacy: Investigating data interpretation along the novice—expert continuum. Journal of College Science Teaching 45, 84-90 (2015)
26. Lee, S., Kim, S.-H., Kwon, B.C.: Vlat: Development of a visualization literacy assessment test. IEEE transactions on visualization and computer graphics 23, 551-560 (2017)
27. Boy, J., Rensink, R.A., Bertini, E., Fekete, J.-D.: A principled way of assessing visualization literacy. IEEE transactions on visualization and computer graphics 20, 1963-1972 (2014)
28. Perez, C.C.: Invisible Women: Exposing Data Bias in a World Designed for Men. Random House (2019)
29. Valdez, A.C., Ziefle, M., Sedlmair, M.: A framework for studying biases in visualization research. (2017)
30. Bedek, M.A., Nussbaumer, A., Huszar, L., Albert, D.: Methods for Discovering Cognitive Biases in a Visual Analytics Environment. Cognitive Biases in Visualizations, pp. 61-73. Springer (2018)
31. Schwendimann, B.A., Rodriguez-Triana, M.J., Vozniuk, A., Prieto, L.P., Boroujeni, M.S., Holzer, A., Gillet, D., Dillenbourg, P.: Perceiving learning at a glance: A systematic literature review of learning dashboard research. IEEE Transactions on Learning Technologies 10, 30-41 (2017)

7.12 Appendix L. Taking advantage of the software product line paradigm to generate customized user interfaces for decision-making processes: a case study on university employability

PeerJ Computer Science

Taking advantage of the software product line paradigm to generate customized user interfaces for decision-making processes: a case study on university employability

Andrea Vázquez-Ingelmo[1], Francisco J. García-Peñalvo[1] and Roberto Therón[1,2]

[1] GRIAL Research Group, Department of Computer Science and Automatics, University of Salamanca, Salamanca, Spain
[2] VisUSAL Research Group, Department of Computer Science and Automatics, University of Salamanca, Salamanca, Spain

ABSTRACT

University employment and, specifically, employability has gained relevance since research in these fields can lead to improvement in the quality of life of individual citizens. However, empirical research is still insufficient to make significant decisions, and relying on powerful tools to explore data and reach insights on these fields is paramount. Information dashboards play a key role in analyzing and visually exploring data about a specific topic or domain, but end users can present several necessities that differ from each other, regarding the displayed information itself, design features and even functionalities. By applying a domain engineering approach (within the software product line paradigm), it is possible to produce customized dashboards to fit into particular requirements, by the identification of commonalities and singularities of every product that could be part of the product line. Software product lines increase productivity, maintainability and traceability regarding the evolution of the requirements, among other benefits. To validate this approach, a case study of its application in the context of the Spanish Observatory for University Employability and Employment system has been developed, where users (Spanish universities and administrators) can control their own dashboards to reach insights about the employability of their graduates. These dashboards have been automatically generated through a domain specific language, which provides the syntax to specify the requirements of each user. The domain language fuels a template-based code generator, allowing the generation of the dashboards' source code. Applying domain engineering to the dashboards' domain improves the development and maintainability of these complex software products given the variety of requirements that users might have regarding their graphical interfaces.

Subjects Human–Computer Interaction, Software Engineering
Keywords SPL, DSL, Domain engineering, Dashboards, Employability, Code generation

Submitted 9 November 2018
Accepted 2 June 2019
Published 1 July 2019

Corresponding author
Andrea Vázquez-Ingelmo,
andreavazquez@usal.es

Academic editor
Helen Petrie

Additional Information and Declarations can be found on page 24

DOI 10.7717/peerj-cs.203

How to cite this article Vázquez-Ingelmo A, García-Peñalvo FJ, Therón R. 2019. Taking advantage of the software product line paradigm to generate customized user interfaces for decision-making processes: a case study on university employability. *PeerJ Comput. Sci.* **5**:e203 http://doi.org/10.7717/peerj-cs.203

INTRODUCTION

The concept of employability has increasingly gained relevance over the last decades. There is a reason: knowing which factors increase the possibility to obtain a job or to perform better in current job positions could be decisive to improve individual and collective life quality.

However, this concept is still far away from having a straightforward definition (*Chadha & Toner, 2017*). As the literature suggests, employability can be seen as a capability to gain employment or as a set of skills and knowledge required to perform effectively in the workplace, among other definitions (*Universities UK & Confederation of British Industry, 2009*; *Hillage & Pollard, 1998*; *Yorke, 2006*). This lack of consensus when defining employability makes the research in this field a complicated task, given the fact that the definition of its factors depends on the perspective used to evaluate it, as well as the socioeconomic context in which employability and employment studies are framed. For these reasons, nowadays research on employability asks for an exploratory approach, to build stronger theoretical foundations.

Researching on employability has many potential benefits, aiming not only at knowing the variables that affect the capability to gain employment and have a successful work career, but also to exploit this knowledge to help policymakers and institutions with their missions. This knowledge can contribute to the creation of greater policies, focusing on the detected factors to enhance people's chances to obtain better employment. Specifically, educational institutions like universities could benefit from this knowledge. These institutions play a vital role regarding the employability of individuals (*García-Peñalvo, 2016*), as they are in charge of transmitting knowledge and a series of skills to their students. By promoting the most relevant skills and capabilities that affect employability, it could be possible to increase the alignment of education with the labor market.

However, generating knowledge in such a study field is not a trivial task. As it has been introduced, there could be several variables involved in the research of students' employment and employability, so it is necessary to collect significant data volumes to be able to reach valuable insights. In addition to data collection, performing data analysis (*Albright, Winston & Zappe, 2010*) is required to be able to reach useful insights. It is worth noting that analyzing employability data to identify and understand its factors could become a cornerstone in decision-making processes within educational institutions.

Nevertheless, even after performing data analysis, identifying patterns and indicators derived from the analysis outcomes remains a complex challenge. That is why it is crucial to assist decision-makers with powerful tools that allow reaching insights about the domain of the problem, to support decisions with complete and quality information (especially in the academic context, where these processes might have a series of social implications), that is, information and knowledge that has been extracted through visual analysis.

Information dashboards are one of the most commonly used software products for visual data analysis and knowledge extraction (*Few, 2006*; *Sarikaya et al., 2018*). In a domain like employability, these tools can support exploratory studies through a set of graphical and interactive resources, allowing users to envision data more understandably (*Tufte & Graves-Morris, 2014*) and identify relevant relations, indicators or patterns among large sets of data. It is essential to bear in mind that information dashboards are not just a set of aesthetic graphs and visualizations; they have to effectively transmit information to answer the questions of the users regarding the target domain. Moreover, this is not a trivial job, because of two main reasons: data and users themselves.

On the one hand, users do not have a set of standard and static requirements; they could demand different features or design attributes given their specific goals or needs. On the other hand, data is continuously increasing and evolving nowadays, so it is foreseeable that new information requirements will arise in time. Returning to the employability subject, information requirements in this domain might change in many different ways as this concept could demand new kind of variables or larger amounts of data to explore emerging dimensions or to perform more in-depth analyses.

For these reasons, information dashboards not only need to be useful concerning functionality but also be customizable to adapt to specific user requirements. Also, they should be flexible and scalable regarding its data sources and structures, making the development and maintenance of information dashboards even more complicated. Of course, these issues could be addressed by developing particular dashboards for each involved user to achieve every specific goal, but clearly, this solution would be time-consuming and would require a lot of resources during the development and maintenance phases. Also, scalability would be almost impossible, as new users or changes in the requirements would necessarily imply more resources.

There are, nevertheless, a series of strategies to deal with these challenges. Specifically, software engineering paradigms like software product lines (*Clements & Northrop, 2002*; *Gomaa, 2004*; *Pohl, Böckle & Van der Linden, 2005*) provide powerful theoretical frameworks to address flexibility, scalability and customization in software products that share sets of features within a common domain. Through the analysis of commonalities and variability points in the product domain, it would be possible to reduce the development and maintenance effort of building tailor-made solutions. This paradigm is potentially applicable to dashboards since these software products could be factored into sets of configurable components with configurable features. This paper describes the application of the SPL methodology to the dashboards' domain through the study of their characteristics and the definition of a DSL to manage the product derivation automatically. The main focus of this research is to test the potential usefulness and feasibility of this approach to manage fine-grained features that can be scattered through different code assets, and consequently, to provide a base method for generating personalized dashboard solutions to fit concrete user requirements.

The remainder of this work is structured as follows. Background discusses the background of the problem of generating customized dashboards as well as their application to the employment and employability domain. Context presents the application context and the motivation behind this pilot framework to generate dashboards to support visual analysis on university employment and employability data (framed within the Spanish Observatory for University Employability and Employment studies. Materials and Methods describes the techniques used for the development of an initial approach to a generative dashboard framework. Finally, the Results section exhibits the outcomes of this research to conclude with the discussion of the developed SPL and the conclusions derived from these results.

BACKGROUND

The main idea behind software product lines (SPLs) is that the final products can be derived from a set of configurable core assets, allowing their adaptation to fit specific requirements. These core assets are developed during the domain engineering phase, in which commonality and variability of the target product domain are identified to build a common base of components. Core assets are developed with variability points in which specific functionalities could be injected to obtain new products. Functionalities in SPLs are seen as features; the combination of the defined features within the scope of the line (generally following a feature model (*Kang et al., 1990*) allow stakeholders to build personalized products by reusing and assembling software components.

The SPL paradigm has been applied to a variety of domains: Mobile applications (*Marinho et al., 2010*; *Nascimento, 2008*; *Quinton et al., 2011*); Applications for visualizing population statistics (*Freeman, Batory & Lavender, 2008*); Sensor data visualizations (*Logre et al., 2014*); Variable content documents (*Gómez et al., 2014*); or e-Learning systems (*Ezzat Labib Awad, 2017*).

These practical applications have proved the benefits of this paradigm. However, features usually refer to the software's logic, deflecting attention to the presentation layer. The idea of generating customized dashboards can be seen as a specific case of graphical user interfaces (GUI) automatic generation within SPLs. User interfaces require additional work regarding their implementation; they not only need to be functional but also usable to allow users to complete their tasks efficiently and achieve their goals. That is why the design of user interfaces is present through the whole development process, being time- and resource-consuming job.

Automation regarding GUI generation in software product lines has already been faced in several works. Generally, there is a lack of usability on the generated products that can be addressed by manually designing every product GUI. But this approach is highly inefficient in the SPL paradigm context since all the development time saved could be lost by introducing a manual task (*Hauptmann et al., 2010*). Integration of the GUI design process and the SPL paradigm is required to leverage the benefits of the two approaches (*Pleuss, Botterweck & Dhungana, 2010*). There is, as *Pleuss et al. (2012a)*; *Pleuss et al. (2012b)* pointed out, a dilemma between automation and usability. To address this challenge, they utilized Model-Based UI Development (MBUID) methods to separate the functionality and the appearance of the GUI (*Pleuss, Botterweck & Dhungana, 2010*).

On the other hand, *Gabillon, Biri & Otjacques (2015)* demonstrated the possibility of creating adaptive user interfaces through the Dynamic SPL (DSPL) paradigm and MBUID models by developing a context-aware data visualization tool that can be adapted during runtime.

DSPLs provide a useful paradigm for adapting code at run-time, obtaining adaptive GUIs. *Kramer et al. (2013)* proposed document-oriented GUIs with run-time variations through XML documents (*Kramer et al., 2013*). This context-adaptable feature has also been achieved by *Sboui, Ayed & Alimi (2018)*, by developing a mobile application that is also runtime adaptable through MBUID models and reusable artifacts. In this particular case, the code generation is based on eXtensible Stylesheet Language Transformations (XSLT) and XML files (*Sboui, Ayed & Alimi, 2018*). These works shows not only the viability of GUI generation in the SPL/DSPL paradigms context but also their valuable benefits.

It seems evident that GUI customization requires fine-grained features to achieve the desired usability and design attributes. Fine-grained features mostly require annotative approaches regarding their implementation, given their specialization. Annotative approaches can address this issue because annotations can be arbitrarily specified at different source code fragments (*Kästner & Apel, 2008*; *Kästner, Apel & Kuhlemann, 2008*), and provide a framework for fine-grained automated software composition through feature structure trees (*Apel, Kastner & Lengauer, 2009*).

There are different approaches to manage the implementation of variability at a fine-grained level (*Gacek & Anastasopoules, 2001*). Especially, frame- and template-based approaches provide valuable solutions to address this fine-grained level of variability, allowing the injection of particular fragments of code at any point of the base source code. Frame-based languages, like XML-based Variant Configuration Language (XVCL) (*Jarzabek et al., 2003*), provide a syntax to combine and insert fragments of code through the definition of frames, allowing the separation of concerns regarding the SPL implementation (*Zhang, Jarzabek & Swe, 2001*). Templating can also achieve valuable results; templating libraries such as Jinja2 (*Ronacher, 2008*) provide powerful functionalities to annotate the source code independently of the target programming language (*Clark, 2018*; *Ridge, Gaspar & Ude, 2017*).

The generation of GUI within the context of a product family is still a convoluted field, although the previous work has enlightened the path to improve and leverage the automation and generation of these complex software elements. The complexity mainly comes from human factors and the vast variety of requirements regarding user interfaces.

This work aims to present an application of the SPL paradigm, in this case on the dashboards' domain, considering the fine-grained nature of their features and the necessity of customizing its interaction methods and visual appearance.

CONTEXT

The application of this work is framed within The Spanish Observatory for University Employment and Employability. The following subsections describe this organization's mission and the motivation to generate personalized dashboards to explore its data.

The observatory for university employment and employability

The Observatory for University Employment and Employability (also known as OEEU, its Spanish acronym, http://oeeu.org) is an organization with the vision of becoming an information reference for understanding and exploiting knowledge about employment and employability of students from Spanish universities. To do so, this network of researchers and technicians conduct studies about these fields in the academic context (*Michavila et* al., 2018a; Michavila et al., 2016; Michavila et al., 2018b), through a data-driven approach to recollect, analyze, visualize and disseminate employment and employability data of graduates from Spanish universities.

Firstly, in the data collection phase, universities provide their administrative records and, once this phase is completed, their students answer a questionnaire about different aspects of their education and work career. This process leaves the Observatory with a significant set of variables from the students' sample. For instance, in the 2015 study edition, more than 500 variables were gathered from 13,006 bachelor students. Moreover, in the 2017 study edition, 376 variables were gathered from 6,738 master degree students.

The volume of the data collected makes the presentation of the study results to the Observatory ecosystem's users a challenge, as the latter may have different requirements and necessities regarding the studies' data. For these reasons, an approach based on domain engineering fits the OEEU's needs, allowing an efficient generation of customized dashboards that meet different requirements.

Motivation

As it has been introduced, employment and employability are complex study fields that mainly ask for exploratory analysis, given its relatively initial status of research. In the context of the Spanish Observatory for University Employment and Employability, where a vast set of variables from significant quantities of students are recollected, it is crucial to rely on exploratory visualizations that allow users and administrators to identify at a glance unusual patterns or important data points by enhancing the understanding of the presented information (Card, 1999).

In contrast with explanatory visualizations, in which the primary purpose is to tell a story through data, exploratory tools aim to facilitate users to pose more questions as data is being explored. In essence, explanatory analyses start from a question and use data to answer it. Exploratory analysis, on the other hand, uses data to detect new avenues of research. For instance, when a user does not have a clear question about the data, it will use exploratory research to find patterns or relations among variables. This same user could employ the acquired knowledge to explain the insights reached through previous explorations using an explanatory visualization.

Exploratory visualizations rely intensely on interaction to provide their functionality and to allow users to drill-down datasets, being able to discover new aspects of the domain by directly communicating with the graphical interface. However, an interaction can take many forms, and there is not a single solution to obtain usable and intuitive interfaces valid for every user.

For instance, some users could find useful a visible control panel to manage data if they are going to apply filters, aggregations and so on intensively. On the other hand, other users can demand in-place interaction if they give more importance to having more space for the visualizations (instead of having a permanent control panel consuming screen space). Another example is that users that speak a left-to-right (LTR) or a right-to-left (RTL) language would demand different layouts for the same task, according to their sociodemographic or cultural context (Almakky, Sahandi & Taylor, 2015; Marcus & Gould, 2000). Also, visualization novices could require task-oriented dashboards to support their visual analysis, since their past experience with this kind of tools is a relevant factor when interacting with a system (*Elias & Bezerianos, 2011*).

Once patterns, relations between variables and interesting dimensions have been identified through the exploration of data, even the exploratory nature of a dashboard can change for a more explanatory purpose to present the results understandably and strikingly.

For all these reasons dashboards, their components, their interaction, and even their primary purpose need advanced configuration and customization to fit into different contexts and requirements. Moreover, as it has been aforementioned, SPLs provide a potential solution to efficiently address this customization since visual components and interaction methods could be treated as features of the product line, decreasing the resources needed during the development and maintenance of dashboards.

MATERIALS & METHODS

This section presents the materials and techniques used during the development of this first approach to a framework for generating dashboards to explore employment- and employability-related variables.

Meta-model

The problem to address requires abstract modelling to capture basic features within the dashboards' domain. To do so, a meta-model is proposed. Meta-models are a crucial artefact in model-driven engineering and model-driven architectures (*Kleppe, Warmer & Bast, 2003*), as they allow to define a high-level view of the domain without depending on specific technologies. Therefore, meta-models should remain as simple as possible to eventually, through a series of mappings and transformations, obtain concrete models (*Álvarez, Evans & Sammut, 2001*).

For this generic dashboards' domain, the meta-model found in Fig. 1 is proposed. First of all, a specific user could handle a dashboard. This dashboard could be composed of one or more pages, being these last composed, in turn, by one or more containers. A container could be seen as a row or a column, and it can recursively contain more containers. The container recursion ends with a component, which is any graphic element that can be used in a dashboard. The recursion mentioned above allows the arrangement of any layout by the recurrent combination of rows and columns.

This meta-model eases the vision of the dashboards' domain, and it also allows to identify the common base of any dashboard.

Feature model

The meta-model gives a high-level vision of the dashboards' domain. However, it does not capture concrete features. That is why software product lines rely on feature models (*Kang et al., 1990*) to identify common and variable assets.

Feature models not only serve as a documentation element but also as an important artifact within the development process. The implementation of the core assets and the materialization of variability points on the code must be guided by the previously defined feature model.

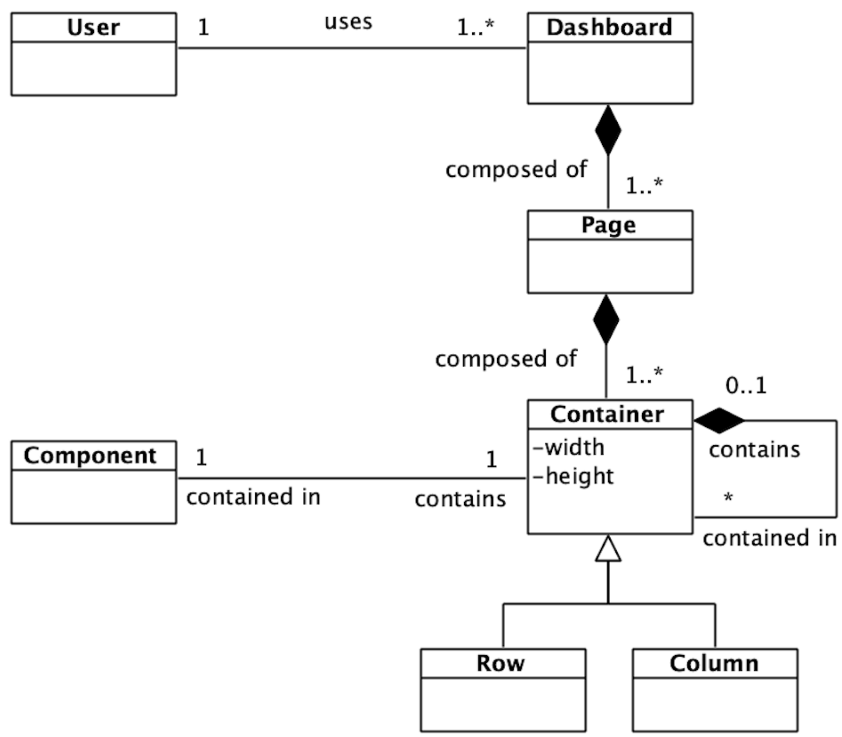


Figure 1 **Dashboard meta-model.** The dashboard meta-model allows a high level view of the target domain.

Full-size DOI: 10.7717/peerjcs.203/fig-1

In this domain, the feature model will capture the dashboards' visualization components, as well as individual features and restrictions of each visualization. The hierarchical structure of the feature model allows to define high-level characteristics and refine them through the tree structure until reaching the lower-level features (i.e., fine-grained features). This structure makes the scalability of features easier, since adding new features involves the addition of new nodes to the feature tree uniquely.

For the Observatory's dashboards, three main configurable visual components (features) have been defined: a scatter diagram, a chord diagram and a heat map. These visualizations address the requirements of the Observatory's data but can be reused for other data domains. Also, it is possible to specify a global filter that affects the data of all components previously defined. These high-level features of the dashboards' product line are presented in Fig. 2.

A detailed view of the scatter diagram feature can be seen in Fig. 3. It has a set of subsequent features, either mandatory, optional or alternative. One mandatory feature is the base logic of the scatter diagram (i.e., the component layout construction and its primary logic). Another mandatory feature is the initial data that the diagram will be showing on different dimensions since it must be specified. Among the optional features, it is possible to determine whether a tooltip will show up when hovering on data points if a set of controls will support the data exploration, or the capacity to zoom or export the diagram. Also, a title for the visualization can be included.

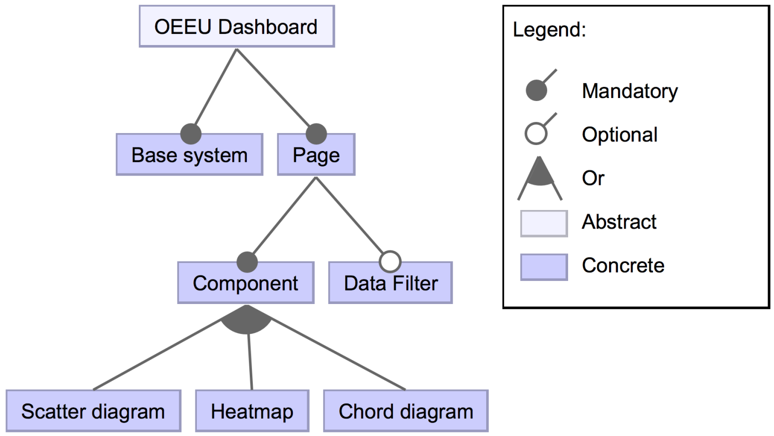


Figure 2 High-level view of the feature diagram. This feature diagram shows high-level components that could compose the dashboard.
Full-size DOI: 10.7717/peerjcs.203/fig-2

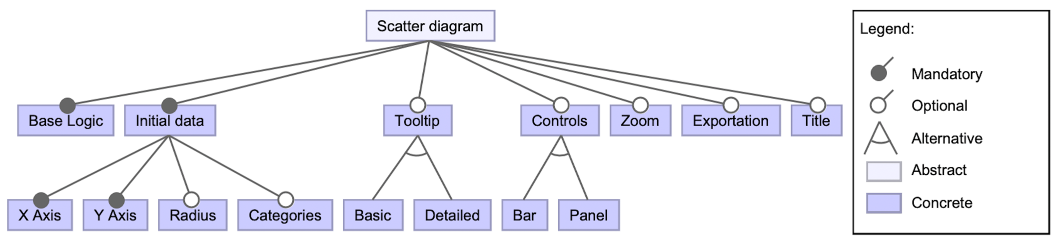


Figure 3 High-level view of the scatter diagram component's features. This snippet of the feature model shows the possible features regarding the scatter diagram component.
Full-size DOI: 10.7717/peerjcs.203/fig-3

For the sake of simplicity, some of the lower-level features have been omitted in Fig. 3. For instance, the bar and panel control features have subsequent features. The detailed features for a panel type control are shown in Fig. 4 to provide an example. A control panel will rely on its underlying logic, and it can count on different optional features, like data selectors to dynamically change the visualization's presented data; in case of the X and Y axes, these selectors could be located within the control panel space or in-place controls (i.e., situated near the scatter diagram axes). Other possible features involve having an overview that shows a detailed view of a data point when hovering, data filters, among others.

The feature diagram provides a high-level and organized overview of the SPL, improving the organization of the source code and development tasks.

Domain-specific language

There is, however, a necessity of connecting the previous models to the dashboards' source code to be generated (*Voelter & Visser, 2011*). A Domain-Specific Language (DSL) has been designed to accomplish this connection. This DSL is based on the identified domain's features, by structuring them with XML technology (*Bray et al., 1997*) and by validating the model restrictions with an XML schema (*Fallside, 2000*). XML technology provides

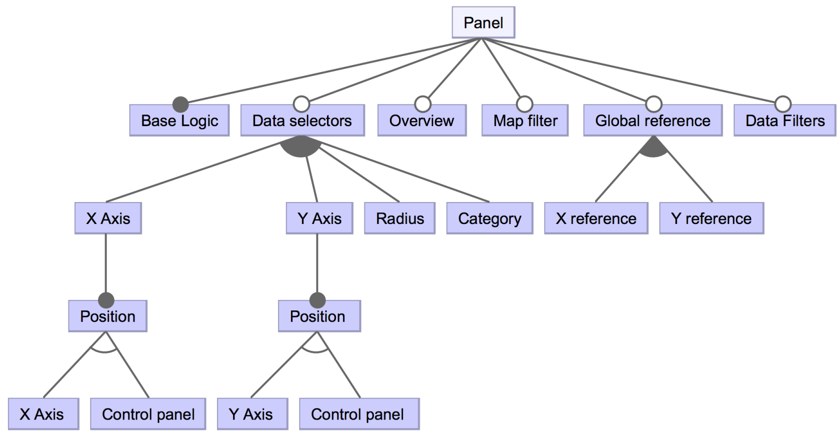


Figure 4 **High-level view of a component's panel subsequent features.** This part of the feature diagram shows lower-level features regarding the components' control panel.
Full-size DOI: 10.7717/peerjcs.203/fig-4

```
<xs:element name="ScreensConfig">
    <xs:complexType>
        <xs:choice>
            <xs:element name="PageGroup"...>
            <xs:element name="Page">
                <xs:complexType>
                    <xs:sequence>
                        <xs:element name="DataFilter" minOccurs="0">
                            <xs:complexType...>
                        </xs:element>
                        <xs:element name="Components"...>
                        <xs:element name="Layout" type="LayoutType"/>
                    </xs:sequence>
                    <xs:attribute name="page_id" type="xs:string"/>
                </xs:complexType>
                <xs:unique name="unique-page_id-2">
                    <xs:selector xpath="Page"/>
                    <xs:field xpath="@page_id"/>
                </xs:unique>
            </xs:element>
        </xs:choice>
    </xs:complexType>
</xs:element>
```

Figure 5 **Snippet of the DSL schema.** It is possible to specify the dashboard layout and its elements (i.e., data filters, components, etc.).
Full-size DOI: 10.7717/peerjcs.203/fig-5

a readable and easy-to-parse manner for injecting functionalities or requirements in a system, fostering flexibility since these rules are not directly defined (or hard-coded) in the source code.

The following examples describe the DSL developed for this work. Following the meta-model, every dashboard will be composed by one or more pages, each page with its configuration (i.e., layout and components, as seen in Fig. 5), and each page component with its setting (given the feature model, as seen in Fig. 6).

Data resources of each visual component are represented by the XSD generic type "*anyType*", to decouple the data structure and format from the presentation, and also to open up the possibility of injecting dynamic data sources without affecting the DSL syntax.

```
<xs:element name="Components">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="Component" maxOccurs="unbounded">
                <xs:complexType>
                    <xs:choice>
                        <xs:element name="ScatterDiagram"...>
                        <xs:element name="Heatmap"...>
                        <xs:element name="ChordDiagram"...>
                    </xs:choice>
                </xs:complexType>
            </xs:element>
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name="Layout" type="LayoutType"/>
```

Figure 6 **DSL schema regarding the specification of the dashboard components.** It is possible to see the link between the feature model elements and the XML schema elements (e.g., the components that could compose the dashboard).

Full-size DOI: 10.7717/peerjcs.203/fig-6

```
<xs:element name="ScatterDiagram">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="Title" type="xs:string"...>
            <xs:element name="Zoom" type="xs:string"...>
            <xs:element name="Exportation"...>
            <xs:element name="InitialData"...>
            <xs:element name="Controls" minOccurs="0"...>
            <xs:element name="Tooltip" minOccurs="0"...>
        </xs:sequence>
        <xs:attribute name="component_id"
                      type="xs:string"/>
    </xs:complexType>
</xs:element>
```

Figure 7 **DSL schema regarding the specification of the scatter diagram component.** This part of the DSL represents the available features for the scatter diagram component.

Full-size DOI: 10.7717/peerjcs.203/fig-7

In Figs. 6 and 7 the resemblance of the XML schema structure with the feature model can be appreciated. The hierarchical nature of XML matches with the hierarchical structure of feature diagrams. This resemblance allows better traceability of the features involved in the product line, because the syntax of the DSL is obtained from the feature model, thus providing a computer-understandable specification of the SPL, necessary to process the requirements and to automate the dashboard generation. In this current approach, the dashboard's feature model serves as documentation, but, as it will be discussed, it would be extremely valuable to create a programmatic link between this model and the DSL specification, in order to propagate and reflecting any feature model change automatically in the DSL, improving maintainability.

Finally, Fig. 8 shows how the layout of the dashboard is specified in terms of rows, columns and components (following, again, the meta-model previously presented). The

```xml
<xs:complexType name="LayoutType">
    <xs:choice>
        <xs:element name="RowGroup" type="LayoutType"/>
        <xs:element name="ColumnGroup" type="LayoutType"/>
        <xs:element name="Row" type="LayoutType" maxOccurs="unbounded"/>
        <xs:element name="Column" type="LayoutType" maxOccurs="unbounded"/>
        <xs:element name="Component">
            <xs:complexType>
                <xs:simpleContent>
                    <xs:extension base="xs:anySimpleType">
                        <xs:attribute name="ref" type="xs:string"/>
                    </xs:extension>
                </xs:simpleContent>
            </xs:complexType>
        </xs:element>
        <xs:element name="DataFilter">
            <xs:complexType>
                <xs:simpleContent>
                    <xs:extension base="xs:anySimpleType">
                        <xs:attribute name="ref" type="xs:string"/>
                    </xs:extension>
                </xs:simpleContent>
            </xs:complexType>
        </xs:element>
    </xs:choice>
    <xs:attribute name="width" type="xs:string" use="optional"/>
    <xs:attribute name="height" type="xs:string" use="optional"/>
</xs:complexType>
```

Figure 8 XML type for specifying the dashboard's layout. The dashboard layout (previously modeled through the dashboard meta-model) is specified by creating a custom type.
Full-size DOI: 10.7717/peerjcs.203/fig-8

DSL combines both the meta-model and feature model designs to obtain a specific syntax to configure all the aspects regarding the generation of final products.

The whole schema for the DSL can be consulted at the following GitHub repository https://github.com/AndVazquez/dashboard-spl-assets (*Vázquez-Ingelmo, 2018*).

Code generator

To put together all the developed assets and concepts, a code generator has been developed to manage the generation of functional dashboards. The generator interprets the DSL (i.e., XML configuration files) and selects the appropriate template (i.e., core assets of the SPL) to configure them by injecting the chosen features, obtaining the dashboards' final source code. The code templates and XML configuration files are managed by the developers following the elicited user requirements.

The inputs and outputs of the code generator can be seen in Fig. 9.

Code templates

The next challenge regarding the implementation of this SPL involves the choice of the techniques for materializing the product line's variability points. In this case, personalization is focused on the visual elements of the system's presentation layer, which require fine-grained variability (*Kästner & Apel, 2008*). Coarse-grained variability involves the addition and removal of full components, which is also useful for this approach (users may prefer a scatter diagram over a chord diagram to achieve their goals, removing the last from the dashboard). However, visual components themselves (referring to the elements that compose them) also require high variability to fit into different requirements,

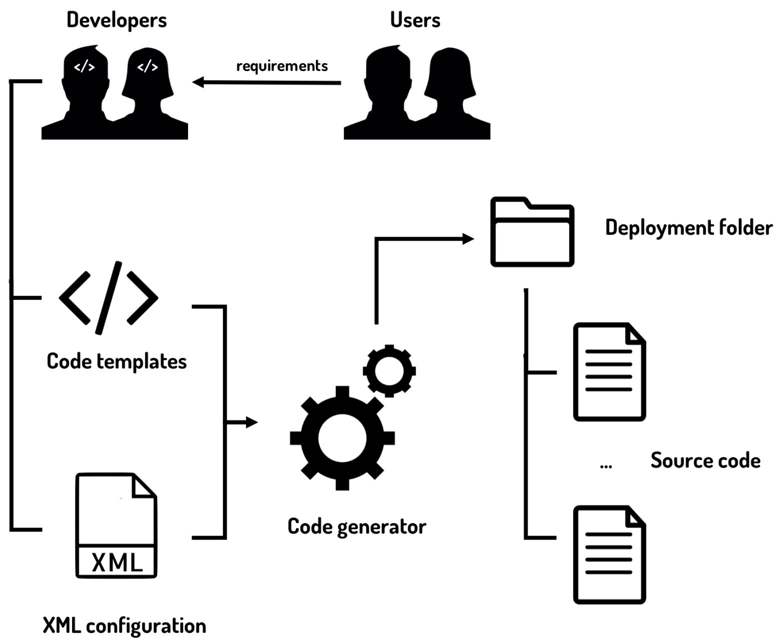


Figure 9 Code generator inputs and outputs. The code generator is fed with the code templates and the XML configuration files to provide the final source code of the dashboard.
Full-size DOI: 10.7717/peerjcs.203/fig-9

so fine-grained variability needs to be accomplished. There exist different approaches to implement fine-grained software composition, as in the case of FeatureHouse (*Apel, Kastner & Lengauer, 2009*), which uses superimposition and feature structure trees (FSTs), however, not every method supports the currently required granularity, which involves even statement-level variability. Fine granularity often prohibits superimposition approaches (*Apel, Kästner & Lengauer, 2013*).

The mechanism chosen to reach the desired feature granularity is based on template engines. Template engines allow to tag sections and parameterize units of source code to inject concrete values later and obtain complete source files. This mechanism accomplishes the necessity of materializing the variable features of a tangible product of the line.

Jinja2 (*Ronacher, 2008*) was selected as the engine for developing the core assets of this SPL. This template engine allows the definition of custom tags, filters and even macros, being the last one of the essential features to organize the core assets. As described in (*Kästner & Apel, 2008*), fine-grained approaches can make the source code tedious to read and maintain. By declaring every variant feature on different macros to compose them subsequently, it is possible to achieve high cohesion and loose coupling on the SPL feature implementation process, improving reusability and source code organization by grouping the different functionalities by its parent feature. There was no need to implement extensions of the Jinja2 implementation and mechanisms, as its current syntax was sufficient for the annotative approach followed.

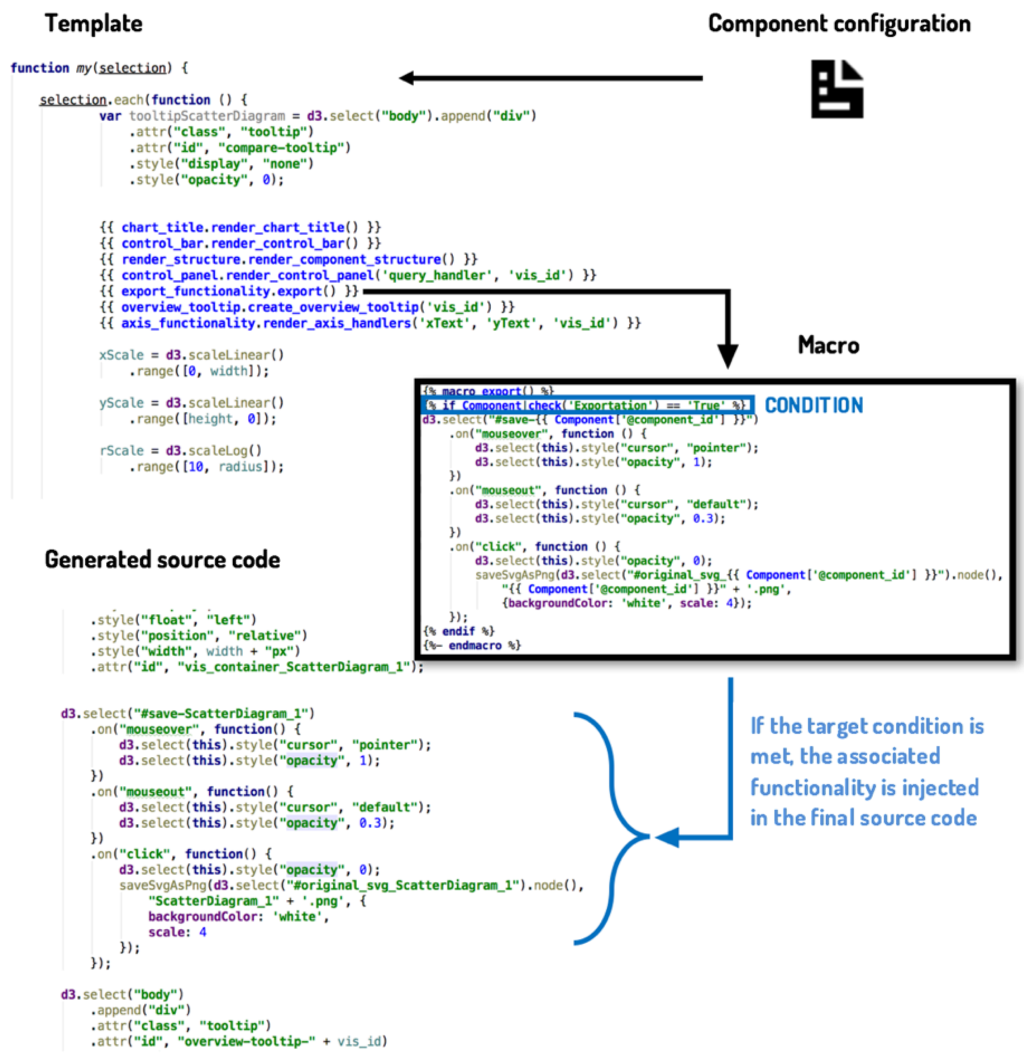


Figure 10 Workflow of the code generation process. A simplified view of the code generator behavior.
Full-size DOI: 10.7717/peerjcs.203/fig-10

A diagram of the detailed workflow for generating the source code can be seen in Fig. 10. The code templates for this case study can be also consulted at https://github.com/AndVazquez/dashboard-spl-assets (*Vázquez-Ingelmo, 2018*).

RESULTS

Generated dashboards

As it has been already introduced, the Observatory collects important datasets to research the employability and employment of graduates from Spanish universities. Relying on a customizable exploratory tool would increase the chances of discovering interesting patterns or relations within these complex fields. The dashboards of this case study have a series of particular requirements due to the data domain and the specific characteristics of the Observatory studies. For instance, the developed data visualizations exploit different

Figure 11 Results derived from the first configuration. Through this configuration is possible to apply different filters simultaneously to each scatter diagrams to observe how patterns evolve.
Full-size DOI: 10.7717/peerjcs.203/fig-11

dimensions of the Observatory's collected variables. Also, the generated Observatory's dashboards needed to be connected to the organization's GraphQL API (*Facebook, 2016*) that allow users to retrieve data statistics on demand (*Vázquez-Ingelmo, Cruz-Benito & García-Peñalvo, 2017*; *Vázquez-Ingelmo, García-Peñalvo & Therón, 2018a*; *Vázquez-Ingelmo, García-Peñalvo & Therón, 2018b*; *Vázquez-Ingelmo, García-Peñalvo & Therón, 2018c*), decoupling the data resources from the visual components' logic.

In this section, the results derived from the application of the presented dashboard product line within the university employment and employability domain are described. By tuning SPL through particular configurations, it is possible to obtain tailored solutions for different requirements and tasks.

Configuration #1. Comparison of different values is one of the most relevant tasks regarding the exploration of university employability and employment data. These comparisons could enlighten which factors affect employability and employment to a greater or lesser extent, leading to the possibility of conducting deeper analyses.

For example, by configuring a dashboard with two scatter diagrams side by side, it is possible to apply different filters to each one and observe how data patterns evolve (Fig. 11). Also, adding the global reference feature to both diagrams helps to make comparisons by adding a reference line marking the unfiltered and disaggregated values.

It is possible to appreciate, for example, that men graduates are more optimistic when commenting opinions about their future wages and the possibility of developing a working career in Spain (*Michavila et al., 2018b*). However, these diagrams also allow seeing at a glance that Arts and Humanities and Sciences graduates are more pessimistic about their future than their counterparts in other branches of knowledge, which are more clustered. For instance, only 40% of Sciences women graduates think that they could have a working career in Spain within five years.

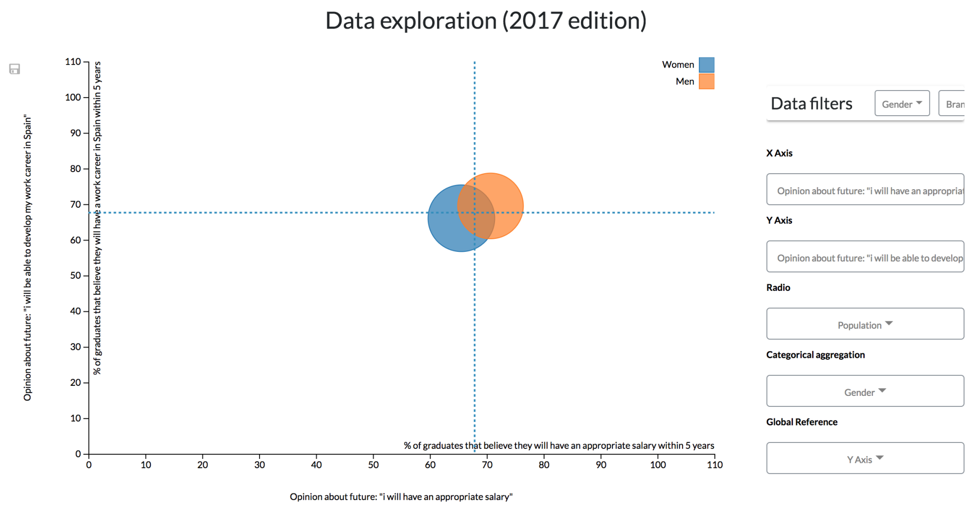


Figure 12 Results derived from the second configuration. The scatter diagram shows the link between different students' opinions classified by gender.

Full-size DOI: 10.7717/peerjcs.203/fig-12

This configuration enables the user to explore data through the combination of different aggregations, variables and filters.

Configuration #2. The previous configuration, however, could be complex for some users by having to control two diagrams at the same time to align different factors. A single scatter diagram could be added to the dashboard to drill-down data. It is possible to add another dimension to the scatter diagram component by mapping numerical variables through the radius of the visualization's data points.

For instance, following the same example of the first configuration, the differences between male and females can be observed by a gender aggregation of the data. In this case, the population of each group is mapped through the radius of the points (Fig. 12).

However, to see how the branch of knowledge affects the value of these variables, similarly to the previous configuration, it is necessary to continuously filter data by every single branch (Fig. 13). This configuration is then not recommendable when continuous, and more complex comparisons (such as the one made in the previous scenario) are required.

If, on the other hand, data exploration is not continuously required by a user, the controls could be allocated within a top bar (Fig. 14) that can be hidden to give more space to the visualizations.

Configuration #3. On the other hand, different pages focused on different data variables or data dimensions could be configured. This functionality allows freedom when arranging the content of the dashboards' pages to make it understandable for every particular user.

In the Observatory's case, a user might prefer having the dashboard screens organized by the study edition, being able to navigate through them thanks to a navigation bar (Fig. 15).

Or if preferred, it could be specified that each page will exploit a different set of data variables; for example, having a single tab to explore the students' competences (Fig. 16).

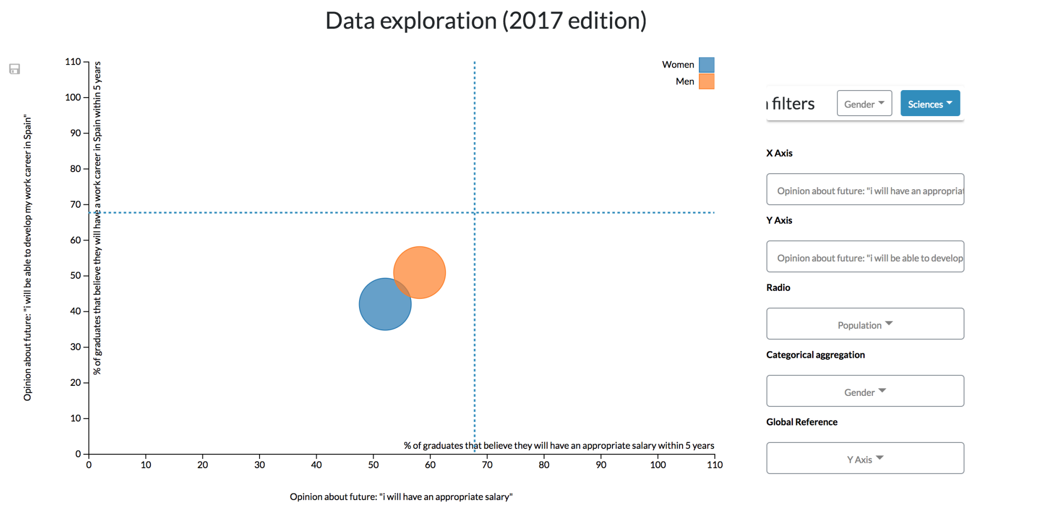


Figure 13 **Results derived from the second configuration.** The scatter diagram shows the link between different students' opinions classified by gender and filtered by the branch of knowledge, showing only the results related to Science students.

Full-size DOI: 10.7717/peerjcs.203/fig-13

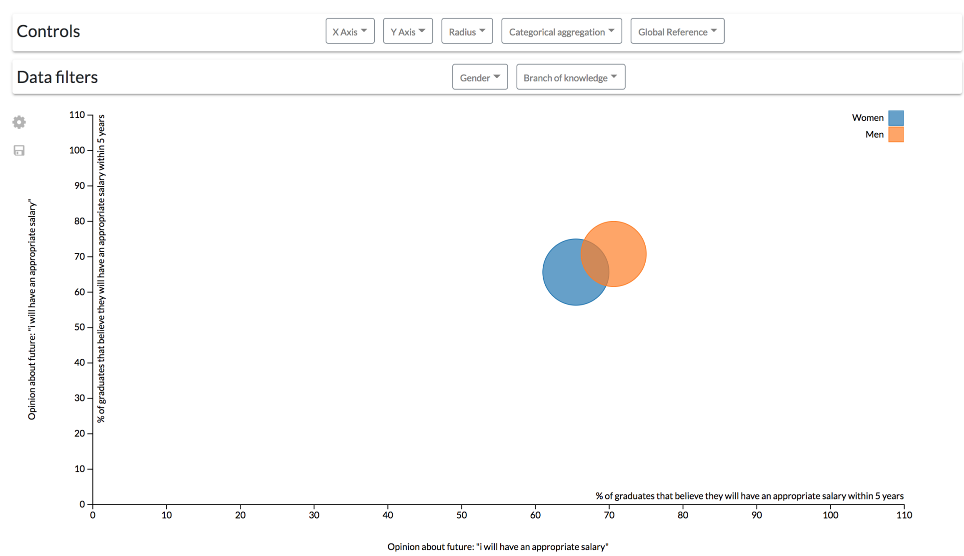


Figure 14 **Modification of the second configuration to change the controls location.** The controls for the scatter diagram are arranged in a bar on top of the visualization.

Full-size DOI: 10.7717/peerjcs.203/fig-14

Through this view it is possible to see a misalignment between the perceived level that the graduates have about their skills and the perceived level of contribution of the studies regarding the acquisition of that skills, and also between that possessed level and the perceived required level in their job positions (*Michavila et al., 2018b*).

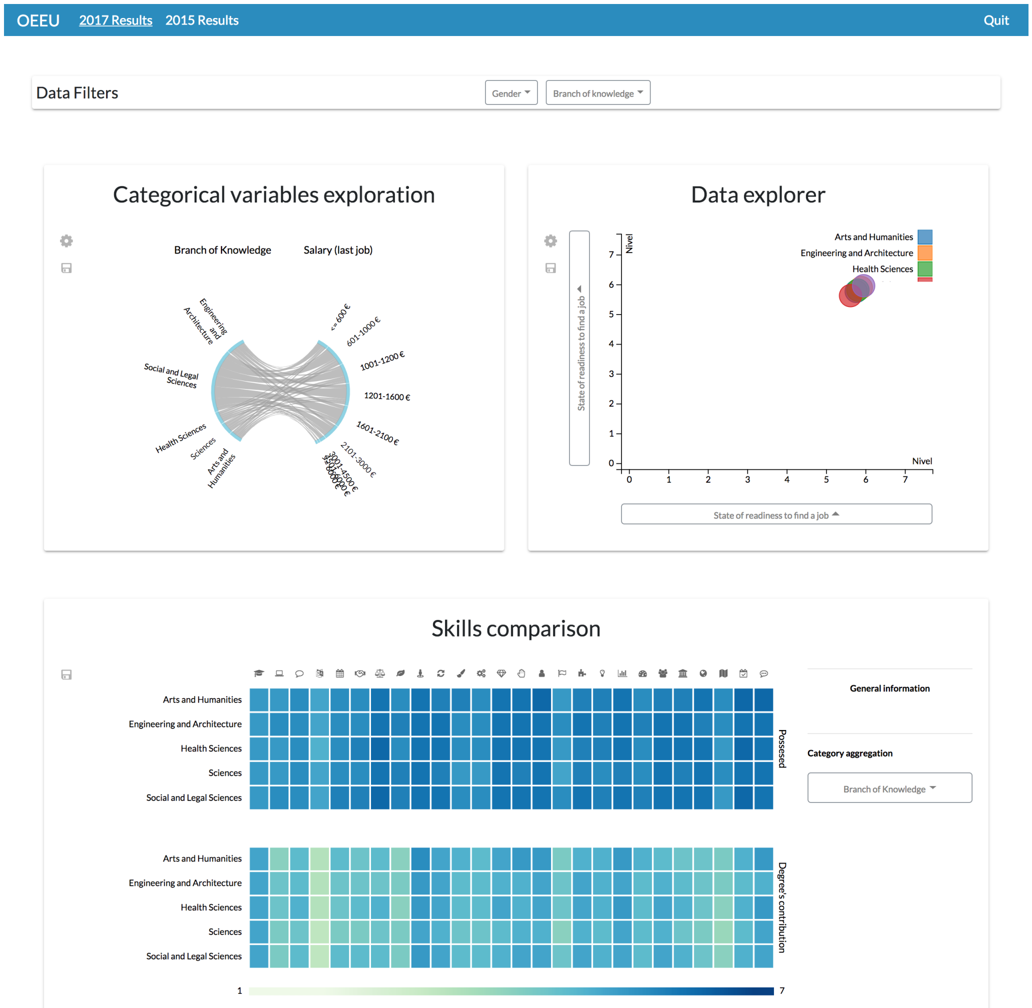


Figure 15 Dashboard involving different information visualizations. By specifying the layout of the dashboard it is possible to achieve dashboards with different components, each one with its own features.
Full-size DOI: 10.7717/peerjcs.203/fig-15

The previous dashboards are a quite tiny set of the available combinations that can be achieved through the SPL configuration, but they should serve as an example to show the possibilities of having a framework for generating personalized dashboards.

Product metrics

The metrics for the SPL are the following regarding its feature model:

- Feature model height: 9
- Features: 146
- Optional features: 106

The number of valid configurations has been omitted, given the recursion of the dashboards' composition (as highlighted in the dashboard meta-model), so infinite valid configurations can be generated.

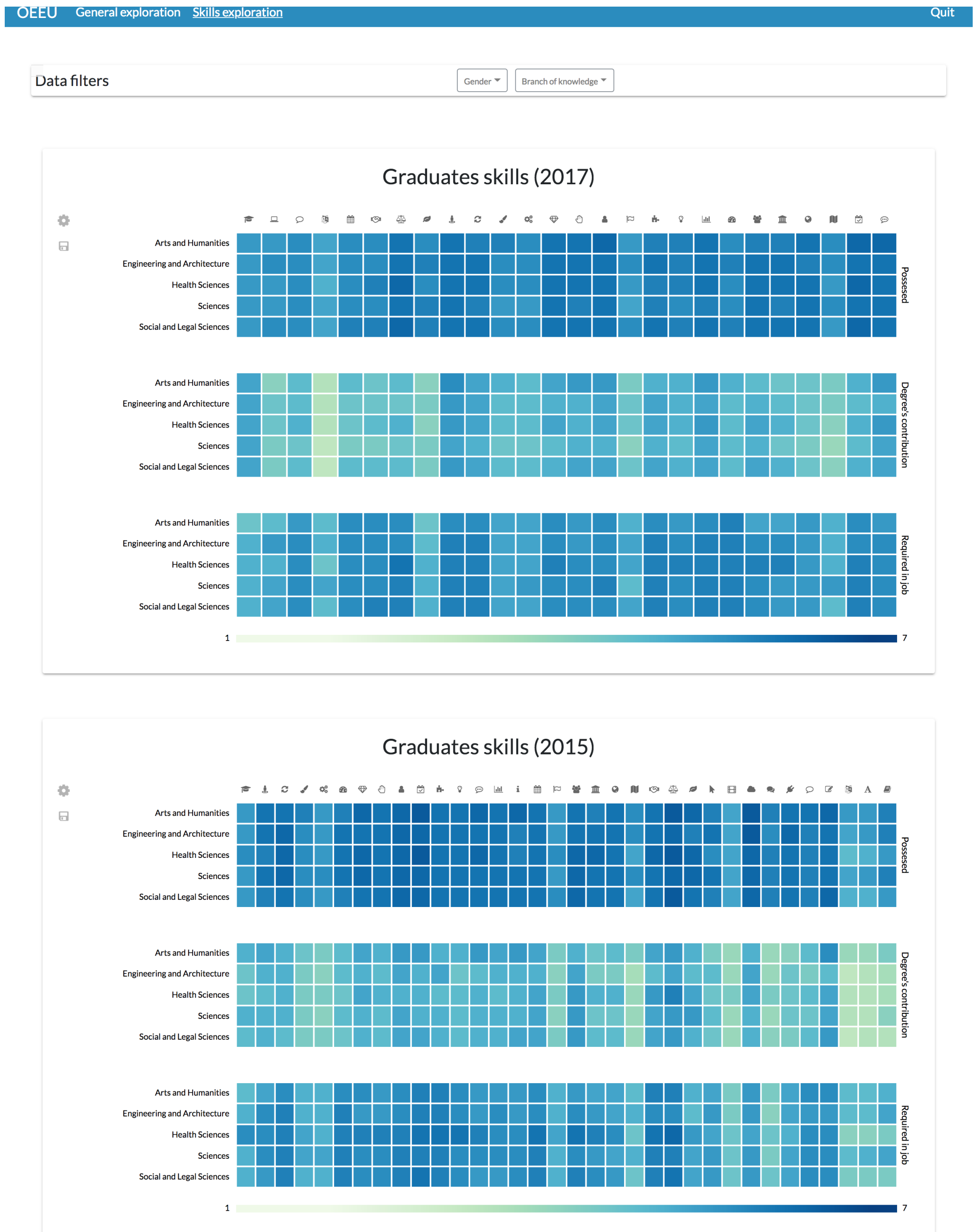


Figure 16 **Possible layout configuration for comparing students' skills through different study editions.** This configuration can be useful to identify lack of skills at-a-glance or their evolution through time.
Full-size DOI: 10.7717/peerjcs.203/fig-16

Regarding the core-assets (i.e., the templates' source code), the following metrics have been calculated (*El-Sharkawy, Yamagishi-Eichler & Schmid, 2018*):

- Lines of feature code (LoF): 2,638 lines of feature code. This metric is the addition of every line of code affected by any Jinja2 directive (i.e., every annotated line of code). It is a size metric that gives a high-level view about the source code associated to the SPL features.

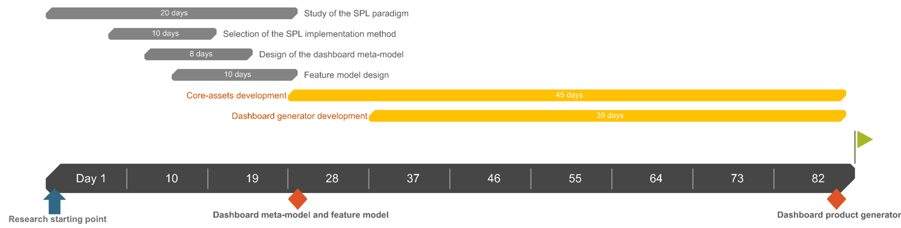


Figure 17 **Simplified Gantt diagram of the SPL development.** The Gantt diagram shows each task regarding the SPL development including its contextualization and design.

Full-size DOI: 10.7717/peerjcs.203/fig-17

- Fraction of annotated lines of code (PLoF): 48.39%. This is a variability density metric showing that the SPL's products have a 51.61% of common code (2,814 lines of code are not annotated).
- Scattering of variation points: this metric counts the number of times that a feature appears in the code (i.e., appears in a Jinja2 condition directive). High scattering values decreases the readability of the code. By refactoring the code into macros that contain all code associated to a specific feature, the scattering is reduced.

Given the complex domain in which the product line has been applied (i.e., the dashboards' domain), the scattering of the variation points was one of the main concerns, as high scattering would make the code even more complex. That was the reason to arrange the feature code into macros as a solution to address the scattering of variability points.

Development time improvement

The development of the presented SPL, including its conceptualization and design, took 82 days, as illustrated through a simplified Gantt diagram in Fig. 17. The core assets development task includes all the artifacts regarding the SPL (i.e., the DSL, the templates, etc.).

Before implementing this approach, a dashboard template with the same components and KPIs was the solution to offer all the results held in the Observatory's study, so universities could compare their individual results with the global, aggregated results. The development of the mentioned dashboard template took 15 days. However, this static approach limited universities to freely explore their data, as mentioned in other sections.

Five of the 50 universities were interviewed to capture their dashboard requirements and to estimate the elicitation process time consumption. However, this estimation should be considered as speculative given the variability of the complexity of the elicitation process, and especially, given the number of different universities (i.e., users) involved. Nevertheless, the requirement elicitation took one day for the interviewed universities.

Given the project's potential continuity, the dashboard implementation process would mainly consume time regarding requirements elicitation by using the presented SPL approach, decreasing the time spent on development processes. Without this approach, the information dashboards implemented for future Observatory's employability study editions would remain static and generalized for each involved user.

Building a personalized dashboard consume resources in terms of requirement elicitation and design, but also in terms of implementation or development. If the development phases are automated, then the main benefit is not only decreasing the development time of individual dashboards, but also, if necessary, devoting more time to the requirements identification and design phases, which, in the end, are the backbone of well-constructed dashboards. That is why, although significant time was consumed for the implementation of the dashboard SPL (82 days), it can be seen as an investment for the future, specifically in environments where significant quantities of user profiles are involved.

DISCUSSION

The application of domain engineering and the SPL paradigm to identify and factorize information dashboard functionalities has shown its usefulness to generate different dashboards with a set of common assets through the study of the dashboards' domain. The obtained results are fairly valuable, and open new paths for applying this approach to other data domains with new requirements.

Dashboards are complex software solutions that could be highly beneficial when adequately designed and tailored for specific users. These products can support decision-making processes, assisting visual analysis by presenting information in an understandable manner. However, the variety of profiles involved in these processes and their different definitions of "understandable" makes the implementation of dashboards a time- and resource-consuming task, since a dashboard configuration that is highly useful for one user could be pointless for the rest of them. What is more, dashboards can be composed of several elements, from simple visualizations to different linked views, cross-filtering capabilities, interaction methods, handlers, etc., thus making the dashboards' domain a complex domain not only because of the different profiles of potential users, but because of the great quantity of feasible combinations of these "dashboard elements" to build a proper solution. In addition, these features can be very fine-grained; in user-centered systems, a slight modification on visualization types, interaction patterns, layouts, color palettes, etc. could be crucial regarding the final perceived usability of the product.

Relying on a framework to easily generate information dashboards would allow stakeholders to focus on the information requirements and their refinement to provide better results when seeking valuable insights on large datasets. Also, it opens up the possibility to automatically adapt the dashboards' configurations to match dynamic requirements based on the device used (*Cruz-Benito et al., 2018b*) or other factors.

The factorization of the dashboards' components into individual features allow fine-grained reusability and a set of customization options. This fine-grained customization enables the possibility of having highly functional and exploratory-centered visualizations as well as more basic visual components more centered on the explanation of insights through the addition or removal of low-level features. The achieved granularity provides a foundation to develop not only whole visualization components, but also new interaction methods and design features that can be easily interchangeable to fulfill particular sets of user requirements.

An annotative method of implementation was undertaken using macros to encapsulate individual functionalities. This method takes all the benefits from the annotative approach (fine-grained customization) and avoids its code verbosity and scalability issues by dividing the core assets into base templates and macros (*Kästner & Apel, 2008*). Although there were other possibilities to implement the variability points, such as superimposition approach (which did not fulfilled the requirements for performing this approach, as discussed in the Materials & Methods section) like the FeatureHouse framework (*Apel, Kastner & Lengauer, 2009*; *Apel & Lengauer, 2008*) or the XVCL (*Jarzabek et al., 2003*) mechanism (which fits the feature granularity requirements of this domain), the final decision of using a templating engine allowed the direct connection of the designed DSL with the final source code, providing a higher level language to specify the dashboards' features, as well as the possibility of organizing the variability points into macros to increase readability, traceability and maintainability by having all the code associated to a feature in the same source file.

The chosen technology to implement the DSL was XML. The decision of implementing directly the DSL in XML technology was made because of the hierarchical nature of XML, and its resemblance to the hierarchical structure of the feature diagram, thus being the designed DSL a computer-understandable "translation" of the feature model for the dashboard generator to process. However, this language could be not as human-readable as other DSL solutions, generating issues if a non-expert user wants to specify its dashboards requirements by himself. Creating a friendly user interface to allow the dashboards' feature selection without involving direct manipulation of the XML files can be a valuable solution to address these issues and ease the product configuration process in the future.

Customization at functionality level has illustrated to be straightforward, as it is possible to easily vary the behavior of the visual components through the DSL. Visual design attributes customization, however, needs to be faced more deeply, as only the layout composition can be specified in detail at the moment. The visual customization challenge cannot be overlooked since dashboards not only have to provide valuable functionality; they should offer that functionality through a pleasant and usable interface (*Few, 2006*; *Sarikaya et al., 2018*).

On the other hand, this work has addressed customization focused on the presentation layer of dashboards, but with the SPL paradigm, architectural design can also be customized in order to provide different functional features regarding data processing, interoperability, storage, performance, security, etc., achieving a complete customizable dashboard solution, not only focusing on the visual presentation.

Regarding data acquisition, the developed tool was integrated with the Observatory's GraphQL API to provide dynamic data exploration. The connection to this particular type of data source involved the implementation of specific connectors to decouple the visualizations from the particular source. The variability of data sources is another identified challenge to be addressed through this approach, to support different data formats or data structures. Although counting on a GraphQL API facilitated the data retrieval by the unification of data requests, it is essential to enable the specification of other data retrieval methods.

Product metrics showed that significant feature code was needed to address high customizability of the dashboards (48.39% of the source code was annotated). Also, arranging the feature code into macros helped to increase features' traceability as well as to decrease the scattering of the variability points throughout the code, making the code more readable and maintainable.

The approach can decrease the development time of individualized dashboards for each involved university. As presented in the results section, the SPL not only offered space for development time improvements, but also enabled the capacity of offering customized solutions, which was previously regarded as unviable given the time constraints of the Observatory's project. Embracing the SPL paradigm can be seen as an investment for the future for projects with a common domain and with continuity over time.

Finally, it is clear that interesting patterns can be discovered thanks to the application of this dashboard SPL on the employability and employment fields. The Observatory's data provide a great context to perform more advanced analyses to enlighten this complex domain.

Having powerful visualization tools allow reaching insights about patterns or factors to guide the execution of more complex analyses and make decisions about the actions to take or the future research directions, like developing machine learning (ML) models (*García-Peñalvo et al., 2018*). Regarding this last field, having visualization tools to explore the input data before training any ML model could help to build better and more accurate models through an appropriate feature selection phase guided by the previously reached insights (*Hall, 1999*).

The main weaknesses and limitations of this solution come from the preliminary nature of the framework; it is crucial to further validate the usability of the automatically generated products to show their usefulness to the main beneficiaries of the dashboards: the users, as well as assess its implementation in other domains. The approach needs to be further generalized to provide a more versatile method and to match also development requirements (available technology or preferred programming languages), although results seem promising. Automating the generation of dashboards given their goal, their context, their end users, etc. could be extremely beneficial due to the vast potential of impact that these tools have (*Sarikaya et al., 2018*).

CONCLUSIONS

A domain engineering approach has been applied to the dashboards' domain to obtain a SPL of this type of software solution. By the identification of commonalities and variability points, a dashboard meta-model has been developed as well as a feature model to capture the different customization dimensions of the SPL.

The SPL has been developed through an annotative approach using code templates and macros (forming the core assets of the family of products). A DSL has been designed to facilitate and automate the application engineering process. The configuration files based on the DSL feed a code generator in charge of adding or removing the product features. The presented approach was applied within the Spanish Observatory for

University Employability and Employment system, to provide a variety of dashboard configurations that enable the exploitation and exploration of different dimensions regarding employability and employment data.

Future research lines will involve refinements of the meta-model and the DSL, usability testing of the obtained products and A/B testing (*Cruz-Benito et al., 2018a*; *Cruz-Benito et al., 2018b*; *Kakas, 2017*; *Siroker & Koomen, 2013*) on different configurations. Architectural customization could be supported to add more coarse-grained features like a visualization recommendation engine (*Gotz & Wen, 2009*; *Vartak et al., 2017*; *Voigt et al., 2012*), interface language translation or data preprocessing techniques before its presentation. Finally, the customization levels of the dashboards' visual design and data sources need to be further addressed.

ADDITIONAL INFORMATION AND DECLARATIONS

Funding

This work was supported in part by the Spanish Government Ministry of Economy and Competitiveness throughout the DEFINES project (Ref. TIN2016-80172-R), in part by the PROVIDEDH project, funded within the CHIST-ERA Programme under the national grant agreement: PCIN-2017-064 (MINECO, Spain) and in part by La Caixa Foundation. The work of A Vázquez-Ingelmo was supported by the Spanish Ministry of Education and Vocational Training under an FPU fellowship (FPU17/03276). There was no additional external funding received for this study. The funders helped with the contextualization of the research.

Grant Disclosures

The following grant information was disclosed by the authors:
Spanish Government Ministry of Economy and Competitiveness: Ref. TIN2016-80172-R.
CHIST-ERA Programme: PCIN-2017-064.
La Caixa Foundation.
Spanish Ministry of Education and Vocational Training: FPU17/03276.

Competing Interests

The authors declare there are no competing interests.

Author Contributions

- Andrea Vázquez-Ingelmo conceived and designed the experiments, performed the experiments, analyzed the data, contributed reagents/materials/analysis tools, prepared figures and/or tables, performed the computation work, authored or reviewed drafts of the paper.
- Francisco J. García-Peñalvo and Roberto Therón conceived and designed the experiments, performed the experiments, analyzed the data, contributed reagents/-materials/analysis tools, performed the computation work, authored or reviewed drafts of the paper, approved the final draft.

Data Availability

The following information was supplied regarding data availability:

Data and assets are available at https://github.com/AndVazquez/dashboard-spl-assets. (DOI: 10.5281/zenodo.1478134).

REFERENCES

Albright SC, Winston W, Zappe C. 2010. *Data analysis and decision making*. Mason: Cengage Learning.

Almakky H, Sahandi R, Taylor J. 2015. The effect of culture on user interface design of social media—a case study on preferences of saudi arabians on the arabic user interface of facebook. *World Academy of Science, Engineering Technology International Journal of Social, Behavioral, Educational, Economic, Business Industrial Engineering* **9**:107–111 DOI 10.5281/zenodo.1337791.

Álvarez JM, Evans A, Sammut P. 2001. Mapping between levels in the metamodel architecture. In: *International conference on the unified modeling language*. Berlin, Heidelberg: Springer, 34–46 DOI 10.1007/3-540-45441-1_4.

Apel S, Kastner C, Lengauer C. 2009. FEATUREHOUSE: language-independent, automated software composition. In: *Proceedings of the 31st international conference on software engineering*. Washington, D.C.: IEEE Computer Society, 221–231.

Apel S, Kästner C, Lengauer C. 2013. Language-independent and automated software composition: the FeatureHouse experience. *IEEE Transactions on Software Engineering* **39**:63–79 DOI 10.1109/TSE.2011.120.

Apel S, Lengauer C. 2008. Superimposition: a language-independent approach to software composition. In: *International conference on software composition*. Berlin, Heidelberg: Springer, 20–35 DOI 10.1007/978-3-540-78789-1_2.

Bray T, Paoli J, Sperberg-McQueen CM, Maler E, Yergeau F. 1997. Extensible markup language (XML). *World Wide Web Journal* **2**:27–66.

Card M. 1999. *Readings in information visualization: using vision to think*. San Francisco: Morgan Kaufmann.

Chadha D, Toner J. 2017. Focusing in on employability: using content analysis to explore the employability discourse in UK and USA universities. *International Journal of Educational Technology in Higher Education* **14**:33 DOI 10.1186/s41239-017-0071-0.

Clark S. 2018. Render your first network configuration template using Python and Jinja2. *Available at https://blogs.cisco.com/developer/network-configuration-template*.

Clements P, Northrop L. 2002. *Software product lines*. Boston: Addison-Wesley.

Cruz-Benito J, Sánchez-Prieto JC, Vázquez-Ingelmo A, Therón R, García-Peñalvo FJ, Martín-González M. 2018a. How different versions of layout and complexity of web forms affect users after they start it? A pilot experience. In: Rocha Á, Adeli H, Reis LP, Costanzo S, eds. *Trends and advances in information systems and technologies*. Cham: Springer, 971–979 DOI 10.1007/978-3-319-77712-2_92.

Cruz-Benito J, Vázquez-Ingelmo A, Sánchez-Prieto JC, Therón R, García-Peñalvo FJ, Martín-González M. 2018b. Enabling adaptability in web forms based on user characteristics detection through A/B testing and machine learning. *IEEE Access* **6**:2251–2265 DOI 10.1109/ACCESS.2017.2782678.

El-Sharkawy S, Yamagishi-Eichler N, Schmid K. 2018. Metrics for analyzing variability and its implementation in software product lines: a systematic literature review. *Information Software Technology* **106**:1–30 DOI 10.18420/se2019-53.

Elias M, Bezerianos A. 2011. Exploration views: understanding dashboard creation and customization for visualization novices. In: *IFIP conference on human-computer interaction*. Berlin, Heidelberg: Springer, 274–291 DOI 10.1007/978-3-642-23768-3_23.

Ezzat Labib Awad A. 2017. Enforcing customization in e-Learning systems: an ontology and product line-based approach. PhD thesis, Universitat Politécnica de Valéncia.

Facebook. 2016. GraphQL. *Available at https://facebook.github.io/graphql/* (accessed on 24 July 2017).

Fallside DC, Walmsley P. 2004. XML Schema part 0: Primer second version. Cambridge: W3C.

Few S. 2006. *Information dashboard design*. Sebastopol: O'Reilly Media.

Freeman G, Batory D, Lavender G. 2008. Lifting transformational models of product lines: a case study. In: *International conference on theory and practice of model transformations*. Berlin, Heidelberg: Springer, 16–30 DOI 10.1007/978-3-540-69927-9_2.

Gabillon Y, Biri N, Otjacques B. 2015. Designing an adaptive user interface according to software product line engineering. In: *The eighth international conference on advances in computer-human interactions*. Lisbon, Portugal, 86–91.

Gacek C, Anastasopoules M. 2001. Implementing product line variabilities. In: *ACM SIGSOFT software engineering notes*. New York: ACM, 109–117.

García-Peñalvo FJ. 2016. The third mission. In: *Education in the knowledge society*. Vol. 17. Salamanca: Universidad de Salamanca, 7–18.

García-Peñalvo FJ, Cruz-Benito J, Martín-González M, Vázquez-Ingelmo A, Sánchez-Prieto JC, Therón R. 2018. Proposing a machine learning approach to analyze and predict employment and its factors. *International Journal of Interactive Multimedia and Artificial Intelligence* **5(2)**:39–45 DOI 10.9781/ijimai.2018.02.002.

Gomaa H. 2004. Designing software product lines with UML: from use cases to pattern-based software architectures. Boston: Addison Wesley Longman Publishing Co., Inc.

Gómez A, Penadés MC, Canós JH, Borges MR, Llavador M. 2014. A framework for variable content document generation with multiple actors. *Information and Software Technology* **56**:1101–1121 DOI 10.1016/j.infsof.2013.12.006.

Gotz D, Wen Z. 2009. Behavior-driven visualization recommendation. In: *Proceedings of the 14th international conference on intelligent user interfaces*. New York: ACM, 315–324.

Hall MA. 1999. Correlation-based feature selection for machine learning. PhD thesis, University of Waikato, Hamilton, New Zealand.

Hauptmann B, Bauer B, Harhurin A, Pleuss A. 2010. Supporting derivation and customization of user interfaces in software product lines using the example of web applications. Master's thesis, Technische Universität München.

Hillage J, Pollard E. 1998. *Employability: developing a framework for policy analysis*. London: DfEE London.

Jarzabek S, Bassett P, Zhang H, Zhang W. 2003. XVCL: XML-based variant configuration language. In: *Proceedings of the 25th international conference on software engineering*. Washington, D.C.: IEEE Computer Society, 810–811.

Kakas AC. 2017. A/B Testing. In: Sammut C, Webb GI, eds. *Encyclopedia of machine learning and data mining*. New York: Springer.

Kang KC, Cohen SG, Hess JA, Novak WE, Peterson AS. 1990. *Feature-oriented domain analysis (FODA) feasibility study*. Pittsburgh: Carnegie-Mellon University, Software Engineering Institute.

Kästner C, Apel S. 2008. Integrating compositional and annotative approaches for product line engineering. In: *Proc GPCE workshop on modularization, composition and generative techniques for product line engineering*. 35–40.

Kästner C, Apel S, Kuhlemann M. 2008. Granularity in software product lines. In: *Proceedings of the 30th international conference on software engineering*. New York: ACM, 311–320.

Kleppe AG, Warmer J, Bast W. 2003. *MDA explained. The model driven architecture: practice and promise*. Boston: Addison-Wesley Longman Publishing Co. Inc.

Kramer D, Oussena S, Komisarczuk P, Clark T. 2013. Using document-oriented GUIs in dynamic software product lines. In: *ACM SIGPLAN Notices*. New York: ACM, 85–94.

Logre I, Mosser S, Collet P, Riveill M. 2014. Sensor data visualisation: a composition-based approach to support domain variability. In: *European conference on modelling foundations and applications*. Berlin, Heidelberg: Springer, 110–116.

Marcus A, Gould EW. 2000. Crosscurrents: cultural dimensions and global Web user-interface design. *Interactions* **7**:32–46 DOI 10.1145/345190.345238.

Marinho FG, Lima F, Ferreira Filho JB, Rocha L, Maia ME, De Aguiar SB, Dantas VL, Viana W, Andrade RM, Teixeira E. 2010. A software product line for the mobile and context-aware applications domain. In: *International conference on software product lines*. Berlin, Heidelberg: Springer, 346–360.

Michavila F, Martínez JM, Martín-González M, García-Peñalvo FJ, Cruz-Benito J. 2016. *Barómetro de empleabilidad y empleo de los universitarios en España, 2015 (Primer informe de resultados)*. Madrid: Observatorio de Empleabilidad y Empleo Universitarios.

Michavila F, Martínez JM, Martín-González M, García-Peñalvo FJ, Cruz Benito J. 2018a. Empleabilidad de los titulados universitarios en España. Proyecto OEEU. *Education in the Knowledge Society* **19**:21–39 DOI 10.14201/eks20181912139.

Michavila F, Martínez JM, Martín-González M, García-Peñalvo FJ, Cruz-Benito J, Vázquez-Ingelmo A. 2018b. *Barómetro de empleabilidad y empleo universitarios. Edición Máster 2017*. Madrid, España: Observatorio de Empleabilidad y Empleo Universitarios.

Nascimento LMD. 2008. *Cores assets development in software product lines-towards a practical approach for the mobile game domain*. Riga, Latvia: Lambert Academic Publishing.

Pleuss A, Botterweck G, Dhungana D. 2010. Integrating automated product derivation and individual user interface design. In: *Fourth international workshop on variability modelling of software-intensive systems*. Linz, Austria.

Pleuss A, Hauptmann B, Dhungana D, Botterweck G. 2012a. User interface engineering for software product lines: the dilemma between automation and usability. In: *Proceedings of the 4th ACM SIGCHI symposium on Engineering interactive computing systems*. New York: ACM, 25–34.

Pleuss A, Hauptmann B, Keunecke M, Botterweck G. 2012b. A case study on variability in user interfaces. In: *Proceedings of the 16th international software product line conference-vol. 1*. New York: ACM, 6–10.

Pohl K, Böckle G, Van der Linden FJ. 2005. *Software product line engineering: foundations, principles and techniques*. New York: Springer-Verlag New York, Inc.

Quinton C, Mosser S, Parra C, Duchien L. 2011. Using multiple feature models to design applications for mobile phones. In: *Proceedings of the 15th international software product line conference, vol. 2*. New York: ACM, 23.

Ridge B, Gaspar T, Ude A. 2017. Rapid state machine assembly for modular robot control using meta-scripting, templating and code generation. In: *Humanoid Robotics (Humanoids), 2017 IEEE-RAS 17th International Conference*. Piscataway: IEEE, 661–668.

Ronacher A. 2008. Jinja2 Documentation. *Available at http://jinja.pocoo.org/docs/2.10/: Jinja2*.

Sarikaya A, Correll M, Bartram L, Tory M, Fisher D. 2018. What do we talk about when we talk about dashboards? *IEEE Transactions on Visualization Computer Graphics* **25**:682–692.

Sboui T, Ayed MB, Alimi AM. 2018. A UI-DSPL approach for the development of context-adaptable user interfaces. *IEEE Access* **6**:7066–7081 DOI 10.1109/ACCESS.2017.2782880.

Siroker D, Koomen P. 2013. *A/B testing: the most powerful way to turn clicks into customers*. Hoboken: John Wiley & Sons.

Tufte E, Graves-Morris P. 2014. *The visual display of quantitative information.; 1983*. Cheshire: Graphics Press.

Universities UK, Confederation of British Industry. 2009. *Future fit: preparing graduates for the world of work*. London: CBI, Universities UK.

Vartak M, Huang S, Siddiqui T, Madden S, Parameswaran A. 2017. Towards visualization recommendation systems. *ACM SIGMOD Record* **45**:34–39.

Vázquez-Ingelmo A. 2018. Code repository that supports the research presented in the paper Taking advantage of the software product line paradigm to generate customized user interfaces for decision-making processes: a case study on university employability. *Available at https://github.com/AndVazquez/dashboard-spl-assets*.

Vázquez-Ingelmo A, Cruz-Benito J, García-Peñalvo FJ. 2017. Improving the OEEU's data-driven technological ecosystem's interoperability with GraphQL. In: Dodero JM, Ibarra Sáiz MS, Ruiz Rube I, eds. *Fifth international conference on technological ecosystems for enhancing multiculturality (TEEM'17) (Cádiz, Spain, October 18–20 (2017)*. New York: ACM.

Vázquez-Ingelmo A, García-Peñalvo FJ, Therón R. 2018a. Application of domain engineering to generate customized information dashboards. In: *HCI International 2018 20th conference on human-computer interaction (15-20 2018)*. Las Vegas: Springer.

Vázquez-Ingelmo A, García-Peñalvo FJ, Therón R. 2018b. Domain engineering for generating dashboards to analyze employment and employability in the academic context. In: *6th international conference on technological ecosystems for enhancing multiculturality. Salamanca, Spain*. New York: ACM DOI 10.1145/3284179.3284329.

Vázquez-Ingelmo A, García-Peñalvo FJ, Therón R. 2018c. Generation of customized dashboards through software product line paradigms to analyse university employment and employability data. In: *Learning analytics summer institute Spain 2018—LASI-SPAIN 2018*. León, Spain.

Voelter M, Visser E. 2011. Product line engineering using domain-specific languages. In: *Software Product Line Conference (SPLC), 2011 15th International*. Piscataway: IEEE, 70–79.

Voigt M, Pietschmann S, Grammel L, Meißner K. 2012. Context-aware recommendation of visualization components. In: *The fourth international conference on information, process, and knowledge management (eKNOW)*. Citeseer, 101–109.

Yorke M. 2006. *Employability in higher education: what it is-what it is not*. Heslington, York: Higher Education Academy York.

Zhang H, Jarzabek S, Swe SM. 2001. XVCL approach to separating concerns in product family assets. In: *International symposium on generative and component-based software engineering*. Efurt, Germany: Springer, 36–47.

7.13 Appendix M. Dashboard meta-model for knowledge management in technological ecosystem: a case study in healthcare

proceedings

MDPI

Proceedings

Dashboard Meta-Model for Knowledge Management in Technological Ecosystem: A Case Study in Healthcare †

Andrea Vázquez-Ingelmo [1,*], Alicia García-Holgado [1], Francisco José García-Peñalvo [1] and Roberto Therón [2]

[1] GRIAL Research Group, Computer Science Department, University of Salamanca, 37008 Salamanca, Spain; aliciagh@usal.es (A.G.-H.); fgarcia@usal.es (F.J.G.-P.)

[2] VisUSAL, GRIAL Research Group, Computer Science Department, University of Salamanca, 37008 Salamanca, Spain; theron@usal.es

* Correspondence: andreavazquez@usal.es; Tel.: +34-923294500 (ext. 3433)

† Presented at the 13th International Conference on Ubiquitous Computing and Ambient Intelligence UCAmI 2019, Toledo, Spain, 2–5 December 2019.

Published: 20 November 2019

Abstract: Informal caregivers play an important role in healthcare systems in many countries. They have a high impact on reducing care costs related to dependent persons because their support prevents institutionalization. A technological ecosystem has been defined to support informal caregivers using psychoeducation techniques. This ecosystem should include a dashboard to support decision-making processes related to the wellbeing of patients and caregivers. A dashboard meta-model was used to obtain a concrete model for the presented context. This meta-model allows defining dashboards adapted to the users' needs and specific data domains.

Keywords: model-driven development; dashboard; meta-model; knowledge management; healthcare; technological ecosystem

---

1. Introduction

Nowadays, knowledge is the driver for development in any context. It has become the most important strategic factor in corporate operations [1] because it is associated with the capabilities of companies to achieve a competitive advantage [2,3]. Within an organization, knowledge is not only electronic or printed documents; the employees' own knowledge and the implicit knowledge within the organization's processes are part of the assets related to knowledge [4].

Knowledge management is one of the key priorities of institutions and companies; they invest significant resources in developing their capacity to share, create, and apply new knowledge to improve both their internal and external business processes [5]. On the other hand, according to Davenport et al. [6], the implementation of knowledge management processes within an organization could be expensive, meaning that not all organizations have the capability and the resources to profit from their knowledge.

These aims and problems are present in any organization, but gain importance when the organization is part of the healthcare system. In this context, knowledge management processes affect not only patients and their relatives directly or indirectly, but also healthcare professionals. In particular, there is a need to improve knowledge management processes related to dependent persons owing to the aging of the population, with a special emphasis in developed countries. The number of persons over 60 years is growing faster than all younger age groups [7], and these numbers have an impact on the cost of care and the resources needed for this population.

From a technological point of view, technological ecosystems have emerged as a solution focused on improving knowledge management in a heterogeneous context. They provide a general framework that allows defining and developing any type of technological solution in which data and information are the backbone of the problem [8–10]. In particular, they can be employed to manage knowledge related to patients with cognitive and physical impairments, in addition to the people involved in their care, with a particular focus on their caregivers, both formal (professionals that provide care to patients) and informal (persons that provide care to patients).

Technological ecosystems can combine different software components with a set of human elements, such as methodologies or management workflows. In previous works, a technological ecosystem for supporting informal caregivers has been proposed [11,12]. Among the software components, a dashboard is proposed. Dashboards are one of the most useful tools for generating knowledge about certain domains. Information dashboards allow the identification of patterns, outliers, relationships, and so on, fostering insight delivery through visual analysis [13,14].

However, dashboards need to be adapted to their audience [15], to specific data domains, and to the tasks that will be performed to analyze these data, among other factors. The main reason is that there is not a "one-size-fits-all" when referring to dashboards; users have different mental models [16], goals, experience, literacy, domain knowledge, and so on [17–26], making the design process of a dashboard a complex task, where the elicitation of requirements can be seen as the backbone process, as it will drive the subsequent phases and design decisions.

Although a complex process, the development of a tailored dashboard has huge benefits; if the dashboard is developed for a specific user and a specific dataset, it will match the identified necessities, making the dashboard more effective, comfortable, and usable for the user, thus fostering the generation of knowledge and properly supporting the user's decision-making processes.

However, in order to deliver tailored dashboards without consuming significant amounts of resources and time, this process must be automatized. By automatizing and creating a pipeline for generating dashboards, the complexity of matching particular requirements can be reduced, delivering effective dashboards adapted to a specific context (data, users, goals, and so on) in less time.

There are different methods to ease the adaptation of these tools to concrete contexts: configuration wizards (where the users can customize or build their dashboards through explicit interactions) [27], agents that retrieve user behaviour data and reconfigure the display accordingly [28], configuration files that hold the specific requirements and are rendered into a properly configured dashboard [29,30], and so on. However, one of the most powerful methods to enable the generation of concrete products given a set of requirements is model-driven engineering [31]. Model-driven paradigms leverage high-level models for instantiating concrete models. Therefore, abstracting the features of dashboards into a set of common characteristics can support a generative process through model mappings and instantiations of concrete dashboard products.

One of the main benefits of using a dashboard meta-model is the possibility of instantiating dashboards adapted to different contexts through a single artifact, not only fostering reuse at development phases, but also allowing knowledge reuse. This characteristic is very useful in the healthcare context; the technological ecosystem mentioned above should adapt to different healthcare scenarios, such as hospitals, nursing homes, or health centers.

The present work aims to define a dashboard based on the meta-model to support caregivers in order to be included in the eHealth technological ecosystem for caregivers.

The rest of the paper is organized as follows. Section 2 describes the methodology used to instantiate the dashboard from the meta-model. Section 3 presents the meta-model to define information dashboards. Section 4 describes the architecture of the technological ecosystem in which the dashboard will be included. Section 5 presents the dashboard for supporting caregivers. Finally, Section 6 summarizes the main conclusions of this work.

2. Methodology

Model-driven development (MDD) allows separating the data and the operations specification of the system from the technical details related to a specific program language or platform. The object management group (OMG) provides the model-driven architecture (MMA) as a proposal to implement MDD. It provides a framework for software development that uses models to describe the system to be built [32]. The main difference between MDD and MDA is that the OMG proposal uses a set of standards such as meta-object facility (MOF), unified modeling language (UML), XML (Extensible Markup Language) metadata interchange (XMI), and query/view/transformation (QVT).

The dashboard meta-model and the healthcare dashboard model are part of the four-layer meta-model architecture provided by the OMG, in which a model at one layer is used to specify models in the layer below [33]. In particular, the dashboard meta-model is an instance of MOF (i.e., an M2-model), and the healthcare dashboard is a model instantiated from this meta-model (i.e., an M1-model). The M0-model is not used because the implementation is not the object of study of the present work.

The healthcare dashboard is part of the technological ecosystem for caregivers, which is based on a meta-model defined and validated in previous works. The first version of the ecosystem meta-model is based on MOF, and the last validated version is an instance of Ecore [34]. Both versions are M2-models, and the model of the technological ecosystem for caregivers is an M1-model [12]. The model has served as a map to develop and deploy the ecosystem in a real context. The C4 model was used to represent the architecture of the final system. The C4 model was inspired by the UML and the 4+1 model for software architecture. It is a simplified version of the underlying concepts, designed to make it easier for software developers to describe and understand how a software system works and to minimize the gap between the software architecture description and the source code [35].

3. Dashboard Meta-Model

The dashboard meta-model was elaborated on, taking into account three main dimensions or elements: the user, the layout, and the components. The whole personalization process must be driven by end-users; this is why this entity is included in the meta-model. The characteristics that a user has (preferences, disabilities, knowledge about the displayed data's domain, visualization literacy, and bias) influence the dashboard components to match specific necessities.

On the other hand, users have a series of goals that they will try to accomplish through the information displayed on the dashboards. Goals can be decomposed into different low-level tasks, and the selected dashboard components must support the identified tasks to help users reach their goals (Figure 1). For more detail regarding this section of the meta-model, please refer to the work of [36].

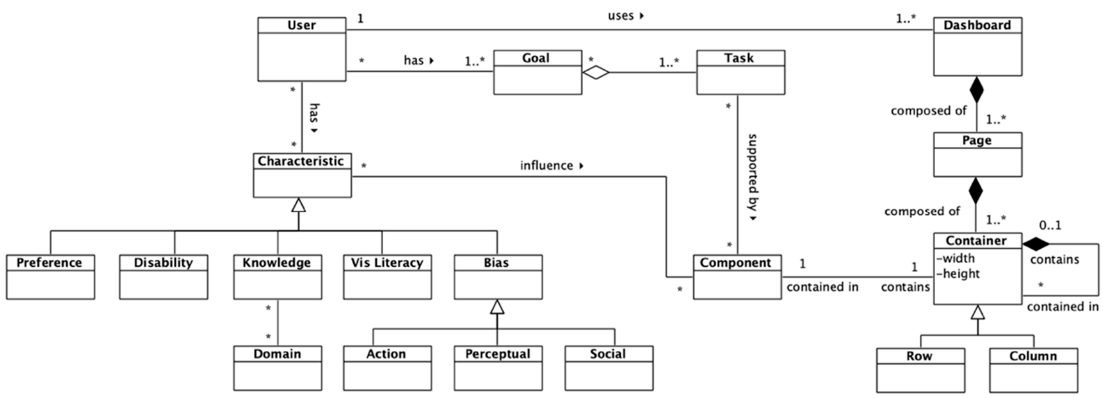


Figure 1. User and layout section of the dashboard meta-model proposal.

The purpose of the layout section of the dashboard is to model the generic structure of a dashboard. Dashboards can be composed of different pages, with different containers (rows or columns) that hold different components. This can be seen in the right section of Figure 1.

Regarding the components of the dashboard, several elements were identified. The main components of dashboards are the information visualizations that present the data, although dashboards can also contain controls (handlers, filters, and so on), graphic resources, or text. A visualization can be affected by global controls, as well as by "local" controls, that is, controls that only affect a single visualization.

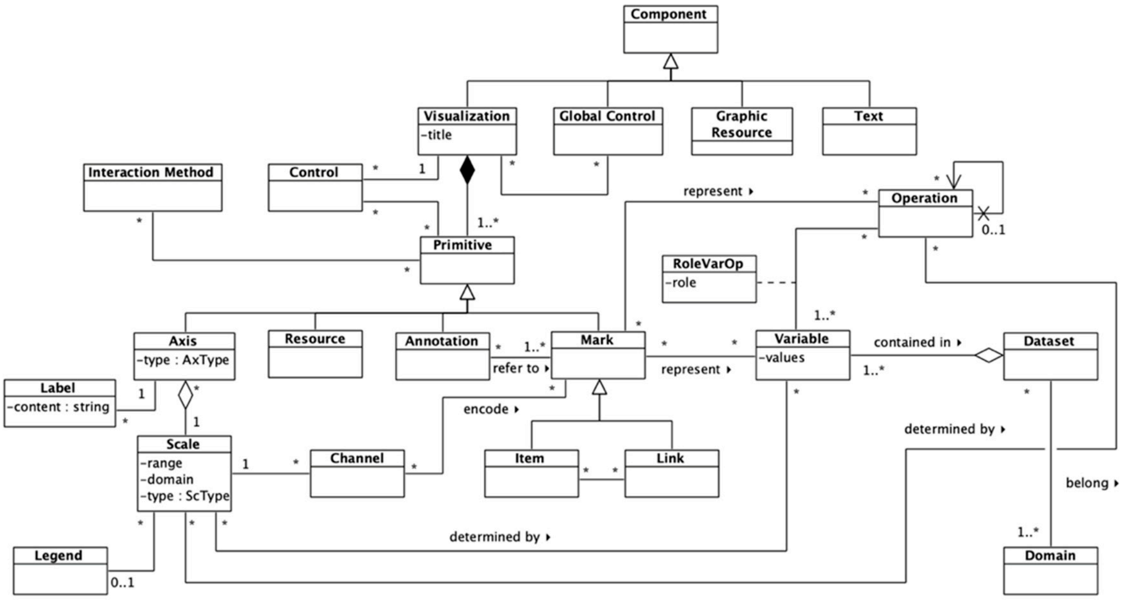


Figure 2. Components section of the dashboard meta-model proposal.

The *Primitive* class is a high-level class that encompasses basic elements for composing visualizations. These elements can be axes, annotations, marks, and resources (images, text, and so on). Interaction methods and local controls can modify these primitives to enable drill-down. Axes contain information about the scales, and thus about some channels that can influence a visual mark. An axis is always associated to a scale, and a single axis cannot represent more than one scale at once; however, a scale can be represented in several axes, if providing redundant information is necessary, for example.

On the other hand, there are popular terms to refer to data elements, but the most used among the literature are "marks" and "visual channels" or "visual encodings" [27,37–39]. Marks can be of two types: items or links. While items represent nodes, points, and so on, links represent connections, containments, and so on, among items [39]. Marks can hold not only raw information about data variables, but also about the results of operations, following the PTAH meta-model presented in the work of [40]. Different channels can be used to encode data values visually.

Channels are associated with a scale, which maps a variable's or operation's values to specific channel values, modifying the properties of visual marks (position, color, opacity, size, and so on). The *RoleVarOp* association class models the role of a variable within an operation. Also, a recursive relationship in the *Operation* class was included to support potential chained operations.

Finally, datasets can be related to different domains. This *Domain* class is connected to the previously described *Knowledge* meta-class shown in Figure 1 (user section of the meta-model). This association captures the familiarity or understanding that a user can have regarding the data being displayed on the dashboard.

The dashboard meta-model presented in Figures 1 and 2 is connected to the ecosystem meta-model (available https://doi.org/10.5281/zenodo.1066369). The ecosystem meta-model does not represent each software component in detail; it only provides a high-level abstraction of the ecosystem, in which software components are represented as black boxes. The dashboard, although it is one of the most important elements of a knowledge management ecosystem, is one of these black boxes because of its complexity. The dashboard meta-model provides the required details to model that part of the ecosystem. These details, however, are still at a M2-level following the OMG's four-layer architecture, because the dashboard meta-model is an instance of MOF (although the dashboard meta-model and the learning ecosystem meta-model are at different abstraction levels).

Moreover, some elements in the dashboard meta-model connect this proposal with the ecosystem meta-model. First, the users are an important part of both meta-models, the human factor has an important role, and the technology is defined and evolved to support their needs. On the other hand, the *Goals* in the dashboard meta-model are represented as *Objectives* in the ecosystem meta-model. These elements are translated into *Information Fl*ows as a way to model the interactions among components in the ecosystem meta-model, and into *Tasks* in the dashboard meta-model, as a way to define which components will be available in the dashboard and which interactions they will provide.

4. Healthcare Ecosystem for Caregivers

The aim of the technological ecosystem for caregivers is to support the learning and knowledge management processes to develop and enhance the caregiving competences both at home and in care environments of formal and informal caregivers [12]. In particular, the ecosystem allows providing psychoeducation [41] to dependent persons and informal caregivers in order to alleviate the physical and mental health problems that they suffer, such as work overload, depression, or anxiety.

The ecosystem makes it possible to provide remote access to different services. It is composed of a set of software components (Figure 3) and is based on a set of management and methodological input streams—a business plan, a training plan, and a medical protocol. First, it provides remote teaching–learning environments to support both informal and formal caregivers. Through *Discover*, psychoeducation is accessible to these profiles, so they can obtain answers to the questions that arise daily during their care duties and psychological support, as well as information, advice, and guidance.

Second, *SocialNet* is an online tool that provides a private social network composed of a set of private and safe areas, called walls, for each patient [42]. The main users are the relatives of the patients and their caregivers. In some cases, patients can also access *SocialNet* to publish their activities and view the contents published by their caregivers and relatives. Finally, the caregivers' managers can access the social network, but only to manage which caregivers control a patient's wall (this relationship is not represented in Figure 3 to avoid lines crossing the whole system).

Third, the *dashboard* is a tool to support decision-making processes. In particular, it is focused on supporting caregivers' managers to make decisions about the workload of the caregivers, as well as the activity of the patients based on the insights from the different components of the ecosystem.

Moreover, two software components provide support to other components, the *User Manager* that centralizes the users' data management and the access, and a tool to support the analysis of the data from *Discover*, *SocialNet*, and an external database with medical and personal information about the patients. The *Data Analysis Support* provides the datasets for the dashboard component.

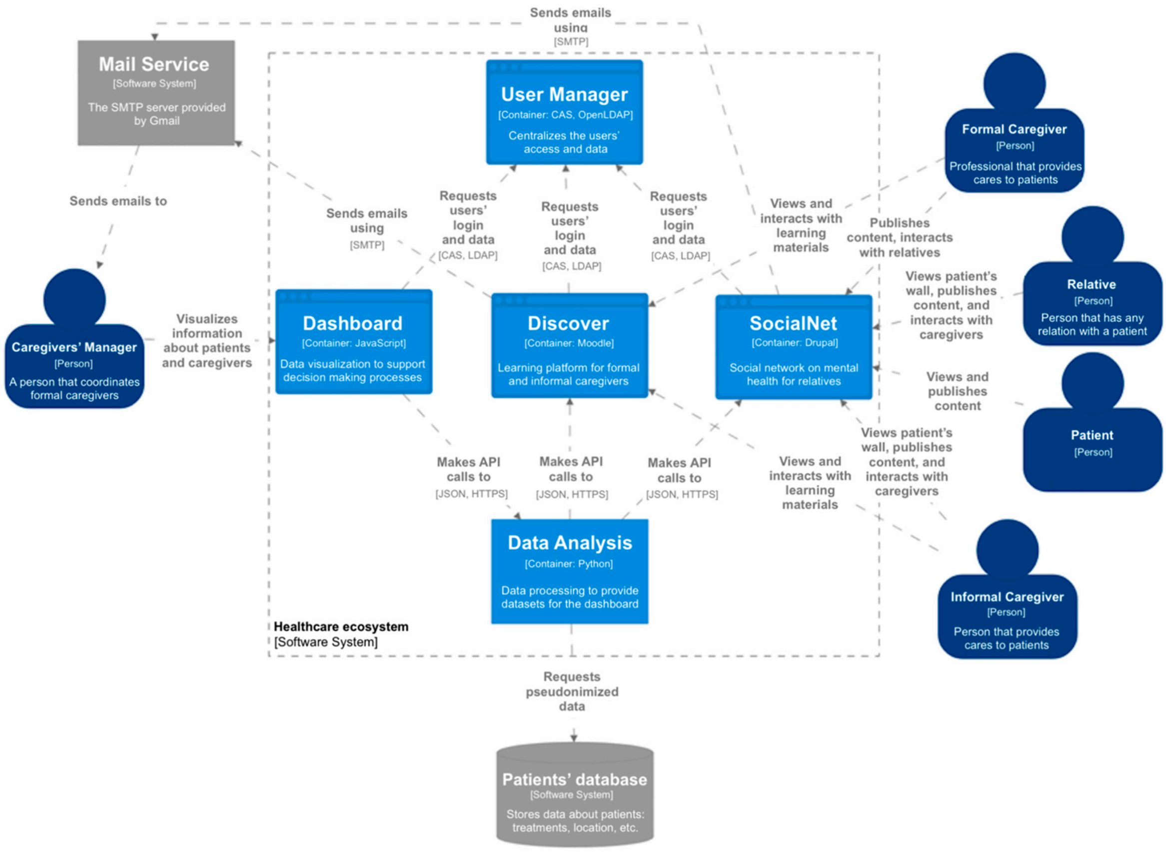


Figure 3. Healthcare ecosystem architecture in the C4 model. This diagram is also available on http://doi.org/10.5281/zenodo.3490541.

5. Dashboard for Supporting Caregivers

Technological ecosystems support decision-making processes based on the knowledge created or transformed into the different components of the ecosystem. In the healthcare ecosystem, these processes should be supported in order to improve the physical and mental health problems of the caregivers, both formal and informal, and the patients with cognitive and physical impairments.

One of the main goals of information dashboards is to facilitate decision-making processes. In particular, it is important to facilitate these processes to caregivers' managers. Moreover, the data generated from the different software components and the patients' database should be processed in a way that it is understandable by the managers. Visualizations adapted to the skills of the caregivers' managers can provide useful information to support the decisions related to patients and caregivers. The visual analysis allows the identification of patterns, outliers, relationships, and so on.

A series of goals have been extracted from the caregivers' manager profile that will be addressed in this case study. Caregivers' managers have a series of objectives regarding the analysis of the collected data. These requirements can be classified into two main goals: insights about the patients' relatives and insights about the workload of the caregivers. The generated knowledge from the obtained insights can support decision-making regarding the distribution of caregivers to reduce workload and the impact of the patients' relatives on their healthcare.

Figure 4 shows the instantiation of the meta-model's section that addresses the users' goals. On the one hand, the first goal regarding relatives has the following description: "To analyze the relationship between the attention given by relatives and the patient's health". In this case, relatives (i.e., persons that have any kind of relationship with the patient) are the targets of the analysis. Two lower-level tasks arise to achieve that goal; that is, observe the relatives' data and observe the patients' data. Given the fact that the goal asks for "analyzing a relationship", a dashboard component must support the mentioned tasks to enable managers to obtain insights. The selected component is a scatter chart that will be detailed later.

On the other hand, it is necessary to provide support to the second goal: "To gain insights about the workload of the caregivers". The workload can be visualized in different ways and can involve different variables. In this case, two components were selected to support the second goal: a heat map and a treemap, to let managers identify patterns or relevant data points regarding the caregivers' distribution along time and among patients.

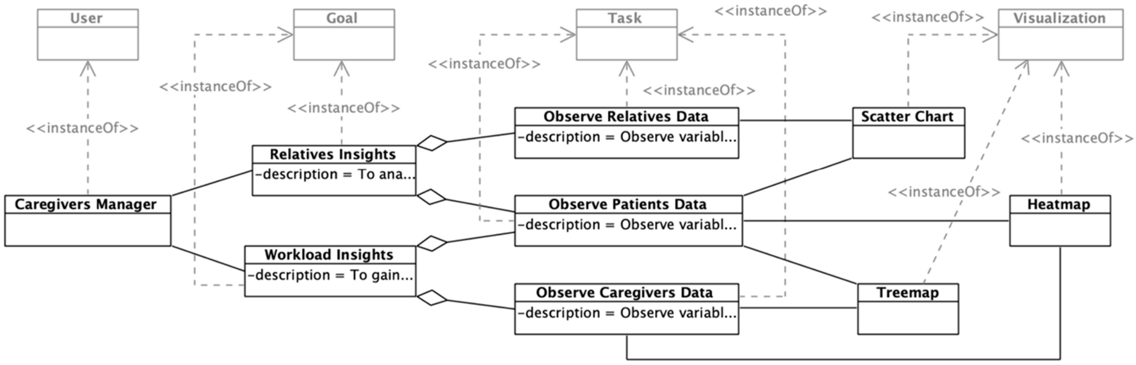


Figure 4. Dashboard meta-model instantiation regarding the user goals and tasks.

Moreover, the caregiver manager profile has the following characteristics: high domain knowledge about the context (i.e., healthcare, dementia, and cognitive impairment). This profile also has low visualization literacy, which, along with the domain knowledge, will influence the selection of proper components and features for the dashboard, as reflected in the meta-model.

As this case study is focused on a user profile (and not on an individual user), preferences, bias, and disabilities are not considered, because they are very personal data that cannot be generalized into a unique profile. However, these characteristics would have the same influencing role on the dashboard as the domain knowledge and visualization literacy. The user profile instance can be seen in Figure 5.

Using the identified characteristics and goals, a "Manager Dashboard" was instantiated. This dashboard consists of a page with two rows, with the top one containing two columns. These specific containers hold the components, in this case, the three aforementioned visualizations: a scatter chart, a heat map, and a treemap. The dashboard layout and the location of the visualizations are given by the structure and space available for the components. Finally, each visualization can be decomposed on primitives that support the introduced requirements.

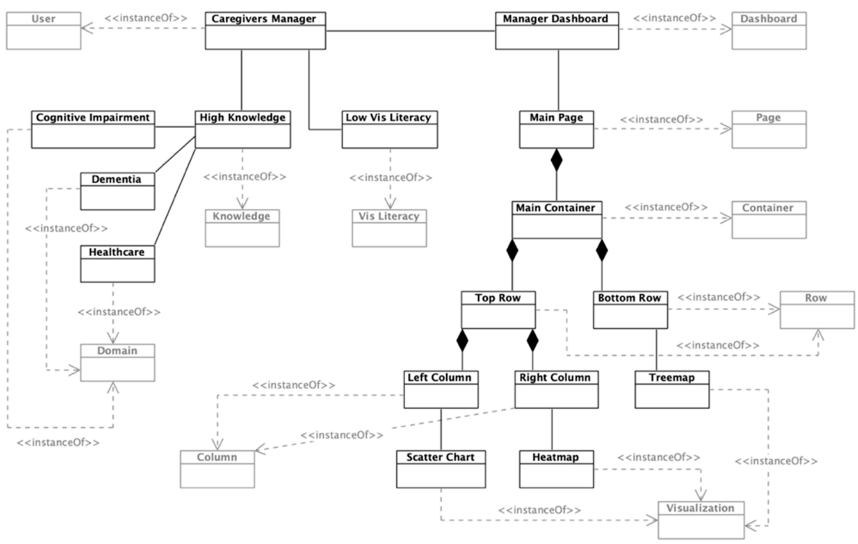


Figure 5. Dashboard meta-model instantiation regarding the user profile and dashboard layout.

First, a scatter chart can be composed of different primitives that will hold different data dimensions encoded through different channels (Figure 6). In this case, two data variables will be encoded through the scatter chart: the patient's health and the patient's given attention (by his/her relatives). These two variables are encoded through two linear scales that will position each data point (circles) within the visualization container. The scales are visible through two axes: the *X-Axis* and *Y-Axis*. Each axis has a label to ease the readability of the visualization.

The two represented variables are part of a health dataset, which belongs to three main domains: *Healthcare*, *Dementia*, and *Cognitive Impairment*. This visualization enables caregivers' managers to obtain insights about the relationship between the patients' health and their attention levels given.

The next visualization is a heat map (Figure 7). The instantiated heat map is composed of a series of cells (rectangles, in this case) positioned by two categorical scales (weekday and time interval). The actual workload is encoded by a monochromatic color scale that would be more intense if the workload is high, and less intense if the workload is low, so each rectangle has three channels: X and Y positions and color. The color scale is associated with a legend to increase the readability of the visualization. The color encoding values are the outcome of an operation, more specifically, an aggregation (count) of the workload variable by weekday and time interval, obtaining the total workload for each individual cell of the heat map. This visualization allows the recognition at first sight of the days and time intervals in which the workload is higher, and enables the manager to make decisions about the distribution of caregivers to improve efficiency.

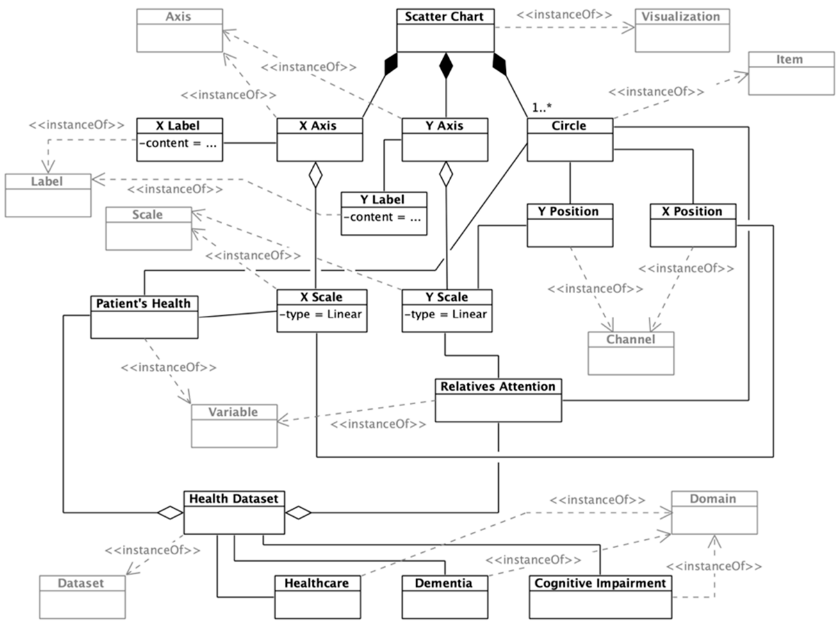


Figure 6. Dashboard meta-model instantiation regarding the scatter chart component.

Finally, the treemap component would hold information about the patients assigned to each caregiver (Figure 8). The caregivers' workload regarding patients is encoded through rectangles, the size of which varies depending on the quantity of time devoted to each patient. These data are obtained by grouping the time devoted variable by caregiver and patient, performing a *Count* operation. Each rectangle's color also encodes the caregiver in charge of the patient.

The mentioned rectangles are enclosed within a container rectangle that represents each caregivers' patients. The container rectangle is an instance of a *Link* mark, which allows encoding containment and connection relationships among visual items [39]. The instantiated treemap encodes the workload-by-patient of each caregiver, supporting managers in analyzing how to balance each caregiver's workload without reducing patient care.

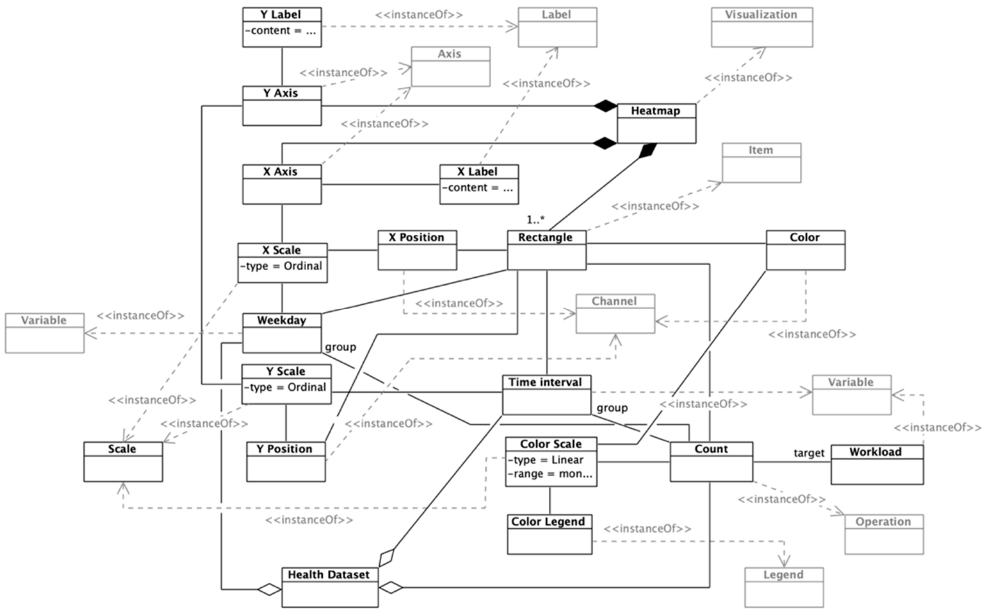


Figure 7. Dashboard meta-model instantiation regarding the heat map component.

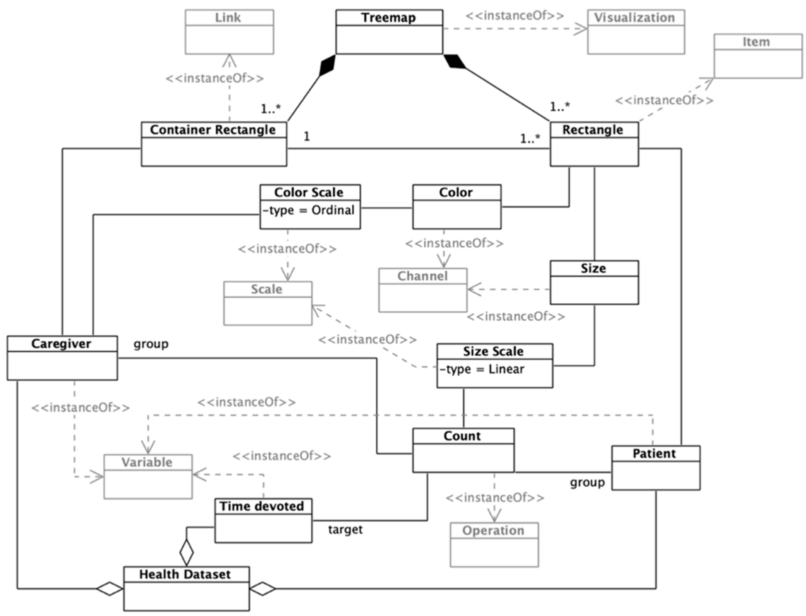


Figure 8. Dashboard meta-model instantiation regarding the treemap component.

To provide a visual example of how the instantiated dashboard would look, a simple dashboard following the described structure and features was implemented (Figure 9).

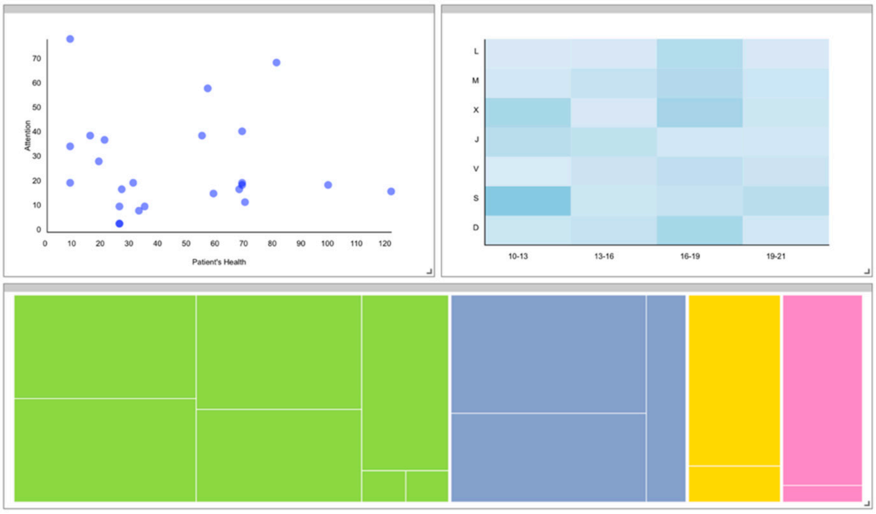


Figure 9. A dashboard prototype that follows the described instance.

6. Conclusions

Technological solutions are used in the medical area not only to support the daily activities in health organizations, but also to improve treatments. In particular, there is a need to improve the support received by informal and formal caregivers and dependent persons at their homes. The proposed ecosystem aims to provide a set of services to prevent the institutionalization of the dependent persons, enabling them to stay at home and thus reducing care costs. The ecosystem offers psychoeducation tools to caregivers and provides tools to involve relatives in the patients' daily life.

The dashboard for supporting caregivers is part of the healthcare ecosystem. It aims to provide tools based on visual analysis to support decision-making processes related to the workload of caregivers and the situation of the patients. The dashboard should be adapted to the different needs of its users. Furthermore, the medical contexts in which the ecosystem could be implemented are very different, so the ecosystem should be adapted to these different contexts.

The dashboard meta-model provides a "template" for generating concrete dashboard solutions. This meta-model was instantiated to obtain a concrete model for the presented context; that is, healthcare. Three visualizations were selected and instantiated to support the caregivers' managers' goals and decisions with data.

The meta-model supports the instantiation of fine-grained features regarding visualizations by distinguishing basic primitives that can be combined to build any type of chart. The meta-model structure makes possible any combination of marks, axes, annotations, resources, and so on, avoiding the necessity of relying on pre-defined charts, thus allowing more freedom when building custom visualizations for certain dashboards.

Introducing the user in the meta-model is essential, as their information goals are the backbone of the design processes. Characteristics including biases, literacy, preferences, disabilities, and so on, however, should also be considered to ensure that the dashboard is developed, taking into account different users' dimensions that could influence their user experience. As has been shown, the users' characteristics and goals were crucial to select appropriate visualizations and encodings. Future research lines will involve the refinement of the meta-model through the addition of constraints, rules, and design guidelines, seeking the support of the automatic generation of concrete dashboards.

Author Contributions: Conceptualization, A.V.-I., A.G.-H., F.J.-G.P., and R.T.; Methodology, A.V.-I. and A.G.-H.; Software, A.V.-I.; Validation, A.V.-I., A.G.-H., F.J.-G.P., and R.T.; Formal Analysis, A.V.-I., A.G.-H., F.J.-G.P., and R.T.; Investigation, A.V.-I., A.G.-H., F.J.-G.P., and R.T.; Resources, A.V.-I., A.G.-H., F.J.-G.P., and R.T.; Writing—Original Draft Preparation, A.V.-I., and A.G.-H.; Writing—Review & Editing, F.J.-G.P.; Supervision, F.J.-G.P. and R.T.; Project Administration, F.J.-G.P.; Funding Acquisition, F.J.-G.P. and A.V.-I.

Funding: This research was partially funded by the Spanish Government Ministry of Economy and Competitiveness throughout the DEFINES project grant number [TIN2016-80172-R] and the Ministry of

Education of the Junta de Castilla y León (Spain) throughout the T-CUIDA project [SA061P17]. This research was supported by the Spanish *Ministry of Education*, *Culture*, and *Sport* under an FPU fellowship (FPU17/03276).

Acknowledgments: The authors would like to thank the InterAction and eLearning Research Group (GRIAL) for its support to conduct the present research; https://grial.usal.es.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Spender, J.C. Making knowledge the basis of a dynamic theory of the firm. *Strateg. Manag. J.* **1996**, *17*, 45–62.
2. Teece, D.J. Strategies for managing knowledge assets: The role of firm structure and industrial context. *Long Range Plan.* **2000**, *33*, 35–54.
3. Nonaka, I.; Takeuchi, H. *The Knowledge-Creating Company: How Japanese Companies Create the Dynamics of Innovation*; Oxford University Press: Oxford, UK, 1995.
4. King, W.R. Knowledge management and organizational learning. In *Knowledge Management and Organizational Learning*; King, W.R., Ed.; Springer US: Cham, Switzerland, 2009; Volume 4, pp. 3–13.
5. Hargreaves, A. *Teaching in the Knowledge Society: Education in the Age of Insecurity*; Teachers College Press: New York, NY, USA, 2003.
6. Davenport, T.H.; De Long, D.W.; Beers, M.C. Successful knowledge management projects. *Sloan Manag. Rev.* **1998**, *39*, 43–57.
7. *World Population Prospects: The 2017 Revision, Key Findings and Advance Tables. Esa/p/wp/248*; United Nations, Department of Economic and Social Affairs, Population Division: San Francisco, CA, USA, 2017.
8. García-Peñalvo, F.J. Technological ecosystems for enhancing the interoperability and data flows. *J. Inf. Technol. Res.* **2018**, *11*, vi-x.
9. García-Peñalvo, F.J.; García-Holgado, A. *Open Source Solutions for Knowledge Management and Technological Ecosystems*; IGI Global: Hershey, PA, USA, 2017.
10. García-Holgado, A.; García-Peñalvo, F.J. The evolution of the technological ecosystems: An architectural proposal to enhancing learning processes. In Proceedings of the First International Conference on Technological Ecosystem for Enhancing Multiculturality (teem'13), Salamanca, Spain, 14–15 November 2013; ACM: New York, NY, USA, 2013; pp. 565–571.
11. García-Holgado, A.; Marcos-Pablos, S.; Therón, R.; García-Peñalvo, F.J. Technological ecosystems in the health sector: A mapping study of european research projects. *J. Med. Syst.* **2019**, *43*, 100.
12. García-Holgado, A.; Marcos-Pablos, S.; García-Peñalvo, F.J. A model to define an ehealth technological ecosystem for caregivers. In *New Knowledge in Information Systems and Technologies. Worldcist'19 2019. Advances in Intelligent Systems and Computing*; Rocha, Á., Adeli, H., Reis, L., Costanzo, S., Eds.; Springer: Cham, Switzerland, 2019; Volume 932, pp. 422–432.
13. Kehrer, J.; Hauser, H. Visualization and visual analysis of multifaceted scientific data: A survey. *IEEE Trans. Vis. Comput. Graph.* **2012**, *19*, 495–513.
14. Keim, D.A.; Mansmann, F.; Schneidewind, J.; Ziegler, H. Challenges in visual data analysis. In Proceedings of the Tenth International Conference on Information Visualisation (IV'06), London, UK, 5–7 July 2006; pp. 9–16.
15. Berinato, S. *Good Charts: The Hbr Guide to Making Smarter, More Persuasive Data Visualizations*; Harvard Business Review Press: Brighton, MA, USA, 2016.
16. Padilla, L. How do we know when a visualization is good? Perspectives from a cognitive scientist. Available online: https://medium.com/multiple-views-visualization-research-explained/how-do-we-know-when-a-visualization-is-good-c894b5194b62 (accessed on 11 January 2019).
17. Hullman, J.; Adar, E.; Shah, P. The impact of social information on visual judgments. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, Vancouver, BC, USA, 7–12 May 2011; pp. 1461–1470.
18. Kim, Y.-S.; Reinecke, K.; Hullman, J. Data through others' eyes: The impact of visualizing others' expectations on visualization interpretation. *IEEE Trans. Vis. Comput. Graph.* **2018**, *24*, 760–769.
19. Bedek, M.A.; Nussbaumer, A.; Huszar, L.; Albert, D. Methods for discovering cognitive biases in a visual analytics environment. In *Cognitive Biases in Visualizations*; Springer: Berlin, Germany, 2018; pp. 61–73.

20. Valdez, A.C.; Ziefle, M.; Sedlmair, M. A framework for studying biases in visualization research. In Proceedings of the 2nd DECISIVe Workshop, Salamanca, Spain, 22–23 March 2017.
21. Lee, S.; Kim, S.-H.; Kwon, B.C. Vlat: Development of a visualization literacy assessment test. *IEEE Trans. Vis. Comput. Graph.* **2017**, *23*, 551–560.
22. Boy, J.; Rensink, R.A.; Bertini, E.; Fekete, J.-D. A principled way of assessing visualization literacy. *IEEE Trans. Vis. Comput. Graph.* **2014**, *20*, 1963–1972.
23. Maltese, A.V.; Harsh, J.A.; Svetina, D. Data visualization literacy: Investigating data interpretation along the novice—Expert continuum. *J. Coll. Sci. Teach.* **2015**, *45*, 84–90.
24. Toker, D.; Conati, C.; Carenini, G.; Haraty, M. Towards adaptive information visualization: On the influence of user characteristics. In Proceedings of the International Conference on User Modeling, Adaptation, and Personalization, Montreal, QC, Canada, 16–20 July 2012; Springer: Berlin, Germany, 2012; pp. 274–285.
25. Dillon, A. Spatial-semantics: How users derive shape from information space. *J. Am. Soc. Inf. Sci.* **2000**, *51*, 521–528.
26. Conati, C.; Carenini, G.; Hoque, E.; Steichen, B.; Toker, D. Evaluating the impact of user characteristics and different layouts on an interactive visualization for decision making. In *Computer Graphics Forum*; Wiley Online Library, New Jersey, USA: 2014; pp. 371–380.
27. Mackinlay, J.; Hanrahan, P.; Stolte, C. Show me: Automatic presentation for visual analysis. *IEEE Trans. Vis. Comput. Graph.* **2007**, *13*, 1137–1144.
28. Belo, O.; Rodrigues, P.; Barros, R.; Correia, H. Restructuring dynamically analytical dashboards based on usage profiles. In *International Symposium on Methodologies for Intelligent Systems*; Springer: Cham, Switzerland, 2014; pp. 445–455.
29. Vázquez-Ingelmo, A.; García-Peñalvo, F.J.; Therón, R. Domain engineering for generating dashboards to analyze employment and employability in the academic context. In Proceedings of the 6th International Conference on Technological Ecosystems for Enhancing Multiculturality, Salamanca, Spain, 24–26 October 2018.
30. Vázquez-Ingelmo, A.; García-Peñalvo, F.J.; Therón, R. Taking advantage of the software product line paradigm to generate customized user interfaces for decision-making processes: A case study on university employability. *PeerJ Comput. Sci.* **2019**, *5*, e203.
31. Kleppe, A.G.; Warmer, J.; Bast, W. *Mda Explained. The Model Driven Architecture: Practice and Promise*; Addison-Wesley Longman Publishing Co., Inc.: Boston, MA, USA, 2003.
32. Mellor, S.J.; Scott, K.; Uhl, A.; Weise, D. Model-driven architecture. In *Advances in Object-Oriented Information Systems, Proceedings of the Oois 2002 Workshops, Montpellier, France, 2 September 2002*; Bruel, J.-M., Bellahsene, Z., Eds.; Springer: Berlin/Heidelberg, Germany, 2002; pp. 290–297.
33. Álvarez, J.M.; Evans, A.; Sammut, P. Mapping between levels in the metamodel architecture. In *UML 2001—The Unified Modeling Language. Modeling Languages, Concepts, and Tools. Uml 2001. Lecture Notes in Computer Science*; Gogolla, M., Kobryn, C., Eds.; Springer: Berlin/Heidelberg, Geramny, 2001; Volume 2185, pp. 34–46.
34. García-Holgado, A.; García-Peñalvo, F.J. Validation of the learning ecosystem metamodel using transformation rules. *Future Gener. Comput. Syst.* **2019**, *91*, 300–310.
35. Brown, S. *Software Architecture for Developers*; Leanpub: Victoria, BC, Canada, 2018.
36. Vázquez Ingelmo, A.; García-Peñalvo, F.J.; Therón, R.; Conde González, M.Á. Extending a dashboard meta-model to account for users' characteristics and goals for enhancing personalization. In Proceedings of the Learning Analytics Summer Institute (LASI) Spain 2019, Vigo, Spain, 27–28 June 2019.
37. Mackinlay, J. Automating the design of graphical presentations of relational information. *ACM Trans. Graph. (Tog)* **1986**, *5*, 110–141.
38. Iliinsky, N.; Steele, J. *Designing Data Visualizations: Representing Informational Relationships*; O'Reilly Media, Inc.: Sebastopol, CA, USA, 2011.
39. Munzner, T. *Visualization Analysis and Design*; AK Peters/CRC Press: Boca Raton, FL, USA, 2014.
40. Logre, I.; Mosser, S.; Collet, P.; Riveill, M. Sensor data visualisation: A composition-based approach to support domain variability. In *European Conference on Modelling Foundations and Applications*; Springer: Cham, Switzerland, 2014; pp. 101–116.

41. Bäuml, J.; Froböse, T.; Kraemer, S.; Rentrop, M.; Pitschel-Walz, G. Psychoeducation: A basic psychotherapeutic intervention for patients with schizophrenia and their families. *Schizophr. Bull.* **2006**, *32*, S1–S9.
42. Toribio-Guzmán, J.M.; García-Holgado, A.; Soto Pérez, F.; García-Peñalvo, F.J.; Franco Martín, M. Usability evaluation of a private social network on mental health for relatives. *J. Med. Syst.* **2017**, *41*, 137.

7.14 Appendix N. Connecting domain-specific features to source code: towards the automatization of dashboard generation

Cluster Computing
https://doi.org/10.1007/s10586-019-03012-1

Connecting domain-specific features to source code: towards the automatization of dashboard generation

Andrea Vázquez-Ingelmo[1] · Francisco José García-Peñalvo[1] · Roberto Therón[1,2] · Daniel Amo Filvà[3] · David Fonseca Escudero[4]

Received: 9 September 2019 / Revised: 15 October 2019 / Accepted: 29 October 2019

Abstract

Dashboards are useful tools for generating knowledge and support decision-making processes, but the extended use of technologies and the increasingly available data asks for user-friendly tools that allow any user profile to exploit their data. Building tailored dashboards for any potential user profile would involve several resources and long development times, taking into account that dashboards can be framed in very different contexts that should be studied during the design processes to provide practical tools. This situation leads to the necessity of searching for methodologies that could accelerate these processes. The software product line paradigm is one recurrent method that can decrease the time-to-market of products by reusing generic core assets that can be tuned or configured to meet specific requirements. However, although this paradigm can solve issues regarding development times, the configuration of the dashboard is still a complex challenge; users' goals, datasets, and context must be thoroughly studied to obtain a dashboard that fulfills the users' necessities and that fosters insight delivery. This paper outlines the benefits and a potential approach to automatically configuring information dashboards by leveraging domain commonalities and code templates. The main goal is to test the functionality of a workflow that can connect external algorithms, such as artificial intelligence algorithms, to infer dashboard features and feed a generator based on the software product line paradigm.

Keywords SPL · Domain engineering · Meta-model · Information dashboards · Feature model · Artificial intelligence · Automatic configuration

1 Introduction

Nowadays, a lot of technological contexts ask for tailored products; new user profiles have arisen due to the extended use of technologies, and these profiles might demand distinct features in their products. The use of software tools is not restricted and limited to technical profiles anymore, which have expanded the variety of products that one can find to solve specific problems.

This increase in technology usage has also increased the quantity of available and generated data. People can use technology to explore information and to support their decision-making processes [1].

However, the exponential growth of data is a challenge for analyzing and generating knowledge from them, as complex patterns, for example, are less easy to detect at a glance. That is why disciplines like information visualization study how data can be encoded to foster insight delivery.

Tools like information dashboards arrange data into different views to ease the analysis of large datasets and to generate knowledge [2]. Dashboards are powerful tools, but designing and developing them are not trivial tasks; the variety of user-profiles induces the necessity of crafting dashboards according to specific user needs. In other

✉ Andrea Vázquez-Ingelmo
andreavazquez@usal.es

1 GRIAL Research Group, Computer Science Department, Research Institute for Educational Sciences, University of Salamanca, Salamanca, Spain

2 VisUSAL Research Group, Computer Science Department, University of Salamanca, Salamanca, Spain

3 Departament d'informàtica, La Salle, Universitat Ramón Llull, Barcelona, Spain

4 Departament d'arquitectura, La Salle, Universitat Ramón Llull, Barcelona, Spain

words, dashboards need to be tailored to enhance user experience and to achieve particular user goals. However, tailoring dashboards for any potential user profile would consume significant resources and the most important: significant time.

Developing a user-specific dashboard involves the analysis of the user goals and required tasks, the analysis of the user herself/himself, etc., in addition to the actual development of the dashboard, which also consumes time and resources.

Given this situation, different paradigms arose to ease the tailoring of these kind of products by reusing artifacts or by developing dashboard generators [3]: from configuration wizards that enable users to build their own dashboards, structured files that are rendered into final products, agents that personalize the tools based on the users' behavior, etc., to software engineering frameworks like model-driven engineering or the software product line paradigm.

The latter software engineering methods provide a framework to build systems based on high-level resources (meta-models, core assets, etc.) that can be instantiated into concrete products. Indeed, one of the main benefits of the Software Product Line (SPL) paradigm is the capacity of tailoring products to match particular user needs without consuming several resources and development time [4, 5] by increasing reusability not only at code-level but also at knowledge-level.

The SPL paradigm is a powerful approach to leverage solutions framed within a specific domain [5, 6]. It has been used in several contexts [7–12] and achieved significant results, decreasing development times, and time-to-market. This paradigm is organized into two phases: domain engineering and application engineering [5, 6]. The domain engineering phase is focused on analyzing the target domain in which the products will be framed, obtaining a series of artifacts that will be the inputs for the application engineering phase, in which the developed models, assets, etc., are employed to build specific products according to the clients' needs.

In this paradigm, the domain engineering phase is essential. This phase thoroughly analyzes the domain to find reusability opportunities. Through the analysis and extraction of commonalities among potential products, it is possible to model abstract properties that can be instantiated and configured into concrete solutions. These properties or features are thus, essential for understanding the nature of the domain's products.

There exist different methods to account for a domain's features [13]. However, one of the most used methodologies to model SPL's features is the feature-oriented domain analysis (FODA) [14]. This method allows the representation of every possible product of an SPL in terms of a set of mandatory, optional, and alternative features.

The variability points (i.e., the locations in which the product line variability are injected) can be materialized through different methods [15]; selecting the right method is based on available resources, time and the product line's features characteristics, like the required granularity [16].

Although information dashboards can present different forms, they share common features (visual encodings, layouts, low-level tasks, etc.), which makes the SPL paradigm a suitable approach for building tailored dashboards. In fact, successful examples of the application this approach to information dashboards can be found in the literature [17, 18].

However, although the SPL paradigm can support the generation of dashboards, the features still need to be manually selected. The configuration process can be a challenge, as users might not know what dashboard configuration better suits their needs, and thus, the dashboard designer wouldn't be able to select the appropriate dashboard features. In this case, the configuration process asks for automation.

How can a set of proper dashboard's features be inferred from users? This paper presents a proposal for generating dashboards employing domain engineering and the possibilities of connecting the SPL's core assets with external prediction algorithms to obtain suitable features.

Specifically, the main goal behind presenting this approach is to establish a foundation for leveraging artificial intelligence algorithms to ease the configuration process of information dashboards.

These algorithms can learn from the users' goals, behaviors and characteristics to infer their needs regarding data and their visualization; given input datasets, artificial intelligence techniques are employed to discover underlying relationships and patterns among variables.

In this context, AI algorithms are intended to be employed to find relationships among users' characteristics and visualization elements, in order to discover rules and use them for crafting tailored displays that foster knowledge generation.

The rest of this paper is organized as follows. Section 2 outlines previous works regarding the application of artificial intelligence within the SPLs and visualization recommendation domains.

Section 3 describes the methodology used in each stage of this work. Section 4 presents the proposed workflow for automatically generating dashboards, followed by Sect. 5, in which a small proof-of-concept to test the feasibility of the approach is performed.

Finally, Sects. 6 and 7 discuss the results and raise some conclusions about the work done, respectively.

2 Background

2.1 Variability mechanisms in software product lines

There exist different techniques to implement variability points in SPLs. It is important to choose wisely given the requirements of the product line itself (i.e., the complexity of the software to develop, its number of features, their granularity requirements, etc.).

Variability points that correspond to a certain feature will be spread across different source files, generally [15]. That is why separating concerns at implementation level is essential to avoid this scattering, as this feature dispersion would decrease code understandability and maintainability.

Implementing each feature in individual code modules can help with the separation of concerns [15], but it is difficult to achieve fine-grained variability through this approach. A balance between code understandability and granularity should be devised to choose both a maintainable and highly customizable SPL.

This section will briefly describe different methods that are potentially suitable to the dashboards' domain given their particular features, and that could be employed to connect the outputs yielded from an external configuration algorithm. However, there are more approaches to implement variability in SPLs that can be consulted in [15].

2.1.1 Conditional compilation

Conditional compilation uses preprocessor directives to inject or remove code fragments from the final product source code. This method allows the achievement of any level of feature granularity due to the possibility of inserting these directives at any point of the code, even at expression or function signature level [19]. Also, although pretended to the C language, preprocessor directives can be used for any language and arbitrary transformations [20]. The main drawback of this approach is the decrease of code readability and understandability as interweaving and nesting these preprocessor directives makes the code maintainability a tedious task [21].

2.1.2 Frames

Frame technology is based on entities (frames) that are assembled to compose final source code files. Frames use preprocessor-like directives to insert or replace code and also to set parameters [15]. An example of a variability implementation method based on frame technology is the XML-based Variant Configuration Language (XVCL) [22]. Through this approach, only the necessary code is introduced in concrete components by specifying frames that contain the code and directives associated with different features and variants. XVCL is independent of the programming language and can handle variability at any granularity level [23].

2.1.3 Template engines

Template engines allow the parameterization and inclusion/exclusion of code fragments through different directives. If the template engine allows the definition of macros, features can be refactored into different code fragments encapsulated through these elements, improving the code organization and enabling variability at any level of granularity. Templating engines can also be language-independent, providing a powerful tool for generating any kind of source file [24] by using programming directives such as loops and conditions.

2.1.4 Aspect-oriented programming

Aspect-oriented programming (AOP) allows the implementation of crosscutting concerns through the definition of aspects, centralizing features that need to be present in different source files through unique entities (aspects) thus improving code understandability and maintainability by avoiding scattered features and "tangled" code [25].

AOP is a popular method to materialize variability points in SPLs due to the possibility of modifying the system behavior at certain points, namely join points [26–28]. However, AOP could lack o of fine-grained variability (i.e., variability at sentence, expression or signature level) and particular frameworks or language extensions are necessary to implement aspects in certain programming languages.

2.2 Automatic visualization recommendations

Selecting the right information visualizations is a challenging step when developing dashboards. This step is also crucial because the effectiveness of the dashboard or visualization can be compromised if the design decisions are not based on the context or the final users.

To ease this process, several authors in the literature aim at automatically recommending visualizations. As the majority of works point out, the particular design of a visualization depends on different aspects, like the domain knowledge of the users, the tasks that the visualization must support or the data characteristics [29].

Developing a dashboard or a visualization need expert knowledge to help with design decisions based on the audience and context to which the product is aimed for. But can this developing process and design decisions be automated? There are different approaches to tackle these challenges.

For example, some methods use visual mapping and rules to recommend a certain visualization based on the target data to be displayed [29]. Through this method, the dataset is analyzed, and a series of rules are applied to select a proper visual mark, scale, channel, etc. based on the type or properties of the data variables. Some tools take advantage of this method to develop suitable visualizations, like Tableau's Show Me [30], Manyeyes [31], or Voyager [32].

On the other hand, VizDeck [33] proposes a set of visualization based on the input data, and the user selects the ones that prefer the most. The system learns from these user interactions, providing the user with similar visualizations subsequently.

Other solutions take into account the context in which the visualization is framed, attending to the domain, experience, or knowledge. For example, in [34] a visualization ontology named VISO is employed no only to annotate data, but also "to represent context and factual knowledge." Then, a ranking process is executed to yield the recommendations using rules to rate the suitability of visual encodings.

User behavior is also considered as a crucial aspect when recommending visualizations. For example, in [35, 36], the user behavior when interacting with a visualization is logged and used later to search for interesting patterns that could mean that the user asks for a different visualization. When the visual task is inferred from the user behavior, a set of ranked annotated visual metaphors is retrieved.

Furthermore, an application of content and collaborative filtering to recommend visualizations can be found in [37]. Through these methods, potentially suitable visualizations for a specific user can be identified.

Finally, applications of neural networks to infer specific features of a visualization are also present in the literature. VizML [38] used the Plotly API[1] to retrieve information about how the Plotly community crafted different visualizations for different datasets to train a set of models. The outputs of these models are a set of visualization design choices at visualization-level and encoding-level, including mark types or properties regarding axes, like if a variable is encoded on the *X* or the *Y*-axis or if the axis is shared.

A similar approach is taken in Data2Vis [39], where the inputs of a sequence to sequence model are data characteristics and the output is a visualization specification in Vega-Lite.

[1] https://plot.ly/.

As well as with the variability mechanisms of software product lines, it can be seen that there are many methods to address the automatic recommendation of suitable visualizations. The next section details the specific methodology followed for this work.

3 Methodology

3.1 Conceptualization: meta-modeling and domain engineering

As introduced, the employed framework involves two software engineering approaches: meta-modeling and the software product line paradigm. The combination of meta-modeling to obtain a domain abstraction with the SPL philosophy of systematically reuse assets provides a powerful framework for building families of products.

The first (and essential) phase in developing SPLs is domain engineering. This phase is conducted to understand the domain in which the potential products of the SPL are framed. But what exactly is the meaning of "understand the domain"? In domain engineering approaches, to understand the domain means to analyze it, to find commonalities and variability points among the domain's products. By identifying commonalities and variations, generic and abstract components can be developed to subsequently combine them, parameterize them, derive them, or to execute any other operation with the goal of obtaining a concrete product [15].

However, how can these domain characteristics be captured? Meta-models can support this conceptualization process. These elements are artifacts from the model-driven architecture paradigm [40, 41] that capture high-level and abstract concepts and document them in a structured representation. Through meta-modeling, the development of general rules, constraints, etc., for a set of related problems is fostered by abstracting common structures and associations found in different domain's instances.

In the end, meta-models can be mapped to concrete model instances and products, following the OMG four-layer meta-model architecture [42]: meta–meta-model layer (M3), meta-model layer (M2), user model layer (M1) and user object layer (M0).

The dashboards domain is indeed complex. There is a huge diversity of dashboards in terms of designs, interactions, supported tasks, displayed data, layouts, etc.

However, using meta-models and domain engineering as conceptualization tools, it can be possible to extract commonalities shared by dashboards and arrange them into abstract models that can be concretized. Figure 1 illustrates this approach.

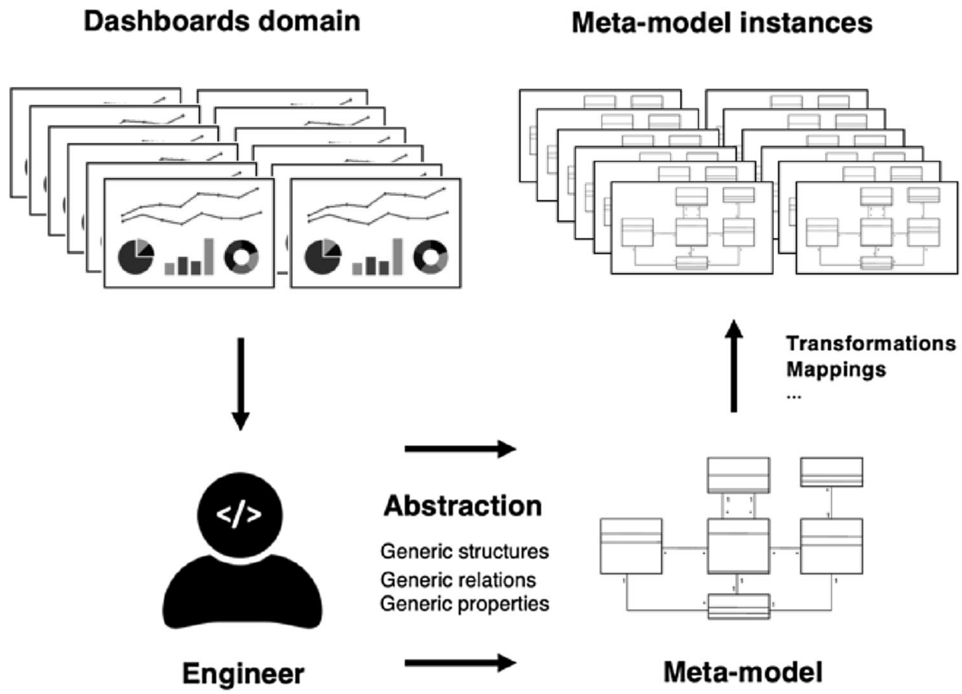


Fig. 1 Using meta-models to understand complex domains

An extract of the developed dashboard meta-model is shown in Fig. 2, in which different primitives that a visualization could have are presented.

This meta-model also accounts for user goals and characteristics, including visualization literacy, domain knowledge, bias, preferences, etc. [44]. Also, components are arranged among different containers to compose the dashboard. The full version of the meta-model can be consulted at [43].

The meta-model helps with the conceptualization of the automatic generation approach and to gain knowledge about abstract features regarding dashboards.

However, to test this first approximation and the configuration process of dashboards, an abbreviated feature model has been developed, which can be consulted in Fig. 3. This feature model reflects the elements that a dashboard display belonging to the product line could have at the moment. In this case, a dashboard display can be composed of different pages with different visualizations.

As presented in the meta-model, a visualization is composed of marks that could encode data through different channels.

The "Base Structure" feature is the core of every visualization, which supports its generation and its basic functionalities.

Of course, this feature model can be extended to support more characteristics (scale types, annotations, interaction patterns, etc.), but in order to test the feasibility of this approach in a straightforward way, the feature model has been kept simple.

3.2 Variability implementation: code templates

One of the challenges regarding the development and configuration of SPLs is the desired features' granularity. In the dashboards' domain, features' are fine-grained; variability is focused on visual and interactive elements of the presentation layer, because slight modifications on different aspects, like interaction patterns, layouts, color palettes, etc., can influence the perceived usability and user experience [45]. In other words, dashboards' features can involve statement-level, and even expression-level granularity [16], meaning that tiny fragments of code can be affected by the SPL configuration process.

Given these granularity requirements, the chosen mechanism to implement the SPL is based on template engines. These tools allow developers to tag sections and to parameterize fragments of source code to subsequently instantiate them with particular values, thus obtaining concrete source files [17].

Jinja2 has been selected as the template engine given its powerful features like the possibility of enclosing code within macros and the multiple directives available (loops, conditional directives, variable definitions, etc.). On the other hand, the chosen technology to implement the SPL's assets (i.e., the code templates' content) is the data visualization framework Vega-Lite.[2] This is not an arbitrary selection; Vega-Lite allows building custom visualizations through a fine-grained configuration of their properties,

[2] https://vega.github.io/vega-lite/.

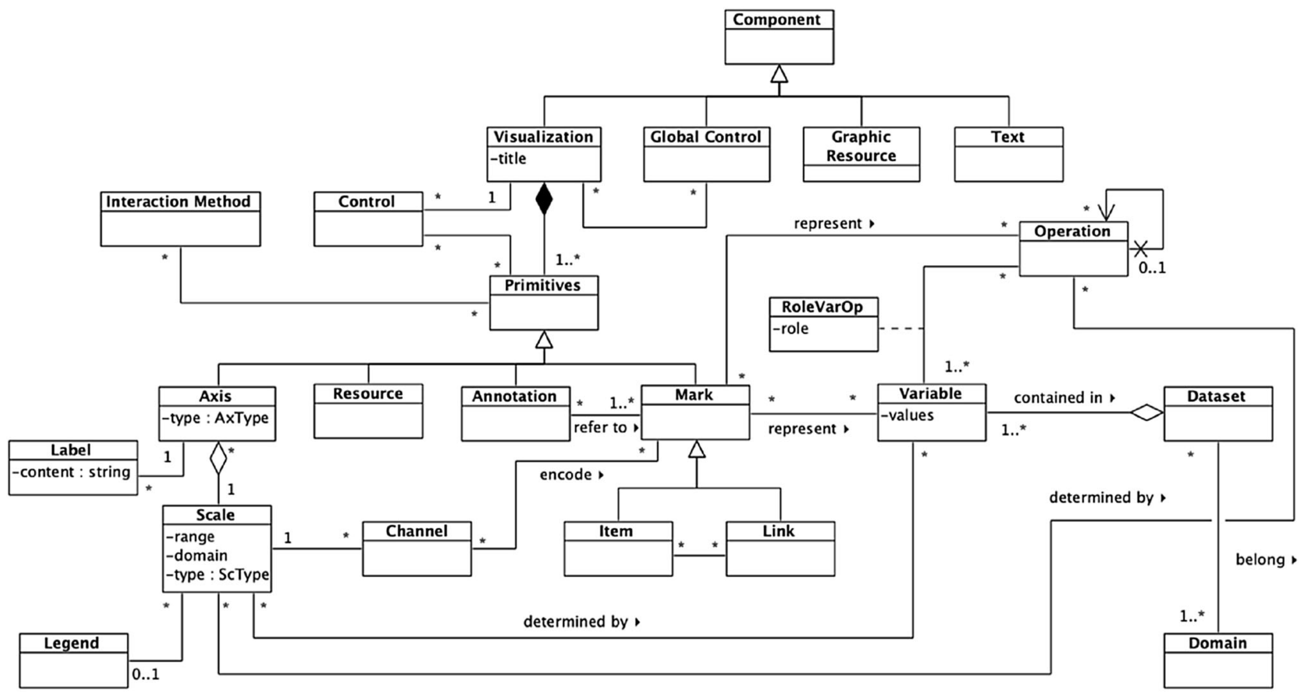


Fig. 2 An extract of the dashboard meta-model where the visualization components are decomposed into abstract primitives [43]

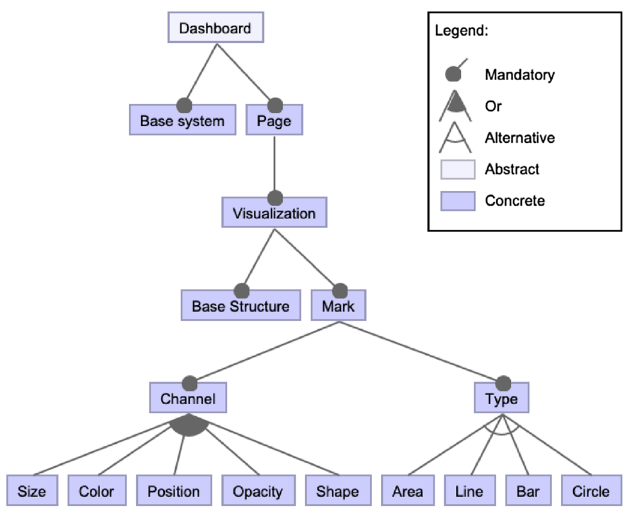


Fig. 3 The dashboard product line's feature model

which matches the requirements of this product line approach and fits with the identified domain's features.

3.3 Configuration proposal: artificial intelligence and visual mapping

As presented in Sect. 2.2, artificial intelligence has arisen as a powerful tool to recommend visualization types and configurations. The fact that users and developers might be biased when designing a dashboard makes artificial intelligence algorithms a useful means to drive design decisions. However, as it will be discussed, it is important to take into account that AI algorithms are fed with data, and if these data are biased, then the algorithm will also be biased [46].

Although creating a product line of dashboards can boost development processes, products still need to be configured; and this is not a trivial task.

Configuration processes are still manual or semi-automatic (supported by tools [47–49]) processes in which user requirements or the product-line's objectives have to be analyzed and materialized by selecting the appropriate features to obtain a suitable product.

This is a complex and crucial process that can compromise the effectiveness of the dashboard.

However, the configuration of a product line with several features involved can be overwhelming, even for an artificial intelligence algorithm. As previously introduced (in Sect. 3.1) the dashboard meta-model contemplates several fine-grained features; inferring every single exact value for every single feature might involve significant quantities of data and resources to get good results.

That is why using visual mapping approaches are also useful for these problems, because general guidelines must be followed to deliver correct visualizations. In this case, a visualization recommendation engine, such as CompassQL [50, 51], can be selected to act as a visual mapping engine.

Specifically, CompassQL allows partial specifications of visualizations and integrates methods for choosing and ranking the best specification.

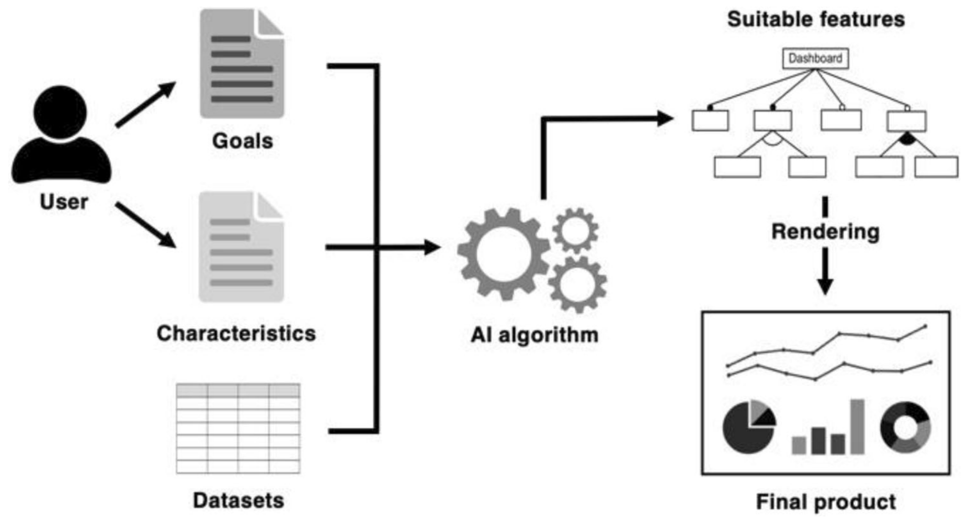


Fig. 4 Inputs and outputs of the proposed dashboards' configuration approach

The following section explains how artificial intelligence and visual mapping can be introduced in a dashboard product line configuration process.

4 Workflow for automatically setting up a dashboard product line

This section proposes a workflow to address the challenge of automatically configuring a product line of dashboards based on user necessities.

Of course, the first step is to identify the inputs and outputs of the whole process. Broadly speaking, there are three main inputs based on the previously presented meta-model in Sect. 3.1: the user's goals, the user's characteristics, and the target datasets. Following previous visualization recommendation methods [29], this method combines different aspects to obtain a hybrid approach.

The reason to use a hybrid approach is that using a single dimension for configuring dashboards or visualizations, for example, using only data characteristics or using only user preferences, can lack effectiveness.

Data characteristics need to be taken into account to build appropriate charts [52], as some visual encodings are not suitable for some type of variables.

However, while a chart can be correctly built, it can be inappropriate if its context and audience are not considered.

User goals are crucial, as visual metaphors, supported tasks or displayed data depend on what are the user's data necessities, as well as users' characteristics, because the visualization literacy, domain knowledge or even bias, can compromise the users' insight delivery process.

On the other hand, in this case, the selected method to manage tailoring capabilities is the software product line paradigm, meaning that the output of the configuration process must be a set of suitable features from the feature model. Figure 4 outlines the inputs and outputs of the process.

In the end, the problem is summarized as the necessity of tuning the core assets parameters to optimize the user experience and the effectiveness of the generated dashboard.

Users' goals, characteristics, and datasets must be structured into machine-readable documents to allow their processing and to automatize the process. The output as well, in this case, a selection of features, can be transformed into a structured file readable by the code generator which will inject the specific parameter values into the templates to generate the final dashboard.

However, as previously shown in Fig. 2, a single visualization needs several primitives, properties, and features to be selected. Identifying and configuring every potential feature would consume several resources and time. This also means that it is necessary to have not only relevant quantities of data to train a model but also information about different types of visualizations.

That is why the following configuration process is proposed. Some basic but relevant features can be predicted using user data and data schemes, for example: predict the number of visualizations that a dashboard should hold given all of the users' goals and expertise.

This could also be applied at "visualization-level," for example: predict the appropriate number of encodings/channels for a visualization or the mark type.

Then, once this information has been collected, the inferred data can be converted into a set of concrete visualizations by using visual mapping.

In this case, to test functionality, CompassQL has been selected as the visual mapping/recommendation engine to deliver the final dashboard. CompassQL enables the automatic selection of channels given the dataset and the variables to visualize.

The structure of this query language allows the connection of the outcomes from an external algorithm to a visualization query through the mentioned code templates, and the mapping results can also be used to generate visualizations with different frameworks.

5 Proof-of-concept

This section outlines the functionality of the product line generation process.

The presented proof-of-concept aims at testing the suitability and feasibility of using code templates to introduce external configurations.

The main requirement of the approach involves the capacity of adapting dashboards based on fine-grained features, following the meta-model described in Sect. 3.1.

The inputs of the approach are the selected features from the product-lines feature model (which need to be previously selected based on user requirements and the dataset structures).

A pilot prototype to test the functionality of the workflow has been implemented. Different code templates have been coded to generate example charts, such as bar charts and scatter plots by using a sample configuration file.

The code templates hold basic and generic code for every visualization, and they can be parameterized to inject concrete features. Figure 5 shows an overview of the process.

Figure 6 shows a code template snippet, where external data is used to parameterize the final code after the rendering process.

The dashboard generator gets the data from the configuration files, and passes them to the code templates to inject concrete values, thus obtaining the final source files.

```
{% for visualization in Dashboard.Visualizations %}
var query{{ visualization.id }} = {
  "spec": {
    "data": {"url": '{{ Dashboard.Data }}'},
    "mark": "{{visualization.mark}}",
    "encodings": [
    {% for channel in visualization.channels %}
      {
        "channel": "?",
        "aggregate": "?",
        "field": "{{channel.field}}",
        "type": "?"
      },
    {% endfor %}
    ]
  },
  "chooseBy": "effectiveness"
};
```

Fig. 6 Code snippet for a code template in which a CompassQL query is instantiated using external parameters

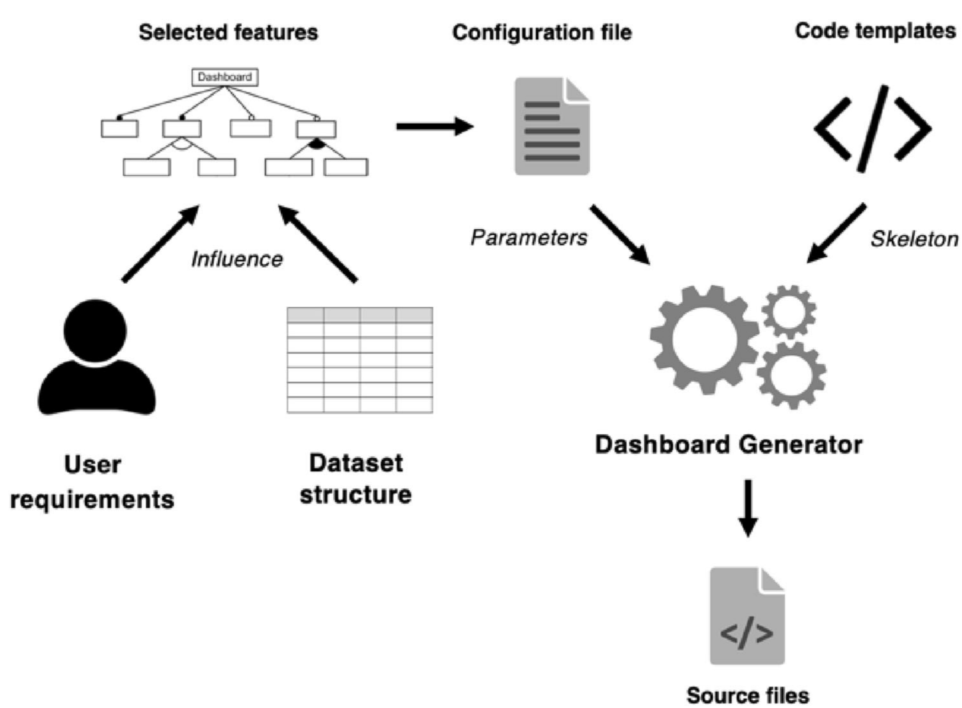


Fig. 5 Workflow overview using code templates

Fig. 7 Example dashboard configuration

```
{"Dashboard": {
    "Data": "node_modules/vega-datasets/data/cars.json",
    "Visualizations": [{
        "id": 1, "mark": "bar",
        "channels": [{"field": "Cylinders"}, {"field": "Acceleration"}]
    },
    {
        "id": 2, "mark": "circle",
        "channels": [{"field": "Horsepower"}, {"field": "Acceleration"},
                     {"field": "Cylinders"}]
    }]
}}
```

This means that, if the outcome of a previous prediction process using users' characteristics is as in Fig. 7, the delivered dashboard could be configured as in Fig. 8.

For example, the previous visualizations can be generated with the visualization framework Semiotic [53], using the CompassQL outcomes after the proper channels had been inferred through visual mapping. Figure 9 shows a code template for generating XY charts, while Fig. 10 shows the final generated visualization, which coincides with the scatter plot generated at the beginning of this section.

In addition, code templates can be used for other features of the software product line, like the addition of

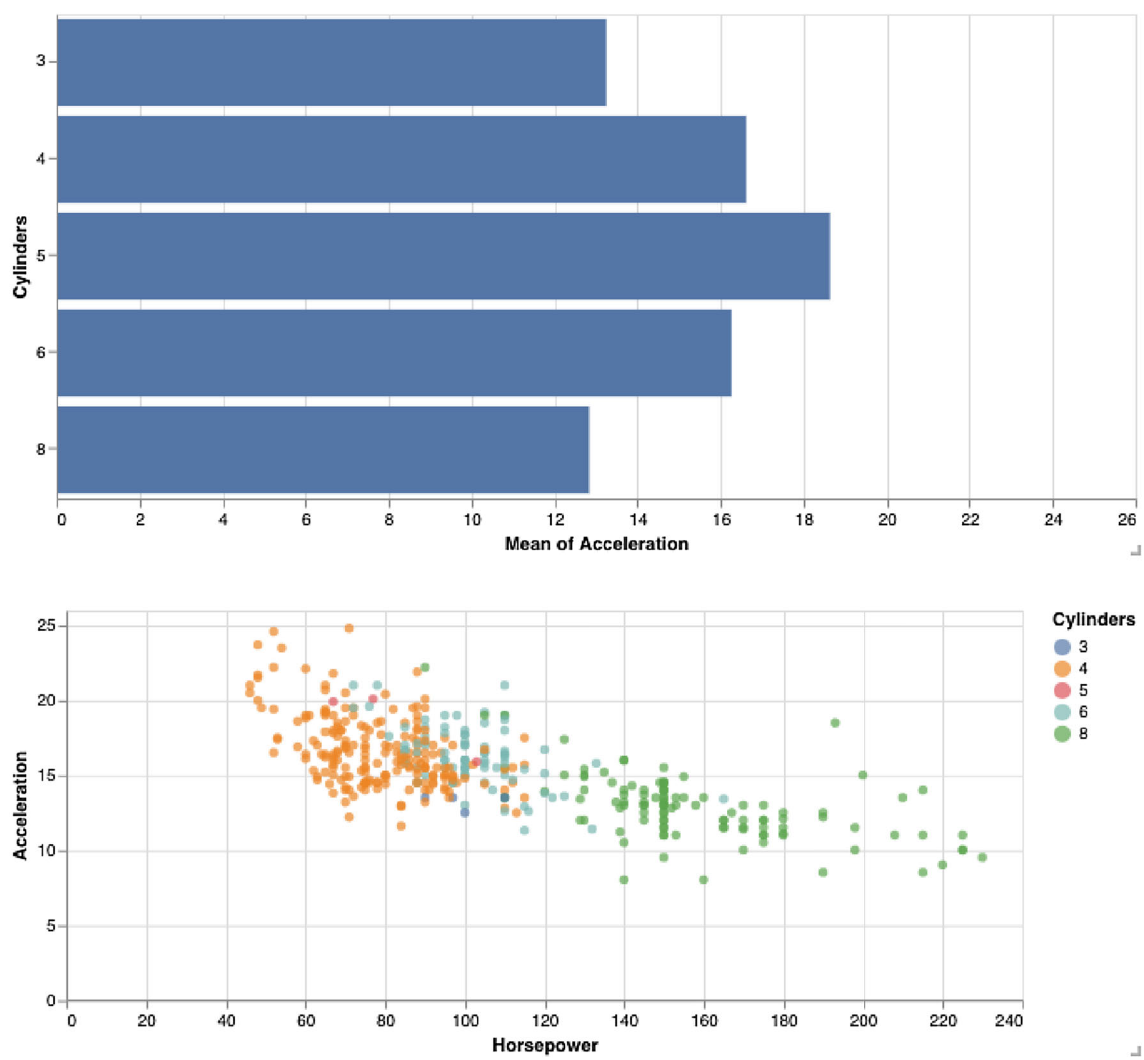


Fig. 8 Example outcome of the generation process

```
{% for visualization in Dashboard.Visualizations %}
  {% if visualization.type == "xy" %}
    var props{{ visualization.id }} = {
      "data": '{{ Dashboard.Data }}',
      "type": "{{visualization.mark}}",
      {% for encoding in visualization.encodings %}
        {% if encoding.channel == "x" %}
          xAccessor: "{{encoding.field}}",
        {% elif encoding.channel == "y" %}
          yAccessor: "{{encoding.field}}",
        {% elif encoding.channel == "color" %}
          pointStyle: d => {
            return {
                fill: colorScale(d.{{encoding.field}})
                }
          },
        {% endif %}
      {% endfor %}
```

Fig. 9 Using the visualization framework Semiotic to generate charts using properties from external files

interaction patterns, controls, styles, etc., following the same approach as in [17].

6 Discussion

This paper proposes an approach that takes advantage of domain engineering and code templates for generating tailored dashboards.

The feasibility of connecting external algorithms' outcomes to a software product line's configuration process using code templates has been tested.

Choosing the right implementation technique is a complex task, because several factors must be taken into account: the aforementioned granularity level, the understandability, and maintainability of the code, the viability of the technique, etc. Having the power of customizing their features at fine-grained level could be highly valuable, as dashboards usually ask to be user-tailored in order to provide useful support for particular and individual goals.

Code templates not only enable fine-grained variability points to be materialized at code-level but also allows the introduction of external algorithms, like artificial intelligence or visual mapping algorithms within the rendering process. The parameterization of code through Jinja2 templates lets the generator to be fed with configuration files no matter their source (if provided with a fixed syntax).

Fine-grained features are very important in a domain like this, changing the mark type or some encoding channel could be decisive for users to accomplish their goals regarding datasets. Code templates allow flexibility and even expression-level variability, which are powerful features for this domain.

Although this approach still lacks powerful maintainability levels, it maintains good requirements traceability by arranging features in a variety of macro definitions. Using XVCL [22] could have been another solution to manage these fine-grained features, but the decision of wrapping the SPL specification through a DSL asked for a more flexible and customizable method such as a template engine.

What is more, a combination of the AOP paradigm with the templating method could be highly beneficial providing both customization capabilities regarding directives and a better technique to manage crosscutting concerns (an issue that a template engine could not solve straightforwardly).

However, although presenting some caveats, the results are promising and prove that a powerful template engine could be a beneficial method to materialize fine-grained variability and to introduce external configuration algorithms within the SPL paradigm context.

Regarding the introduction of artificial intelligence, a hybrid approach for configuring the product line is proposed. The reason behind this approach is that, while

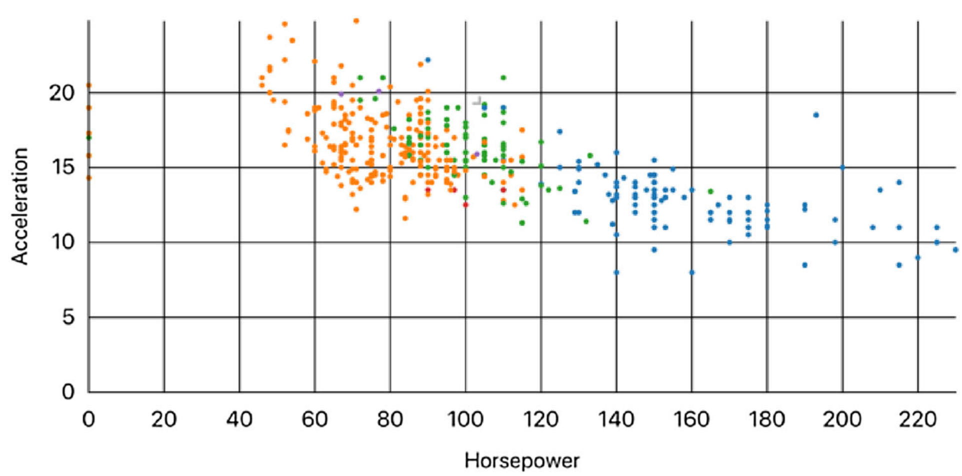


Fig. 10 Generated scatter plot using Semiotic's code

artificial intelligence is powerful for predicting values given an input set of features, visual mapping ensures that basic guidelines and constraints are followed.

Artificial intelligence could be useful for identifying user-profiles and to "translate" these profiles into a set of visualization features. But, in the end, a series of constraints regarding the construction of graphics and visualizations must be followed. That is why this work proposes predicting high-level visualization features, like the number of channels or visualizations within a dashboard, and then using visual mapping to convert this abstract information into concrete visualizations.

Moreover, a meta-model has been developed to extract commonalities from the domain and to understand the relationships among the identified elements. The user has been included in the meta-model as they are the final consumers of the dashboards. Knowing and understanding their characteristics, backgrounds, preferences, expertise, goals, etc., is crucial to deliver dashboards that support their intentions in an effective manner.

However, there are also challenges regarding the introduction of user characteristics and goals in this kind of AI pipelines. On the one hand, user goals are identified and expressed in natural language, meaning that they need preprocessing before using them to predict values. However, structuring goals is not trivial. There exist different frameworks for identifying and linking goals with low-level tasks that could be very useful for this matter [54–56]. On the other hand, user characteristics like expertise, visualization literacy, bias, are not only difficult to structure but to identify and measure, although there are also frameworks that could help with these tasks [57–59].

Another challenge is that this approach needs to be tested with real-world data and seek for improvements, but the functionality of the workflow seems promising.

However, not only tests with real-world data need to be performed. Users are the backbone and the target of visualization tools. Visualization recommendation approaches aim at boosting their capacity to reach insights. Delivering a functional dashboard is important, but in the end, the metrics that prove that the approach is useful are user-related (usability, engagement, acceptance, insights reached, etc.).

Testing an automatically generated product can refine not only the product itself but also the whole generation process by obtaining more feedback data that could feed AI algorithms.

7 Conclusions

This paper describes a proof-of-concept for a dashboard generator that uses users and datasets characteristics.

The functionality of the template-based generator has been tested to prove the feasibility of the approach and the necessity of taking into account fine-grained features in a complex domain like dashboards'. The configuration process proposal takes into account the user goals and characteristics and not only the dataset schema to infer high-level visualization features.

However, the proposed workflow needs to be tested in depth to validate not only the functionality, but the technological acceptance and usability of the generated dashboards. Users are the final consumers of these tools, so they must be involved in the validation of the workflow.

Future research lines will involve the implementation of a more comprehensive framework to generate functional visualizations using fine-grained features, as well as the testing of this approach with real-world data and users, and the iterative improvement of the workflow, to predict more visualizations' features, and thus obtaining a more powerful tailoring process for dashboards.

Acknowledgements This research work has been supported by the Spanish *Ministry of Education and Vocational Training* under an FPU fellowship (FPU17/03276). This work has been partially funded by the Spanish Government Ministry of Economy and Competitiveness throughout the DEFINES project (Ref. TIN2016-80172-R), the PROVIDEDH project, funded within the CHIST-ERA Programme under the national grant agreement: PCIN-2017-064 (MINECO, Spain) and the Ministry of Education of the Junta de Castilla y León (Spain) throughout the T-CUIDA project (Ref. SA061P17).

References

1. Albright, S.C., Winston, W., Zappe, C.: Data Analysis and Decision Making. Cengage Learning, Mason (2010)
2. Few, S.: Information Dashboard Design. O'Reilly Media, Sebastopol (2006)
3. Vázquez-Ingelmo, A., Garcia-Peñalvo, F.J., Therón, R.: information dashboards and tailoring capabilities: a systematic literature review. IEEE Access **7**, 109673–109688 (2019). https://doi.org/10.1109/ACCESS.2019.2933472
4. Clements, P., Northrop, L.: Software Product Lines. Addison-Wesley, Boston (2002)
5. Pohl, K., Böckle, G., Van der Linden, F.J.: Software Product Line Engineering: Foundations, Principles and Techniques. Springer, New York (2005)
6. Gomaa, H.: Designing Software Product Lines with UML: From Use Cases to Pattern-Based Software Architectures. Addison Wesley Longman Publishing Co., Inc., Boston (2004)
7. Kramer, D., Oussena, S., Komisarczuk, P., Clark, T.: Using document-oriented GUIs in dynamic software product lines. In: ACM SIGPLAN Notices, New York, vol. 3, pp. 85–94 (2013)
8. Marinho, F.G., Lima, F., Ferreira Filho, J.B., Rocha, L., Maia, M.E., de Aguiar, S.B., Dantas, V.L., Viana, W., Andrade, R.M., Teixeira, E.: A software product line for the mobile and context-aware applications domain. In: International Conference on Software Product Lines, pp. 346–360. Springer, New York (2010)
9. Ezzat Labib Awad, A.: Enforcing Customization in e-Learning Systems: An Ontology and Product Line-Based Approach. Universitat Politècnica de València, Valencia (2017)

10. Gabillon, Y., Biri, N., Otjacques, B.: Designing an adaptive user interface according to software product line engineering. In: Paper Presented at the Eighth International Conference on Advances in Computer–Human Interactions, Lisbon, Portugal
11. Pleuss, A., Hauptmann, B., Keunecke, M., Botterweck, G.: A case study on variability in user interfaces. In: Proceedings of the 16th International Software Product Line Conference-vol. 1, pp. 6–10. ACM (2012)
12. Quinton, C., Mosser, S., Parra, C., Duchien, L.: Using multiple feature models to design applications for mobile phones. In: Proceedings of the 15th International Software Product Line Conference, vol. 2, p. 23. ACM (2011)
13. Achtaich, A., Ounsa, R., Nissrine, S., Camille, S.: Selecting SPL modeling languages: a practical guide. In: 2015 Third World Conference on Complex Systems (WCCS), pp. 1–6. IEEE (2015)
14. Kang, K.C., Cohen, S.G., Hess, J.A., Novak, W.E., Peterson, A.S.: Feature-oriented domain analysis (FODA) feasibility study. Carnegie-Mellon University, Software Engineering Institute, Pittsburgh (1990)
15. Gacek, C., Anastasopoules, M.: Implementing product line variabilities. In: ACM SIGSOFT Software Engineering Notes, New York, vol. 3, pp. 109–117. ACM (2001)
16. Kästner, C., Apel, S., Kuhlemann, M.: Granularity in software product lines. In: Proceedings of the 30th International Conference on Software Engineering, pp. 311–320. ACM (2008)
17. Vázquez-Ingelmo, A., García-Peñalvo, F.J., Therón, R.: Taking advantage of the software product line paradigm to generate customized user interfaces for decision-making processes: a case study on university employability. PeerJ Comput. Sci. **5**, e203 (2019). https://doi.org/10.7717/peerj-cs.203
18. Logre, I., Mosser, S., Collet, P., Riveill, M.: Sensor data visualisation: a composition-based approach to support domain variability. In: European Conference on Modelling Foundations and Applications, pp. 101–116. Springer, New York (2014)
19. Liebig, J., Apel, S., Lengauer, C., Kästner, C., Schulze, M.: An analysis of the variability in forty preprocessor-based software product lines. In: Proceedings of the 32nd ACM/IEEE International Conference on Software Engineering, vol. 1, pp. 105–114. ACM (2010)
20. Favre, J.-M.: Preprocessors from an abstract point of view. In: Reverse Engineering, Proceedings of the Third Working Conference on 1996, pp. 287–296. IEEE (1996)
21. Spencer, H., Collyer, G.: #ifdef considered harmful, or portability experience with C News (1992)
22. Jarzabek, S., Bassett, P., Zhang, H., Zhang, W.: XVCL: XML-based variant configuration language. In: Proceedings of the 25th International Conference on Software Engineering, pp. 810–811. IEEE Computer Society (2003)
23. Zhang, H., Jarzabek, S., Swe, S.M.: XVCL approach to separating concerns in product family assets. In: International Symposium on Generative and Component-Based Software Engineering, Efurt, pp. 36–47. Springer, New York (2001)
24. Clark, S.: Render your first network configuration template using Python and Jinja2. https://blogs.cisco.com/developer/network-configuration-template (2018)
25. Kiczales, G., Lamping, J., Mendhekar, A., Maeda, C., Lopes, C., Loingtier, J.-M., Irwin, J.: Aspect-oriented programming. In: European Conference on Object-Oriented Programming, pp. 220–242. Springer, New York (1997)
26. Waku, G.M., Rubira, C.M., Tizzei, L.P.: A case study using aop and components to build software product lines in android platform. In: Software Engineering and Advanced Applications (SEAA), 2015 41st Euromicro Conference on 2015, pp. 418–421. IEEE (2015)
27. Heo, S.-H., Choi, E.M.: Representation of variability in software product line using aspect-oriented programming. In: Fourth International Conference on Software Engineering Research, Management and Applications, pp. 66–73. IEEE (2006)
28. Voelter, M., Groher, I.: Product line implementation using aspect-oriented and model-driven software development. In: 11th International Software Product Line Conference. SPLC 2007, pp. 233–242. IEEE. (2007)
29. Kaur, P., Owonibi, M.: A review on visualization recommendation strategies. In: 12th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP 2017), pp. 266–273. SCITEPRESS (2017)
30. Mackinlay, J., Hanrahan, P., Stolte, C.: Show me: automatic presentation for visual analysis. IEEE Trans. Vis. Comput. Graph. **13**(6), 1137–1144 (2007)
31. Viegas, F.B., Wattenberg, M., Van Ham, F., Kriss, J., McKeon, M.: Manyeyes: a site for visualization at internet scale. IEEE Trans. Vis. Comput. Graph. **13**(6), 1121–1128 (2007)
32. Wongsuphasawat, K., Moritz, D., Anand, A., Mackinlay, J., Howe, B., Heer, J.: Voyager: exploratory analysis via faceted browsing of visualization recommendations. IEEE Trans. Vis. Comput. Graph. **22**(1), 649–658 (2015)
33. Key, A., Howe, B., Perry, D., Aragon, C.: Vizdeck: self-organizing dashboards for visual analytics. In: Proceedings of the 2012 ACM SIGMOD International Conference on Management of Data, pp. 681–684. ACM (2012)
34. Voigt, M., Pietschmann, S., Grammel, L., Meißner, K.: Context-aware recommendation of visualization components. In: The Fourth International Conference on Information, Process, and Knowledge Management (eKNOW), pp. 101–109. Citeseer (2012)
35. Gotz, D., Zhou, M.X.: Characterizing users' visual analytic activity for insight provenance. Inf. Vis. **8**(1), 42–55 (2009)
36. Gotz, D., Wen, Z.: Behavior-driven visualization recommendation. In: Proceedings of the 14th International Conference on Intelligent User Interfaces, pp. 315–324. ACM (2009)
37. Mutlu, B., Veas, E., Trattner, C.: Vizrec: recommending personalized visualizations. ACM Trans. Interact. Intell. Syst. **6**(4), 31 (2016)
38. Hu, K., Bakker, M.A., Li, S., Kraska, T., Hidalgo, C.: VizML: a machine learning approach to visualization recommendation. In: Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems, p. 128. ACM (2019)
39. Dibia, V., Demiralp, Ç.: Data2Vis: automatic generation of data visualizations using sequence to sequence recurrent neural networks. IEEE Comput. Graph. Appl. **39**, 33–46 (2019)
40. Pleuss, A., Wollny, S., Botterweck, G.: Model-driven development and evolution of customized user interfaces. In: Proceedings of the 5th ACM SIGCHI Symposium on Engineering interactive computing systems, pp. 13–22. ACM (2013)
41. Kleppe, A.G., Warmer, J., Bast, W.: MDA Explained. The Model Driven Architecture: Practice and Promise. Addison-Wesley Longman Publishing Co., Inc., Boston (2003)
42. Álvarez, J.M., Evans, A., Sammut, P.: Mapping between levels in the metamodel architecture. In: International Conference on the Unified Modeling Language, pp. 34–46. Springer, New York (2001)
43. Vázquez-Ingelmo, A., García-Peñalvo, F.J., Therón, R.: Capturing high-level requirements of information dashboards' components through meta-modeling. In: Paper Presented at the 7th International Conference on Technological Ecosystems for Enhancing Multiculturality (TEEM 2019), León (2019)
44. Vázquez Ingelmo, A., García-Peñalvo, F.J., Therón, R., Conde González, M.Á.: Extending a dashboard meta-model to account for users' characteristics and goals for enhancing personalization. In: Paper Presented at the Learning Analytics Summer Institute (LASI) Spain 2019, Vigo (2019)

45. Vázquez-Ingelmo, A., García-Peñalvo, F.J., Therón, R.: Addressing fine-grained variability in user-centered software product lines: a case study on dashboards. In: World Conference on Information Systems and Technologies, pp. 855–864. Springer, New York (2019)
46. Osoba, O.A., Welser IV, W.: An Intelligence in Our Image: The Risks of Bias and Errors in Artificial Intelligence. Rand Corporation, Santa Monica (2017)
47. Sayyad, A.S., Ingram, J., Menzies, T., Ammar, H.: Scalable product line configuration: a straw to break the camel's back. In: 2013 28th IEEE/ACM International Conference on Automated Software Engineering (ASE), 11–15 Nov 2013, pp. 465–474 (2013)
48. White, J., Schmidt, D.C., Benavides, D., Trinidad, P., Ruiz–Cortés, A.: Automated diagnosis of product-line configuration errors in feature models. In: 2008 12th International Software Product Line Conference, 8–12 Sept 2008, pp. 225–234 (2008)
49. Elsner, C., Ulbrich, P., Lohmann, D., Schröder-Preikschat, W.: Consistent product line configuration across file type and product line boundaries. In: International Conference on Software Product Lines, pp. 181–195. Springer, New York (2010)
50. Wongsuphasawat, K., Qu, Z., Moritz, D., Chang, R., Ouk, F., Anand, A., Mackinlay, J., Howe, B., Heer, J.: Voyager 2: Augmenting visual analysis with partial view specifications. In: Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems, pp. 2648–2659. ACM (2017)
51. Wongsuphasawat, K., Moritz, D., Anand, A., Mackinlay, J., Howe, B., Heer, J.: Towards a general-purpose query language for visualization recommendation. In: Proceedings of the Workshop on Human-In-the-Loop Data Analytics, p. 4. ACM (2016)
52. Berinato, S.: Good Charts: The HBR Guide to Making Smarter, More Persuasive Data Visualizations. Harvard Business Review Press, Brighton (2016)
53. Meeks, E.: Introducing Semiotic for Data Visualization. https://medium.com/@Elijah_Meeks/introducing-semiotic-for-data-visualization-88dc3c6b6926 (2017)
54. Lam, H., Tory, M., Munzner, T.: Bridging from goals to tasks with design study analysis reports. IEEE Trans. Vis. Comput. Graph. **24**(1), 435–445 (2017)
55. Munzner, T.: A nested process model for visualization design and validation. IEEE Trans. Vis. Comput. Graph. **6**, 921–928 (2009)
56. Brehmer, M., Munzner, T.: A multi-level typology of abstract visualization tasks. IEEE Trans. Vis. Comput. Graph. **19**(12), 2376–2385 (2013)
57. Lee, S., Kim, S.-H., Kwon, B.C.: Vlat: development of a visualization literacy assessment test. IEEE Trans. Vis. Comput. Graph. **23**(1), 551–560 (2017)
58. Boy, J., Rensink, R.A., Bertini, E., Fekete, J.-D.: A principled way of assessing visualization literacy. IEEE Trans. Vis. Comput. Graph. **20**(12), 1963–1972 (2014)
59. Bedek, M.A., Nussbaumer, A., Huszar, L., Albert, D.: Methods for discovering cognitive biases in a visual analytics environment. In: Cognitive Biases in Visualizations, pp. 61–73. Springer, New York (2018)

Andrea Vázquez-Ingelmo received the bachelor's degree in computer engineering from the University of Salamanca, Salamanca, in 2016 and the master's degree in computer engineering from the same university in 2018. She is a member of the Research Group of Interaction and eLearning (GRIAL), where she is pursuing her Ph.D. degree in computer sciences. Her area of research is related to human–computer interaction, software engineering, information visualization and machine learning applications.

Francisco José García-Peñalvo received the degrees in computing from the University of Salamanca and the University of Valladolid, and the Ph.D. degree from the University of Salamanca. He is currently the Head of the Research Group Interaction and eLearning and Full Professor in the Computer Science Department at the University of Salamanca. He has led and participated in over 50 research and innovation projects. He was the Vice Chancellor of Innovation with the University of Salamanca from 2007 to 2009. He has authored over 300 articles in international journals and conferences. His main research interests focus on eLearning, computers and education, adaptive systems, Web engineering, semantic Web, and software reuse. He was a Guest Editor of several special issues of international journals, such as *Online Information Review*, *Computers in Human Behavior*, and *Interactive Learning Environments*. He is also a member of the Program Committee of several international conferences and a Reviewer of several international journals. He is currently the Editor-in-Chief of the *International Journal of Information Technology Research* and the *Education in the Knowledge Society Journal*. Besides, he is the Coordinator of the multidisciplinary Ph.D. Programme on Education in the Knowledge Society.

Roberto Therón received the Diploma degree in computer science from the University of Salamanca, the B.A. degree from the Universidade da Coruña, the bachelor's degrees in communication studies and humanities from the University of Salamanca, and the Ph.D. degree from the Research Group Robotics, University of Salamanca. His Ph.D. thesis was on parallel calculation configuration space for redundant robots. He is currently the Manager of the VisUSAL Group (within the Recognized Research Group GRIAL), University of Salamanca, which focuses on the combination of approaches from computer science, statistics, graphic design, and information visualization to obtain an adequate understanding of complex data sets. He has authored over 100 articles in international journals and conferences. In recent years, he has been involved in developing advanced visualization tools for multidimensional data, such as genetics or paleo-climate data. In the field of visual analytics, he develops productive collaborations with groups and institutions internationally recognized as the Laboratory of Climate Sciences and the Environment, France, or the Austrian Academy of Sciences, Austria. He received the Extraordinary Doctoral Award for his Ph.D. thesis.

Daniel Amo Filvà is a Technical Engineer in Computer Science (EUPMT), with a Master's Degree in Education and ICT (UOC) and a Master's Degree in Teacher Training in Secondary Education (UNIR). For over 15 years he has been leading information management projects in companies. Since 7 years ago he has been Coordinator and ICT Manager in different schools, Advisor of national projects on integration of ICT in educational institutions and Teacher trainer and researcher in Learning Analytics. Currently working in La Salle Campus Barcelona (Universitat Ramon Llull) as a teacher. He is part of the EduEnginy's team, a project to promote engineering thinking and social STEAM education through robotics. His main research interests combine: learning analytics, education, STEAM and robotics. Author of the book "Learning Analytics: La narración del aprendizaje a través de los datos".

David Fonseca Escudero is a Technical Engineer in Telecommunications (Image and Sound) by Ramon Llull (1998), Associated Professor of University School—(2002), International Master in Geographic Information Systems, by the University of Girona (2003), Degree in Audiovisual Communication from the Universitat Oberta de Catalunya (2006), Diploma of Advanced Studies by La Salle (2007), Official Master's Degree in Information and Knowledge Society, by the UOC (2009). Doctor by the URL (2011), Accreditation AQU Reader (2014), Accreditation Aggregate/Research AQU (2015), Associated Professor of the University by the URL (2015), Full Professor of the University by the URL (2017), 1st Sexennial of Research by the AQU (2015), period (2014 to 2009). He is also Autodesk Approved Certified Instructor (1998-ACT) in CAD, MAP, REVIT, Autodesk Professional Certified in Autocad.

7.15 Appendix O. Representing Data Visualization Goals and Tasks Through Meta-Modeling to Tailor Information Dashboards

applied sciences

MDPI

Article

Representing Data Visualization Goals and Tasks Through Meta-Modeling to Tailor Information Dashboards

Andrea Vázquez-Ingelmo [1,*], Francisco José García-Peñalvo [1], Roberto Therón [2] and Miguel Ángel Conde [3]

1. GRIAL Research Group, Computer Science Department, University of Salamanca, 37008 Salamanca, Spain; fgarcia@usal.es
2. VisUSAL, GRIAL Research Group, Computer Science Department, University of Salamanca, 37008 Salamanca, Spain; theron@usal.es
3. Department of Mechanics, Computer Science and Aerospace Engineering, University of León, 24007 León, Spain; miguel.conde@unileon.es

* Correspondence: andreavazquez@usal.es; Tel.: +34-92-329-4500 (ext. 3433)

Received: 28 January 2020; Accepted: 26 March 2020; Published: 27 March 2020

Abstract: Information dashboards are everywhere. They support knowledge discovery in a huge variety of contexts and domains. Although powerful, these tools can be complex, not only for the end-users but also for developers and designers. Information dashboards encode complex datasets into different visual marks to ease knowledge discovery. Choosing a wrong design could compromise the entire dashboard's effectiveness, selecting the appropriate encoding or configuration for each potential context, user, or data domain is a crucial task. For these reasons, there is a necessity to automatize the recommendation of visualizations and dashboard configurations to deliver tools adapted to their context. Recommendations can be based on different aspects, such as user characteristics, the data domain, or the goals and tasks that will be achieved or carried out through the visualizations. This work presents a dashboard meta-model that abstracts all these factors and the integration of a visualization task taxonomy to account for the different actions that can be performed with information dashboards. This meta-model has been used to design a domain specific language to specify dashboards requirements in a structured way. The ultimate goal is to obtain a dashboard generation pipeline to deliver dashboards adapted to any context, such as the educational context, in which a lot of data are generated, and there are several actors involved (students, teachers, managers, etc.) that would want to reach different insights regarding their learning performance or learning methodologies.

Keywords: information dashboards; metamodeling; visualization goals; visualization tasks; data visualization; educational dashboards

1. Introduction

Information is the driver of many processes and activities nowadays. Designers, managers, developers, etc., make decisions continuously with the goal of obtaining a benefit or a pursued effect on the context of application. Data-driven decision-making [1] has several advantages, but it is also a complex process in which the involved actors must understand the data they are consuming.

Data can hold underlying patterns that could go unnoticed without performing in-depth analysis. However, the great quantity of available data can make the analysis process a resource- and time-consuming task. For this reason, many tools have emerged to support and ease the discovery of insights through data from different domains [2-6].

Information dashboards are one of these tools. Information dashboards arrange data points into different views, displaying information visually, and allowing users to identify patterns, anomalies, and relationships among variables in a straightforward manner.

However. using an information dashboard does not ensure knowledge generation. There are several factors involved in the process of discovering insights through displayed data, and one of the most important factors that condition the utility of dashboards is their audience (i.e., their end-users).

People have several characteristics that could make their user experience using the same product (in this case, an information dashboard) very different from each other. For example, some users might not have enough visualization literacy (also known as graphicacy [7, 8]), and thus, they might be unable to understand some charts or encodings. On the other hand, their knowledge regarding the data's domain could be low, hindering their ability to reach meaningful insights.

Consequently, dashboards need to be adapted not only to the data they are displaying but also to their audience [9-11]. This is not a trivial task: the data domain and potential user profiles must be researched, among other factors, to obtain a visual display that is really useful for making decisions (this should be the first priority of a dashboard: to support decision-making or knowledge generation).

Tailoring a dashboard is not only complex at design-level (where several factors and design guidelines must be accounted for) but also at the implementation-level. Coding a tailored dashboard for each possible context is a time-consuming process.

Different approaches have been considered in the literature to automatize this implementation process and to decrease the development time of tailored dashboards [12]. From configuration wizards that allow users to choose the charts of their dashboards (e.g., Tableau, https://www.tableau.com/, or Grafana, https://grafana.com/) to model-driven approaches that render personalized dashboards based on formal descriptions of the domain [13-16], among other variety of solutions.

These approaches take into account mainly user preferences, but also the input data structures, business processes, user abilities, user roles, etc. [17]. These factors are extremely relevant to the design process; they can support the selection of appropriate visual metaphors, encodings, or structure of the dashboard to increase its effectiveness and usability.

In fact, users might need not only differing visual metaphors to understand the same dataset, but also a whole different composition of views that hold different data variables. This happens because users could be focused on different variables and could have different questions regarding the same dataset; that is, users have their own goals when referring to a dataset.

Accounting for a user's information goals is essential for the development of information dashboards; they frame and contextualize the tasks that can be carried out with data, as well as the variables that should be involved within the display. However, this information needs to be properly structured to enable its analysis and processing and use it in a dashboard generative pipeline to obtain tailored dashboards automatically.

This paper discusses the main factors that need to be accounted for in dashboard design and extends a dashboard meta-model that identifies core relationships and entities within this complex domain [18-20]. The previously developed meta-model aimed at formalizing a structure for defining information dashboards based on a set of factors, such as the data structure or users' goals, preferences, domain knowledge, visualization literacy, etc. The presented extension focuses on how to structure the users' goals and tasks with the purpose of accounting for these factors in a generative pipeline of dashboards.

Automatizing the generation of information dashboards requires a robust conceptualization, because, in the end, the entities and attributes present in the inputs of the generative process will condition the outputs. The main outcome of this conceptualization work is the definition of a domain specific language (DSL) based on the meta-model with the purpose of using it to materialize abstract dashboard features into specific products. Relying on the meta-model facilitates not only the comprehension of the domain but also sets the first milestone for obtaining a dashboard generative pipeline in which the input is based on the meta-model structure. The main goal is to provide information dashboards adapted to their context.

One of the contexts that could benefit from this approach is the educational context. Educational dashboards [21] are powerful tools for identifying patterns and relationships among learning variables. There could be several roles involved in this context, from students and teachers to managers. These roles can ask for different learning variables and indicators, depending on their needs [22-25]. For example, a teacher could want to reach insights regarding the performance of their students in order to improve his learning methodologies, while a student just want to track her achievements.

Furthermore, users with the same role could be interested in very different aspects of their data, hampering the whole process of designing an educational dashboard. Using a meta-model to organize the requirements of educational dashboards based on the audience could improve the development process, as well as the reuse of the gained knowledge from accounting users' characteristics in subsequent designs.

The rest of this paper is organized as follows: Section 2 contextualizes the relevance of tailored visualizations and dashboards. Section 3 presents the methods used to carry out the study. Section 4 describes the proposed dashboard meta-model, including information about users' goals and tasks. Section 5 discusses the meta-model, while Section 6 outlines the conclusions derived from this work.

2. Background

Information dashboards and visualizations have increased their popularity throughout the years. They enable people to understand complex datasets and gain insights into different domains. However, these tools are not suitable for any context and need to match specific requirements in different situations. Given the necessity and potential benefits of tailored dashboards, several solutions, and proposals to address dashboards' tailoring capabilities can be found in the literature [12].

Selecting the right visual metaphor or encoding is challenging when dealing with dashboards or visualizations. However, this task is essential due to the influence of these design decisions in the effectiveness of the dashboard or visualizations, because a wrong visual metaphor or encoding could lead to mistakes when interpreting data.

Several tools have been proposed to tackle this issue by automatically recommending visualizations. These works point out different factors that influence the specific design of an information visualization, such as the tasks that will be carried out through the visualization, the users' characteristics, the users' behavior, or the dataset structure and characteristics [26].

How can these aspects support an automatic process for recommending the best encodings or visual metaphors to foster knowledge discovery? There are different approaches that try to ease the process of generating visualizations. For example, some methods use visual mapping and rules to recommend a certain visualization based on the target data to be displayed [26]. An algorithm could infer through hardcoded guidelines and rules which visual mark, scale, encoding, etc., suits better the target context by using information regarding the dataset characteristics, such as its structure or data types. Tableau's Show Me [27], Manyeyes [28], or Voyager [29], are some of the tools that use visual mapping to offer tailored recommendations for information visualizations.

The context of application is also a relevant factor when designing visualizations. For example, the work presented in [30] employs a visualization ontology named VISO to annotate data and execute a ranking process. The outcomes of the ranking process are ratings that measure the suitability of visual encodings. End-users' preferences and characteristics are also taken into account in some visualization recommendation strategies. VizDeck [31] analyzes the input data and proposes a set of visualization that the user must rank, selecting the ones that fit better her requirements. This process supports subsequent recommendations because the system learns from the users' interactions.

On the other hand, content and collaborative filtering can also be employed to recommend visualizations [32]. These recommendation strategies yield potentially suitable visualizations for specific users.

Applications of neural networks to infer specific features of a visualization can also be found. For example, VizML [33] used data from Plotly (https://plot.ly/) to analyze graphics that were developed to visualize different datasets. Using this Plotly's graphics corpus, they obtained a model that automatically infers the best characteristics that a visualization should have based on the input data. A similar approach is taken in Data2Vis [34], where data characteristics are "translated" into a concrete visualization specification.

However, these approaches are mainly focused on users' or data characteristics. Some works propose to take into account visualization tasks to rank the effectiveness of two-dimensional visualization types in order to choose the best for each task [35]. As introduced before, the users' goals are crucial to craft effective visualizations, so the potentially involved tasks to reach them must be effectively supported by the generated visualization [36].

3. Materials and Methods

3.1. Metamodeling

The model-driven development (MDD) paradigm [15, 37] enables the abstraction of the requirements involved in the development process of information systems, moving both data and operations specifications away from concrete and lower-level details. The main benefit of abstracting these details is to obtain a meta-model that holds a set of structures and rules shared by any system from the modeled domain. In other words, the meta-model can be employed to drive the development of different systems by instantiating abstract features into specific features. This methodology increases the reuse of components (thus, decreasing the development time), but also the reuse of knowledge because the structures and relationships identified during the development of the meta-model can evolve to obtain better solutions.

The Object Management Group (OMG) proposes the model-driven architecture (MDA) as a guideline to implement the MDD approach. This architecture provides a framework for software development in which the process is driven by models that describe and define the target system [38]. The main difference between MDD and MDA is that MDA determines a set of standards to develop the approach, such as meta-object facility (MOF), unified modeling language (UML), XML (Extensible Markup Language) metadata interchange (XMI), and query/view/transformation (QVT).

The dashboard meta-model is also part of the four-layer meta-model architecture proposed by the OMG, in which a model at one layer is used to specify models in the layer below [39]. In particular, the first version of the dashboard meta-model [20] was an instance of MOF (i.e., an M2-model), so it can be instantiated to obtain M1-models. This meta-model was transformed in an instance of Ecore [40] using Graphical Modelling for Ecore included in Eclipse Modeling Framework (EMF), in order to leverage the different features of this modeling framework (Figure 1).

This meta-model was developed using a domain engineering approach [41, 42], in which similarities and variability points were identified to obtain an abstract picture of the dashboards' domain in terms of these tools' elements and features.

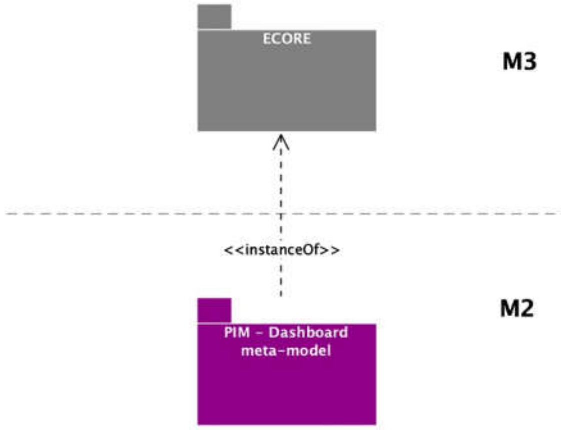


Figure 1. Location of the dashboard meta-model following the OMG architecture.

The goal of the dashboard meta-model is to develop and deploy information dashboards in a variety of real-world contexts. Figure 2 shows a reduced version of the mentioned dashboard meta-model, which can be decomposed in three main elements: the user, the layout, and the components. Figure 3 shows the detailed view of the components' section that was omitted in Figure 2 due to legibility reasons (the whole high-resolution meta-model is available at https://doi.org/10.5281/zenodo.3561320). As can be seen, the dashboard is composed of different pages, which, in turn, is composed of a set of containers that can be organized in rows and columns. These containers hold the dashboard components.

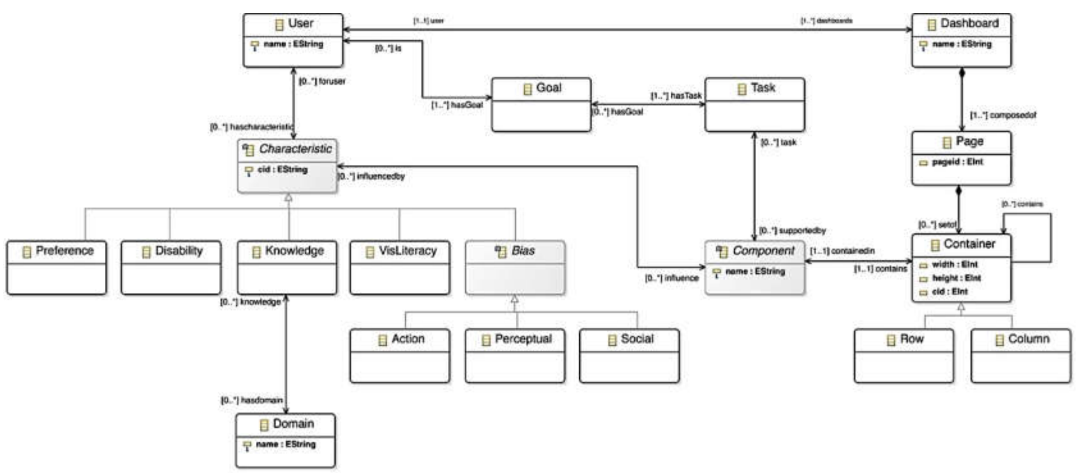


Figure 2. Overview of the dashboard meta-model, including the user, the layout, and the components. This image is available in high resolution at https://doi.org/10.5281/zenodo.3561320. Licensed under CC BY 4.0.

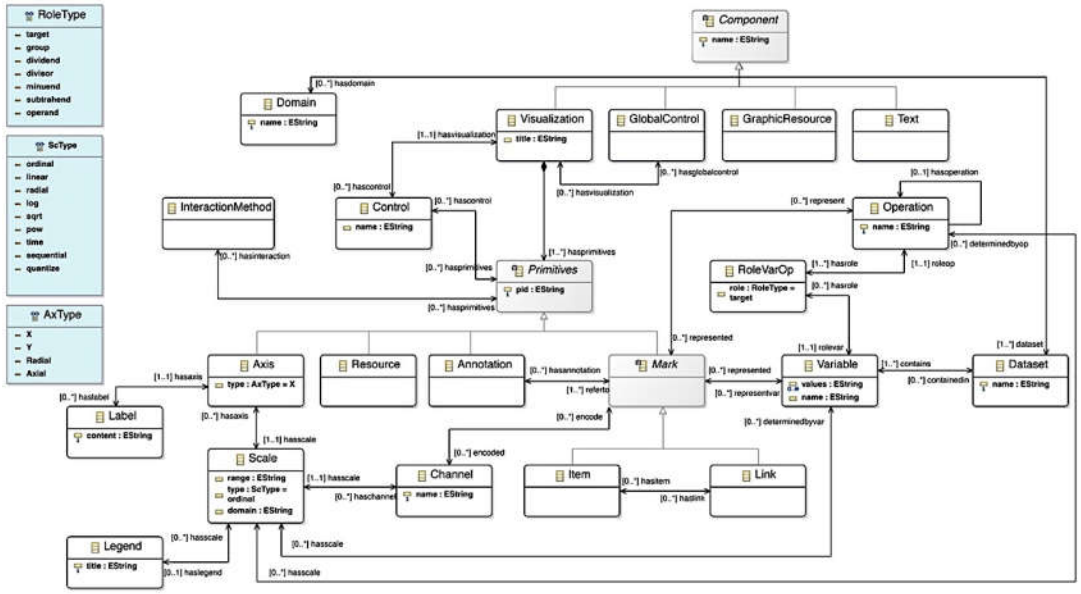


Figure 3. Detailed view of the dashboard meta-model components' definition. This image is available in high resolution at https://doi.org/10.5281/zenodo.3561320. Licensed under CC BY 4.0.

On the other hand, the user is modeled as an entity with two main elements that define her behavior: goals and characteristics. Goals refer to the purposes of the user regarding the displayed data, and they can be broken down into different lower-level analytic tasks that must be supported by the dashboard components in order to reach the identified goals.

Finally, users have different characteristics that also influence the components that form the dashboard. These characteristics include preferences, disabilities, knowledge about the data domain, visualization literacy (or graphicacy [8]), and bias. As will be discussed, it is necessary to account for these characteristics to offer users a tailored dashboard that enables them to reach their analytic goals effectively and with good user experience.

The next section presents an extension of this dashboard meta-model with the aim of holding more information regarding information visualization goals and tasks. The purpose of this extension is to draw attention to the influence that the users' goals have on the components and the functionality of information dashboards. Including this information can support and improve the adaptation of these tools to concrete users, data domains and contexts.

3.2. Visualization Tasks' Taxonomies

Users may have very different intentions or objectives when facing an information visualization or information dashboard. These intentions or purposes define their information goals (i.e., what does the user want to know or discover by using a visualization?). Identifying the audience of the tool and their goals is crucial to design an effective visualization or dashboard [9, 11]. However, visualization goals and tasks are usually tightly coupled to their domains. In order to include this information into a meta-model, it is necessary to obtain abstract definitions of generic goals and tasks that can be instantiated into any domain.

This issue has already been tackled by visualization researchers and practitioners, trying to transform domain-specific tasks into abstract tasks to understand better the different actions that users could take when using dashboards or visualizations and evaluate them. Relying on abstract tasks can help researchers design more effective and efficient components that boost user performance in decision-making processes.

Amar et al. [43] carried out an experiment in which the participants would analyze datasets from different domains. By using affinity diagrams and grouping similar questions raised by the participants, they identified ten low-level analytic tasks, including retrieving a value, filtering, computing a derived value, finding an extremum, sorting, determining a range, characterizing a distribution, finding anomalies, clustering and correlating. Schultz et al. [44] described five dimensions to characterize tasks: goal (the intent of the task), means (method used to reach the goal), characteristics (referring to the data aspects), target (of the analytic task) and cardinality (referring to the scope of the task). These five dimensions enable the definition of individual and compound tasks through 5-tuples. Gotz and Zhou [45] characterized analytic behavior when facing data visualization through multiple levels of granularity: tasks, sub-tasks, actions, and events. Tasks hold richer semantic value, because they represent the users' analytic goals, while events are isolated interactions (hover, click, selections, etc.) that don't possess semantic value, but are essential to reach the analytic goals.

Dimara et al. [46] even utilized a task-based taxonomy to organize different cognitive biases that can be associated with information visualization analytic activity. The identified experimental tasks are estimation, decision, hypothesis assessment, causal attribution, recall, opinion reporting, and others. On the other hand, Munzner described different levels of actions to define user goals regarding information visualizations [47]. In this case, there are three levels of actions: analyze, search, and query. Each of these actions is broken down into more detailed goals, such as annotate, record, discover, enjoy, derive, browse, lookup, compare, etc. [48].

In [49], an analytical goals' classification is proposed to bridge the gap between goals (the questions asked) and tasks (the steps needed to answer the questions). The analysis goals framework consists of nine goals arranged into two axes (the specificity of the goal and the number of populations under consideration): discover observation, describe observation (item), describe observation (aggregation), identify main cause (item), identify main cause (aggregation), collect evidence, compare entities, explain differences and evaluate hypothesis. This framework can be employed along with other task taxonomies, as it provides a bridge to link analysis goals to the steps to achieve them.

Abstracting data visualization tasks is seen as a first step to select an appropriate encoding or interaction method; it is necessary to translate domain-specific questions to generic tasks [50] to provide users with effective visual analysis tools. The taxonomy employed to define the goals' and tasks' space in this work is Amar et al.'s [43]. The main reason for using this specific taxonomy is due to its low-level nature and its widespread use in visualization evaluation. Moreover, this taxonomy can also be used along with the analysis goals framework [49], providing a complete definition of the analysis context.

3.3. Domain Specific Language

A Domain Specific Language (DSL) has been designed to leverage the previously described dashboard conceptualization process and to obtain a powerful asset to generate dashboards. The meta-model provides an abstract but descriptive domain specification. The identified entities, relationships and attributes can be mapped to a concrete language that enables users to understand dashboards' requirements without the necessity of having technical or programming skills. The DSL has been implemented making use of XML [51] technology. This technology provides a readable and easy-to-parse method to specify the identified domain features [52]. The grammar of the DSL can also be described through DTD or XML schema [53]. The XML schema allows the definition of rules and constraints, which are useful to ensure that the language is valid and consistent with the meta-model.

The syntax to define dashboards is completely based on the meta-model entities and relationships, ensuring the coherence between the DSL and the high-level dashboard definition. Figure 4 outlines the mapping process to create the DSL taking the meta-model as an input.

The DSL not only provides a readable manner to instantiate the meta-model into concrete dashboards, but also a structured way to identify the structure of dashboards and their components. As will be discussed, this method could allow the characterization of dashboards in terms of their primitives, and the identification of features that make visualizations effective, trustworthy or, on the other hand, misleading.

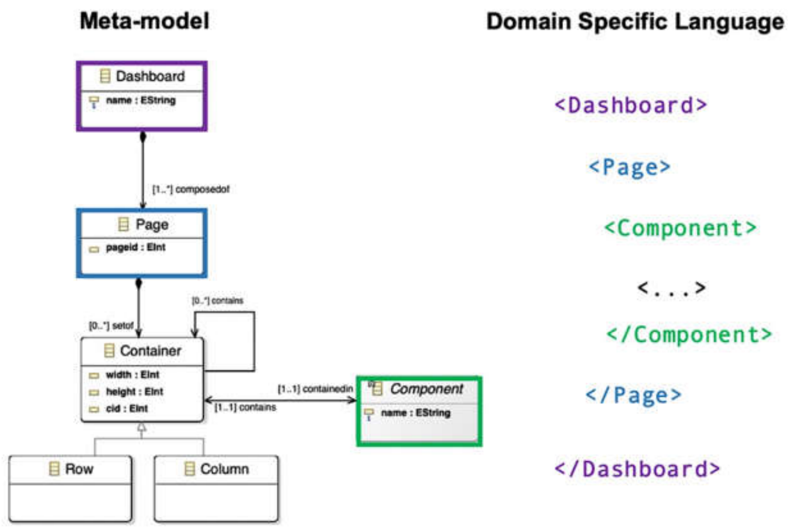


Figure 4. Correspondence between the meta-model entities and the XML entities that are part of the DSL. The subsequent primitives that are part of a component are materialized in the DSL through nested entities and properties, as will be presented in the results section.

3.4. Generation Process

The generation process leverages the DSL to map abstract entities into concrete code pieces that can be combined following the Software Product Lines (SPL) approach [54]. A template-based approach was selected to achieve an automatic generation given its flexibility and fine-grained variability [55]. A Python-based parser is used to read input configuration files (employing the DSL) and to inject the concrete dashboards features into Jinja2 code templates [56]. The result of this process is a set of JavaScript and HTML files that render a specific dashboard configuration. The following section provides examples of the generation process outputs.

4. Results

4.1. Meta-Model Extension

The first outcome of this research is an extension of the previously described dashboard meta-model. Although a slight addition, having information regarding the typology of tasks and the structure of goals is essential to build a robust model that can be instantiated into concrete products. In addition, it would also ease the process of defining data collection tools by determining the necessary data and structures that should be gathered.

First, to include the analysis goal framework [49] into the meta-model, four attributes have been added to the Goal entity. These attributes are a name to identify the goal, its specificity, and its population. The fourth attribute is a description to complement the previous information if needed. The specificity attribute is an enumeration of the four values described in [49]: explore, describe, explain, and confirm, while the population enumeration has two values: single or multiple. The included attributes characterize the user's goals and support their structuration by classifying the goal intent through its specificity and population.

Given the flexibility of the analysis goal framework and the possibility of connecting it with other existing lower-level task taxonomies, the Task class has been complemented with three attributes. Two of the attributes are also a name and a description to enrich the specification of the task. The last attribute is the task type, which can be one of the ten low-level analytical tasks depicted in [43]: retrieve value, filter, compute a derived value, find extremum, sort, determine a range, characterize distribution, find anomalies, cluster and correlate. The extension of the meta-model is shown in Figure 5.

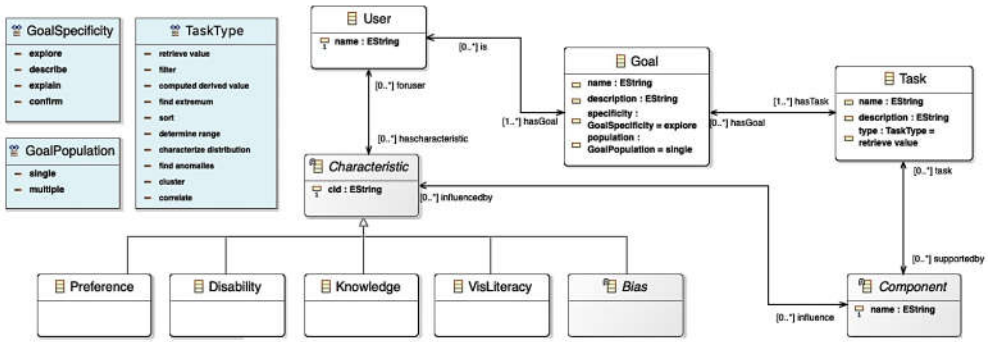


Figure 5. Extended user section of the meta-model. The rest of the meta-model has been omitted for legibility reasons. This image is available in high resolution at https://doi.org/10.5281/zenodo.3625703. Source: [57], licensed under CC BY 4.0.

4.2. Dashboard DSL

As stated before, the dashboard meta-model provides a resource to design a DSL based on the identified relationships, entities and attributes. By using XML, it is possible to materialize dashboard requirements into configuration files. Arranging requirements into structured files allows for processing of the selected features and the generation of products of certain characteristics.

Following the meta-model, an example of a visualization configuration following the DSL syntax is presented in the Figure 6. This configuration would yield a scatter chart with the x-axis representing a variable named "category1", and the y-axis representing a variable named "intensity" from the dataset selected to represent. From Figure 6, other elements represented in the meta-model are also present, such as the channels (that represent variables' values using different encodings) or the position of the component within the dashboard.

```
<Component type="visualization" component_id="1">
    <Position>
        <x>0</x>
        <y>0</y>
        <width>5</width>
        <height>8</height>
    </Position>
    <Title>Visualization #1</Title>
    <Primitives>
        <Axis type="x" linear="true" multi="false">
            <Scale>
                <Accessor>category1</Accessor>
            </Scale>
        </Axis>
        <Axis type="y" linear="true" multi="false">
            <Scale>
                <Accessor>variable1</Accessor>
            </Scale>
        </Axis>
        <Mark type="individual" stacked="false" linear="true">
            <Shape>circle</Shape>
            <Channels>
                <Position_X>
                    <Accessor>category1</Accessor>
                </Position_X>
                <Position_Y>
                    <Accessor>variable1</Accessor>
                </Position_Y>
                <Color>
                    <Accessor>category2</Accessor>
                </Color>
            </Channels>
            <Interactions></Interactions>
            <Style></Style>
        </Mark>
    </Primitives>
</Component>
```

Figure 6. Configuration of a scatter chart using the DSL.

This syntax can be employed for each component involved in the dashboard in order to describe its whole structure and features. Figure 7 shows the definition of a dashboard with three visualizations that display data from a JSON file.

```
<Dashboard>
    <Title>Test Dashboard</Title>
    <Dataset>
        <Path>data.json</Path>
        <Type>JSON</Type>
    </Dataset>
    <Page page_id="1">
        <Component type="visualization" component_id="1">...</Component>
        <Component type="visualization" component_id="2">...</Component>
        <Component type="visualization" component_id="3">...</Component>
    </Page>
</Dashboard>
```

Figure 7. Configuration of a dashboard using the DSL.

The users' characteristics can also be structured following the meta-model. In this case, it could be possible to represent users' goals and tasks regarding their own datasets. Structuring this information is also important to infer which visual components could be more effective depending on the user characteristics and purposes.

Having such structured definition of goals and tasks can support a generative pipeline (Figure 8) in which the user purposes regarding his or her data are analyzed to yield a set of visual components model with their own features defined by the meta-model.

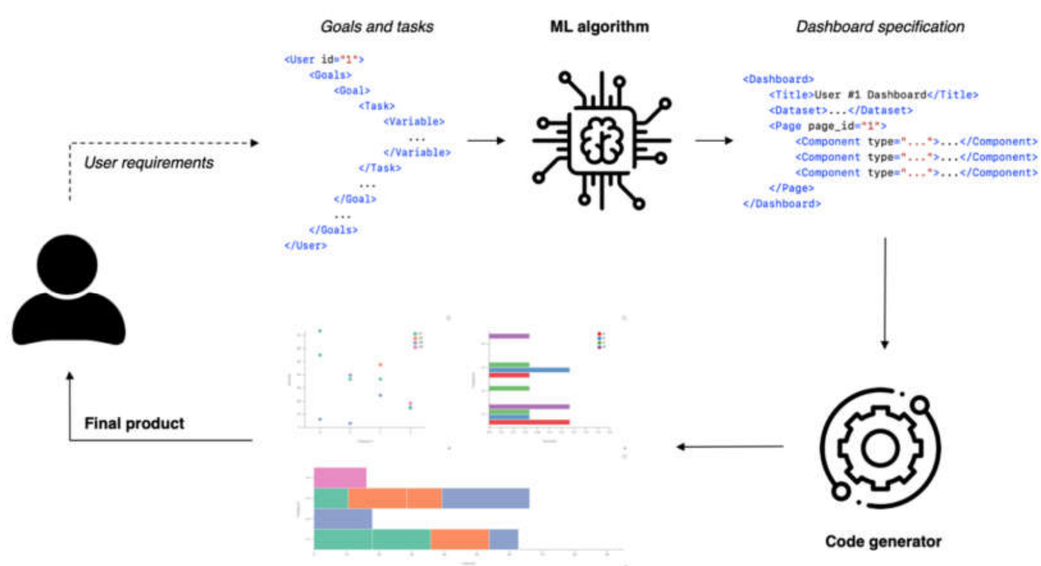


Figure 8. Generative pipeline proposal using the metamodel-based DSL. Icons made by Freepik (www.flaticon.com/authors/freepik).

4.3. Example of Use

A dashboard generator has been implemented taking into account the structure of the DSL. The generator takes as an input the XML configuration files and yields a set of HTML and JavaScript documents holding the logic and features specified through the DSL. The generator logic is based on the software product lines paradigm [54, 55, 58, 59]. The generation process is out of the scope of this paper, but a dashboard example resulting from this process is provided in Figure 9. This dashboard is based on the configuration files previously presented. The visualization on the top-left corner in Figure 9 follows the specification shown in Figure 6.

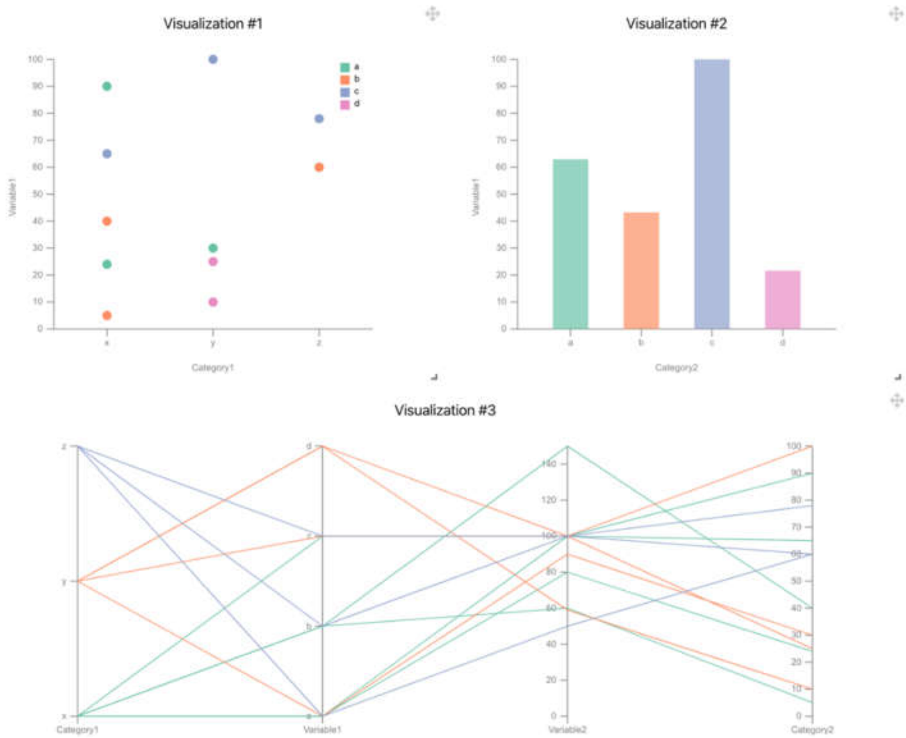


Figure 9. A sample dashboard generated by using the DSL.

This dashboard can be easily modified only by editing the configuration file. Components' channels, scales, styles, etc., allow new specifications to obtain different visual metaphors to convey the same information. For example, Figure 10 shows another example of a dashboard generated using the same configuration file used for the dashboard in Figure 9. The only modification was made on the second (top-right) component, which in this case shows the same information through another visual form after modifying the coordinate system of the axes and visual marks.

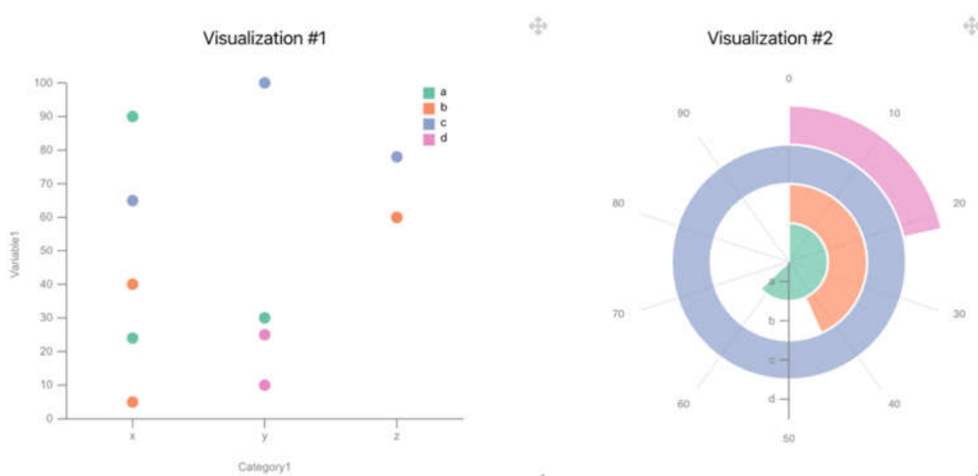


Figure 10. A sample dashboard modified by using the DSL. The second component shows the same information but using a polar coordinate system instead of a Cartesian coordinate system.

5. Discussion

Data-driven approaches can report a lot of benefits, but it is necessary to gain insights and generate knowledge from datasets to make informed decisions. Information dashboards present data in an understandable way, enabling the audience to identify patterns, clusters, outliers, etc. Information dashboards that in the past were mainly employed and reserved for technical and analytical profiles are now spreading their use across any kind of profile. However, it is necessary to understand the audience to design an effective dashboard [10, 11].

In this work, a dashboard meta-model has been presented. The meta-model abstracts the main technical and visual features of information dashboards. The different visual marks, and how they encode variables or operation outputs through channels, as well as other elements such as scales, axes, legends, etc., are part of the meta-model, because these are the primitives that are shared among data visualizations.

In this case, the meta-model primarily focuses on the user. The final user, which will be the entity that would gain insights through the dashboard, can have different characteristics and goals regarding data, and they could change depending on the context. Capturing these traits through the meta-model is essential, as dashboards' features arise from the users' requirements and are influenced by them [10].

The end-user might be seen as an "external" entity that has almost no influence on the dashboard design or technical features, but in the end, these technical features are crucial to deliver a good user experience. This is why user preferences, as well as other characteristics, like user's knowledge level about the data domain, visual literacy, and user's potential biases, are represented in the meta-model and are tightly related to the dashboard elements.

This information provides the most suitable view type by configuring recognizable visual marks or visual metaphors, preferred visual design, etc. Moreover, user disabilities are also considered, such as color blindness, hand tremors, etc., because they refine the dashboards' visual design and/or interaction methods making fonts more visible, choosing right color palettes, mouse sensibility, etc.

Assessing visualization literacy is currently an important research field [60, 61], in order to know beforehand the users' visualization knowledge level and to deliver an understandable (yet effective) set of visualizations for them. Furthermore, the users' knowledge about the data domain should be addressed in the same manner; by providing views with understandable data dimensions and contextual information to mitigate unawareness about the domain [10].

User bias is also an important trait to account for. Users might be influenced by social or cognitive biases that could distort the discovery of knowledge. Bias could lead to valuable information loss [62, 63], that not only could undermine people but also lead wrong decisions by not addressing biases when analyzing data [64].

However, one of the most important aspects is the users' goals or intents regarding the displayed information when they analyze data. These goals provide hints about which visual metaphors or marks are needed because some charts are more effective for some goals (and tasks) than others [35, 65]. A goal framework and a task taxonomy have been included in the dashboard meta-model to provide more detailed (although abstract) information regarding these two elements. Including this information has a lot of benefits, because goals and tasks are usually expressed in natural language. Using these taxonomies and classifications and arranging them into a meta-model allows the structuration of goals and tasks, thus easing their processing. Many taxonomies were available to classify goals and tasks. However, the analysis goal framework [49] was selected for characterizing goals because of the flexibility that it offers for bridging analytical goals to existing task taxonomies and its well-defined inputs and outputs.

On the other hand, for characterizing tasks, the typology developed by Amar et al. [43] was chosen. The main reason is its simplicity and widespread use in data visualization evaluation. This taxonomy is easy to integrate into the meta-model and can be used along with the analysis goal framework to describe different analytical steps with high levels of abstraction.

Because meta-models are prone to evolution, new versions can be developed with other task taxonomies or new entities that better capture the analytic activity of users, but in this case, it is necessary to take into account the impact of this changes on existing artifacts [66]. With this structuration, it was possible to design a DSL to define dashboard requirements with the aim of processing them and generating products with a configuration that supports the different tasks needed for achieving the users' information goals, fostering knowledge discovery.

Another benefit of relying on a DSL is that the requirements of the dashboard are constrained and structured, allowing an easier specification of their functionalities. The DSL makes the dashboard design process more transparent for designers and unburdens them from technical and programming tasks and provides a user-friendlier approach to increase the readability and accessibility of the information held in the meta-model.

The possibility of designing and generating data visualizations automatically is gaining relevance due to the democratization of data. People are continuously exposed to new information, and it is necessary to provide tools to help any user profile, including non-technical profiles, to generate knowledge and gain insights. Tailoring visualizations to specific user profiles not only aims at presenting great quantities of data in a single display, but also at conveying the information, relationships and characteristics "hidden" within raw datasets, taking into account the necessities of the user [10].

Current visualization generation processes focus mainly on the structure of datasets and their variables [33, 34] to develop artificial intelligence algorithms that infer proper visual encodings or visual metaphors. Focusing on datasets is crucial, because their variable types and their domains provide valuable information regarding potential visual encodings. However, there are other important user aspects to account for when dealing with information visualizations, as stated in this paper. Including this information into a generation pipeline could refine these products to increase their effectiveness and efficiency by following a user-centered approach.

That is why it is important to structure tasks and goals into a meta-model and a DSL. Tasks and goals are usually conveyed through natural language, making their processing a complex activity. For example, many machine learning (ML) algorithms need structured data as an input (e.g., random forest, decision tree, linear regression, etc.). If the goal is to make a generative pipeline that yields adapted dashboards based on a ML model, a structured set of goals, tasks and users' characteristics must be provided to allow the algorithms to infer which visual elements could best match the users' needs and the users' context. By using a structured syntax to define the final user, such as the presented DSL, it is easier to train these models to seek for relationships between the users' characteristics and the effectiveness and usability of specific visual metaphors. This generation pipeline can make decision-making processes more accessible and effective for not statistically-trained people or for non-technical profiles, increasing the outcomes and benefits derived from data-driven processes.

The main challenge of using AI to generate information visualizations and dashboards is the retrieval process of all the presented user dimensions to train the models, not only because several factors are involved, but because the information must be precise to map these characteristics into proper dashboard components successfully. Another benefit of the meta-model is that arranging all these requirements into abstract entities can assist the definition of data collection tools by using its structure and relationships to determine which data is necessary and which factors might be related. Materializing these abstract primitives into software components can support the creation of perception questionnaires, such as in [35], with the goal of testing the influence of visualization primitives (visual marks, encodings, scales, etc.) and their relationship with analytic tasks and users' traits.

This approach can be highly valuable in contexts or domains in which different actors with different profiles are involved, like in the educational context. The diversity of roles in the educational context was analyzed in a literature review regarding educational dashboards conducted in [24]. In this literature review it was found that the majority of users are usually teachers, but students, administrators and researchers are also among the main users of these tools. Educational dashboards are also diverse in terms of their objectives; self-monitoring, monitoring of other students and administrative monitoring [24]. In the educational context, dashboards are not only useful to inform tutors about student performance, but can also become tools to motivate students. They can even serve as tools for students to self-regulate and compare their own results [25]. For these reasons, students, teachers, managers can benefit from tailored dashboards to reach more meaningful insights regarding their data interests.

The goals and necessary steps to reach them are crucial to select the visualization primitives that will be present on the dashboard because the visual marks and encodings must support the analytic tasks. For example, not every visual metaphor is useful for identifying correlation [67], so if one of the steps of the analytic goal implies searching for correlation, the visualizations that support the goal must have encodings and visual marks that foster the effectiveness of that task.

This work is focused on the technical aspects of supporting a generative dashboard pipeline by using a high-level definition of these tools through a meta-modeling approach. However, the goal of relying on a meta-model is not only to automatically generate dashboards. Although the usefulness of these artifacts is often limited to their support in model-driven development approaches, the dashboards' domain provides interesting application alternatives.

To develop a meta-model, it is important to shift from low-level and concrete specifications, to high-level and generic specifications. Due to the generic definition of features, meta-models can play other roles when applied to the dashboards domain. In this case, dissecting dashboards and identifying their most defining properties can support domain experts and practitioners in detecting when a dashboard is showing data in a distorted, dubious or inaccurate manner [68] by focusing on visualizations' primitive elements and the features that make graphs potentially misleading.

6. Conclusions

This work presents a dashboard meta-model that accounts not only for the technical and structural features of these tools, but also for the goals and characteristics of their end-users. The main goal of the meta-model is to provide an input for a dashboard generative pipeline, in order to obtain tailored dashboards instantiated from the abstract and high-level characteristics of the meta-model. However, to automatize the generation of dashboards and information visualizations, it is necessary to obtain data regarding the users' goals and characteristics. These data are crucial to generate rules to infer the most suitable dashboard features for each individual situation or context. Structuring dashboards' and users' characteristics into a set of abstract entities could support the definition of data collection tools with the vision of gather information regarding how users with different traits behave when facing dashboards and visualizations.

Several contexts could benefit from the adaptation of information dashboards, especially the educational context, in which data mining and analytics are becoming more widespread given their benefits in supporting decisions regarding learning methodologies [69-75]. Tailored educational dashboards could support knowledge generation through visual analysis, no matter the end user's characteristics, improving and making decision-making processes more accessible.

Future research lines will involve the definition and application of a data collection method in real-world contexts to test which dashboard configurations are more effective depending on the end-user characteristics and goals, and also depending on the dataset domains. With this information, it could be possible to train ML models and to add rules and constraints to the meta-model with the purpose of creating a generation pipeline of tailored dashboards based on reusable software components.

Author Contributions: Conceptualization, A.V.I., F.J.G.P., R.T. and M.A.C.; methodology, A.V.I., F.J.G.P., R.T. and M.A.C.; software, A.V.I.; validation, F.J.G.P., R.T. and M.A.C.; formal analysis, A.V.I., F.J.G.P. and R.T.; investigation, A.V.I., F.J.G.P., R.T. and M.A.C.; resources, A.V.I., F.J.G.P. and R.T.; writing—original draft preparation, A.V.I.; writing—review and editing, F.J.G.P., R.T. and M.A.C.; visualization, A.V.I.; supervision, F.J.G.P., R.T. and M.A.C.; project administration, F.J.G.P. and R.T.; funding acquisition, F.J.G.P. and R.T. All authors have read and agreed to the published version of the manuscript.

Funding: This research was partially funded by the Spanish Government Ministry of Economy and Competitiveness throughout the DEFINES project grant number [TIN2016-80172-R]. This research was supported by the Spanish Ministry of Education, Culture and Sport under a FPU fellowship (FPU17/03276).

Acknowledgments: The authors would like to thank the InterAction and eLearning Research Group (GRIAL) for its support to conduct the present research https://grial.usal.es.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Patil, D.; Mason, H. *Data Driven*; O'Reilly Media, Inc. Sebastopol, CA, USA, 2015.
2. Lu, H.; Zhu, Y.; Shi, K.; Lv, Y.; Shi, P.; Niu, Z. Using adverse weather data in social media to assist with city-level traffic situation awareness and alerting. *Appl. Sci.* **2018**, *8*, 1193.
3. Chang, K.-M.; Dzeng, R.-J.; Wu, Y.-J. An automated IoT visualization BIM platform for decision support in facilities management. *Appl. Sci.* **2018**, *8*, 1086.
4. Cardoso, A.; Vieira Teixeira, C.J.; Sousa Pinto, J. Architecture for Highly Configurable Dashboards for Operations Monitoring and Support. *Stud. Inform. Control* **2018**, *27*, 319–330.
5. Mayer, B.; Weinreich, R. A dashboard for microservice monitoring and management. In Proceedings of the 2017 IEEE International Conference on Software Architecture Workshops (ICSAW), Gothenburg, Sweden, 5–7 April *2017;* pp. 66–69.
6. Michel, C.; Lavoué, E.; George, S.; Ji, M. Supporting awareness and self-regulation in project-based learning through personalized dashboards. *Int. J. Technol. Enhanc. Learn.* **2017**, *9*, 204–226.
7. Aldrich, F.; Sheppard, L. Graphicacy-the fourth'R'? *Prim. Sci. Rev.* **2000**, *64*, 8–11.
8. Balchin, W.G. Graphicacy. *Am. Cartogr.* **1976**, *3*, 33–38.
9. Few, S. *Information Dashboard Design*; O'Reilly Media, Inc. Sebastopol, CA, USA, 2006.
10. Sarikaya, A.; Correll, M.; Bartram, L.; Tory, M.; Fisher, D. What Do We Talk About When We Talk About Dashboards? *IEEE Trans. Vis. Comput. Graph.* **2018**, *25*, 682–692.
11. Berinato, S. *Good Charts*: *The HBR Guide to Making Smarter*, *More Persuasive Data Visualizations*; Harvard Business Review Press: Brighton, MA, USA, 2016.
12. Vázquez-Ingelmo, A.; García-Peñalvo, F.J.; Therón, R. Information Dashboards and Tailoring—A Systematic Literature Review. *IEEE Access* **2019**, *7*, 109673–109688, doi:10.1109/ACCESS.2019.2933472.
13. Kintz, M.; Kochanowski, M.; Koetter, F. Creating User-specific Business Process Monitoring Dashboards with a Model-driven Approach. In Proceedings of the MODELSWARD, 2017, Porto, Portugal, 19-21 February 2017; pp. 353–361.
14. Palpanas, T.; Chowdhary, P.; Mihaila, G.; Pinel, F. Integrated model-driven dashboard development. *Inf. Syst. Front.* **2007**, *9*, 195–208.
15. Pleuss, A.; Wollny, S.; Botterweck, G. Model-driven development and evolution of customized user interfaces. In Proceedings of the 5th ACM SIGCHI Symposium on Engineering Interactive Computing Systems, London, UK, 24–27 June *2013;* pp. 13–22.
16. Logre, I.; Mosser, S.; Collet, P.; Riveill, M. Sensor data visualisation: A composition-based approach to support domain variability. In Proceedings of the European Conference on Modelling Foundations and Applications, York, UK, 21–25 July 2014; pp. 101–116.
17. Vázquez-Ingelmo, A.; García-Peñalvo, F.J.; Therón, R. Tailored information dashboards: A systematic mapping of the literature. In Proceedings of the Interacción 2019, Donostia, Spain, 25–28 June 2019.
18. Vázquez Ingelmo, A.; García-Peñalvo, F.J.; Therón, R.; Conde González, M.Á. Extending a dashboard meta-model to account for users' characteristics and goals for enhancing personalization. In Proceedings of the Learning Analytics Summer Institute (LASI) Spain 2019, Vigo, Spain, 27–28 June 2019.
19. Vázquez-Ingelmo, A.; García-Holgado, A.; García-Peñalvo, F.J.; Therón, R. Dashboard Meta-Model for Knowledge Management in Technological Ecosystem: A Case Study in Healthcare. In Proceedings of the UCAmI 2019, Toledo, Castilla-La Mancha, Spain, 1–15 July 2019.
20. Vázquez-Ingelmo, A.; García-Peñalvo, F.J.; Therón, R. Capturing high-level requirements of information dashboards' components through meta-modeling. In Proceedings of the 7th International Conference on Technological Ecosystems for Enhancing Multiculturality (TEEM 2019), León, Spain, 16–18 October 2019.
21. Yoo, Y.; Lee, H.; Jo, I.-H.; Park, Y. Educational dashboards for smart learning: Review of case studies. In *Emerging Issues in Smart Learning*, Springer. Berlin, Heidelberg 2015; pp. 145–155.
22. Roberts, L.D.; Howell, J.A.; Seaman, K. Give me a customizable dashboard: Personalized learning analytics dashboards in higher education. *Technol. Knowl. Learn.* **2017**, *22*, 317–333.
23. Dabbebi, I.; Iksal, S.; Gilliot, J.-M.; May, M.; Garlatti, S. Towards Adaptive Dashboards for Learning Analytic: An Approach for Conceptual Design and implementation. In Proceedings of the 9th International

Conference on Computer Supported Education (CSEDU 2017), Porto, Portugal, 21–23 April 2017; pp. 120–131.

24. Schwendimann, B.A.; Rodriguez-Triana, M.J.; Vozniuk, A.; Prieto, L.P.; Boroujeni, M.S.; Holzer, A.; Gillet, D.; Dillenbourg, P. Perceiving learning at a glance: A systematic literature review of learning dashboard research. *IEEE Trans. Learn. Technol.* **2017**, *10*, 30–41.
25. Teasley, S.D. Student facing dashboards: One size fits all? *Technol. Knowl. Learn.* **2017**, *22*, 377–384.
26. Kaur, P.; Owonibi, M. A Review on Visualization Recommendation Strategies. In Proceedings of the 12th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP 2017), Porto, Portugal, 27 February–1 March 2017; pp. 266–273.
27. Mackinlay, J.; Hanrahan, P.; Stolte, C. Show me: Automatic presentation for visual analysis. *IEEE Trans. Vis. Comput. Graph.* **2007**, *13*, 1137–1144.
28. Viegas, F.B.; Wattenberg, M.; Van Ham, F.; Kriss, J.; McKeon, M. Manyeyes: A site for visualization at internet scale. *IEEE Trans. Vis. Comput. Graph.* **2007**, *13*, 1121–1128.
29. Wongsuphasawat, K.; Moritz, D.; Anand, A.; Mackinlay, J.; Howe, B.; Heer, J. Voyager: Exploratory analysis via faceted browsing of visualization recommendations. *IEEE Trans. Vis. Comput. Graph.* **2015**, *22*, 649–658.
30. Voigt, M.; Pietschmann, S.; Grammel, L.; Meißner, K. Context-aware recommendation of visualization components. In Proceedings of the Fourth International Conference on Information, Process, and Knowledge Management (eKNOW), Valencia, Spain, 30 January–4 February *2012*; pp. 101–109.
31. Key, A.; Howe, B.; Perry, D.; Aragon, C. Vizdeck: Self-organizing dashboards for visual analytics. In Proceedings of the 2012 ACM SIGMOD International Conference on Management of Data, New York, NY, USA, 20–24 May 2012; pp. 681–684.
32. Mutlu, B.; Veas, E.; Trattner, C. Vizrec: Recommending personalized visualizations. *ACM Trans. Interact. Intell. Syst.* **2016**, *6*, 31.
33. Hu, K.; Bakker, M.A.; Li, S.; Kraska, T.; Hidalgo, C. VizML: A Machine Learning Approach to Visualization Recommendation. In Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems, Glasgow, Scotland, UK, 4–9 May 2019; p. 128.
34. Dibia, V.; Demiralp, Ç. Data2Vis: Automatic generation of data visualizations using sequence to sequence recurrent neural networks. *IEEE Comput. Graph. Appl.* **2019**, *39*, 33–46.
35. Saket, B.; Endert, A.; Demiralp, C. Task-based effectiveness of basic visualizations. *IEEE Trans. Vis. Comput. Graph.* **2018**, *25*, 2505–2512.
36. Vartak, M.; Huang, S.; Siddiqui, T.; Madden, S.; Parameswaran, A. Towards visualization recommendation systems. *ACM Sigmod Rec.* **2017**, *45*, 34–39.
37. Kleppe, A.G.; Warmer, J.; Bast, W. *MDA Explained. The Model Driven Architecture: Practice and Promise*; Addison-Wesley Longman Publishing Co., Inc.: Boston, MA, USA, 2003.
38. Mellor, S.J.; Scott, K.; Uhl, A.; Weise, D. Model-Driven Architecture. In *Advances in Object-Oriented Information Systems*, *Proceedings of the OOIS 2002 Workshops*, *Montpellier*, *France*, *2 September 2002*; Bruel, J.-M., Bellahsene, Z., Eds.; Springer: Berlin/Heidelberg, Germany, 2002; pp. 290–297.
39. Álvarez, J.M.; Evans, A.; Sammut, P. Mapping between Levels in the Metamodel Architecture. In *≪UML≫ 2001—The Unified Modeling Language. Modeling Languages*, *Concepts*, *and Tools. UML 2001. Lecture Notes in Computer Science*; Gogolla, M., Kobryn, C., Eds.; Springer: Berlin/Heidelberg, Germany, 2001; Volume 2185, pp. 34–46.
40. García-Holgado, A.; García-Peñalvo, F.J. Validation of the learning ecosystem metamodel using transformation rules. *Future Gener. Comput. Syst.* **2019**, *91*, 300–310, doi:10.1016/j.future.2018.09.011.
41. Kang, K.C.; Cohen, S.G.; Hess, J.A.; Novak, W.E.; Peterson, A.S. *Feature-Oriented Domain Analysis (FODA) Feasibility Study*; Carnegie-Mellon University, Software Engineering Institute: Pittsburgh, PA, USA, 1990.
42. Voelter, M.; Visser, E. Product line engineering using domain-specific languages. In Proceedings of the 2011 15th International Software Product Line Conference (SPLC), Munich, Germany, 22–26 August *2011*; pp. 70–79.
43. Amar, R.; Eagan, J.; Stasko, J. Low-level components of analytic activity in information visualization. In Proceedings of the IEEE Symposium on Information Visualization, Los Alamitos, CA, USA, 23–25 October 2005; pp. 111–117.
44. Schulz, H.-J.; Nocke, T.; Heitzler, M.; Schumann, H. A design space of visualization tasks. *IEEE Trans. Vis. Comput. Graph.* **2013**, *19*, 2366–2375.

45. Gotz, D.; Zhou, M.X. Characterizing users' visual analytic activity for insight provenance. *Inf. Vis.* **2009**, *8*, 42–55.
46. Dimara, E.; Franconeri, S.; Plaisant, C.; Bezerianos, A.; Dragicevic, P. A task-based taxonomy of cognitive biases for information visualization. *IEEE Trans. Vis. Comput. Graph.* **2018**, *26*, 1413–1432.
47. Munzner, T. *Visualization Analysis and Design*; AK Peters/CRC Press: Boca Raton, FL, USA, 2014.
48. Brehmer, M.; Munzner, T. A multi-level typology of abstract visualization tasks. *IEEE Trans. Vis. Comput. Graph.* **2013**, *19*, 2376–2385.
49. Lam, H.; Tory, M.; Munzner, T. Bridging from goals to tasks with design study analysis reports. *IEEE Trans. Vis. Comput. Graph.* **2017**, *24*, 435–445.
50. Munzner, T. A nested process model for visualization design and validation. *IEEE Trans. Vis. Comput. Graph.* **2009**, *15*, 921–928.
51. Bray, T.; Paoli, J.; Sperberg-McQueen, C.M.; Maler, E.; Yergeau, F. Extensible markup language (XML). *World Wide Web J.* **1997**, *2*, 27–66.
52. Novák, M. Easy implementation of domain specific language using xml. In Proceedings of the 10th Scientific Conference of Young Researchers (SCYR 2010), Košice, Slovakia, 19 May, 2010.
53. Fallside, D.C.; Walmsley, P. *XML Schema Part 0*: *Primer Second Version*; W3C: Cambridge, *MA*, USA, 2004. Available online: https://www.w3.org/TR/xmlschema-0/ (accessed on 27/03/2020).
54. Clements, P.; Northrop, L. *Software Product Lines*; Addison-Wesley: Boston, MA, USA, 2002.
55. Vázquez-Ingelmo, A.; García-Peñalvo, F.J.; Therón, R. Addressing Fine-Grained Variability in User-Centered Software Product Lines: A Case Study on Dashboards. In Proceedings of the World Conference on Information Systems and Technologies, La Toja Island, Galicia, Spain, 16–19 April 2019; pp. 855–864.
56. Ronacher, A. *Jinja2 Documentation*; Jinja2. Available online: https://jinja.palletsprojects.com/en/2.11.x/ (accessed on 27/03/2020).
57. Vázquez-Ingelmo, A. Ecore Version of the Metamodel for Information Dashboards (v2). Available online: https://doi.org/10.5281/zenodo.3561320 (accessed on 27/03/2020).
58. Gomaa, H. *Designing Software Product Lines with UML*: *From Use Cases to Pattern-Based Software Architectures*; Addison Wesley Longman Publishing Co., Inc.: Boston, MA, USA, 2004.
59. Kästner, C.; Apel, S.; Kuhlemann, M. Granularity in software product lines. In Proceedings of the 30th International Conference on Software Engineering, Leipzig, Germany, 10–18, May 2018; pp. 311–320.
60. Lee, S.; Kim, S.-H.; Kwon, B.C. Vlat: Development of a visualization literacy assessment test. *IEEE Trans. Vis. Comput. Graph.* **2017**, *23*, 551–560.
61. Boy, J.; Rensink, R.A.; Bertini, E.; Fekete, J.-D. A principled way of assessing visualization literacy. *IEEE Trans. Vis. Comput. Graph.* **2014**, *20*, 1963–1972.
62. Hullman, J.; Adar, E.; Shah, P. The impact of social information on visual judgments. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, Vancouver, BC, Canada, 7-12 May, 2011; pp. 1461–1470.
63. Kim, Y.-S.; Reinecke, K.; Hullman, J. Data through others' eyes: The impact of visualizing others' expectations on visualization interpretation. *IEEE Trans. Vis. Comput. Graph.* **2018**, *24*, 760–769.
64. Perez, C.C. *Invisible Women: Exposing Data Bias in a World Designed for Men*; Random House: New York, NY, USA, 2019.
65. Sarikaya, A.; Gleicher, M. Scatterplots: Tasks, data, and designs. *IEEE Trans. Vis. Comput. Graph.* **2017**, *24*, 402–412.
66. Iovino, L.; Pierantonio, A.; Malavolta, I. On the Impact Significance of Metamodel Evolution in MDE. *J. Object Technol.* **2012**, *11*, 1–33.
67. Harrison, L.; Yang, F.; Franconeri, S.; Chang, R. Ranking visualizations of correlation using weber's law. *IEEE Trans. Vis. Comput. Graph.* **2014**, *20*, 1943–1952.
68. Cairo, A. *How Charts Lie*: *Getting Smarter about Visual Information*; WW Norton & Company: New York, NY, USA, 2019.
69. Agudo-Peregrina, Á.F.; Iglesias-Pradas, S.; Conde-González, M.Á.; Hernández-García, Á. Can we predict success from log data in VLEs? Classification of interactions for learning analytics and their relation with performance in VLE-supported F2F and online learning. *Comput. Hum. Behav.* **2014**, *31*, 542–550, doi:10.1016/j.chb.2013.05.031.
70. Baepler, P.; Murdoch, C.J. Academic analytics and data mining in higher education. *Int. J. Scholarsh. Teach. Learn.* **2010**, *4*, 17.

71. Ferguson, R. Learning analytics: Drivers, developments and challenges. *Int. J. Technol. Enhanc. Learn.* **2012**, *4*, 304–317, doi:10.1504/IJTEL.2012.051816.
72. Jivet, I.; Scheffel, M.; Drachsler, H.; Specht, M. Awareness is not enough: Pitfalls of learning analytics dashboards in the educational practice. In *Proceedings of the 12th European Conference* on *Technology Enhanced Learning* (*EC-TEL 2017*); Springer. Tallinn, Estonia, 12-15 September 2017; pp. 82–96.
73. Kim, J.; Jo, I.-H.; Park, Y. Effects of learning analytics dashboard: Analyzing the relations among dashboard utilization, satisfaction, and learning achievement. *Asia Pac. Educ. Rev.* **2016**, *17*, 13–24.
74. Liñán, L.C.; Pérez, Á.A.J. Educational Data Mining and Learning Analytics: Differences, similarities, and time evolution. *Int. J. Educ. Technol. High. Educ.* **2015**, *12*, 98–112.
75. Sein-Echaluce, M.L.; Fidalgo-Blanco, Á.; Esteban-Escaño, J.; García-Peñalvo, F.J.; Conde-González, M.Á. Using learning analytics to detect authentic leadership characteristics at engineering degrees. *Int. J. Eng. Educ.* **2018**, in press.

7.16 Appendix P. A Meta-Model Integration for Supporting Knowledge Discovery in Specific Domains: A Case Study in Healthcare

sensors

MDPI

Article

A Meta-Model Integration for Supporting Knowledge Discovery in Specific Domains: A Case Study in Healthcare

Andrea Vázquez-Ingelmo [1,*], Alicia García-Holgado [1], Francisco José García-Peñalvo [1] and Roberto Therón [2]

1 GRIAL Research Group, Computer Science Department, University of Salamanca, 37008 Salamanca, Spain; aliciagh@usal.es (A.G.-H.); fgarcia@usal.es (F.J.G.-P.)

2 VisUSAL, GRIAL Research Group, Computer Science Department, University of Salamanca, 37008 Salamanca, Spain; theron@usal.es

* Correspondence: andreavazquez@usal.es; Tel.: +34-923294500 (ext. 3433)

Received: 30 June 2020; Accepted: 20 July 2020; Published: 22 July 2020

Abstract: Knowledge management is one of the key priorities of many organizations. They face different challenges in the implementation of knowledge management processes, including the transformation of tacit knowledge—experience, skills, insights, intuition, judgment and know-how—into explicit knowledge. Furthermore, the increasing number of information sources and services in some domains, such as healthcare, increase the amount of information available. Therefore, there is a need to transform that information in knowledge. In this context, learning ecosystems emerge as solutions to support knowledge management in a different context. On the other hand, the dashboards enable the generation of knowledge through the exploitation of the data provided from different sources. The model-driven development of these solutions is possible through two meta-models developed in previous works. Even though those meta-models solve several problems, the learning ecosystem meta-model has a lack of decision-making support. In this context, this work provides two main contributions to face this issue. First, the definition of a holistic meta-model to support decision-making processes in ecosystems focused on knowledge management, also called learning ecosystems. The second contribution of this work is an instantiation of the presented holistic meta-model in the healthcare domain.

Keywords: model-driven development; dashboard; meta-model; knowledge management; healthcare; technological ecosystem; health ecosystem; meta-model integration

1. Introduction

At present, society is changing. Three main concepts are used to refer to these changes, information society, knowledge society, and digital society. These concepts are strictly related, and sometimes authors use them as synonyms. Each concept emphasizes an aspect of society. In particular, the digital society concept is used to put a focus on how digital technologies have an digitalization impact on today's society, culture and politics. According to Dufva and Dufva [1], digital technologies are entangled in the structures of society in many different, complex and even contradictory ways. In this context, the concept of information society is used above all when dealing with technological aspects and their effects on economic growth and employment [2].

On the other hand, the concept of the knowledge society emphases that knowledge management is available for most people due to the unlimited access to information. The core element is not the technology but the ability to identify, produce, process, transform, disseminate and use the information to build and apply knowledge for human development [3]. Knowledge management requires training, apprenticeships and other more costly forms of transmission, while information can generally be reproduced at minimal cost [4].

According to Castells [5], what characterizes the current technological revolution is not the centrality of knowledge and information, but the application of such knowledge and information to knowledge generation and information processing/communication devices. Knowledge has become a crucial element for development, given its power as a strategic factor for building new policies, planning new actions and fostering innovation within organizations. Knowledge management is considered a sustainable competitive advantage [6], so organizations expend part of their resources on building their capacity to transform and transfer new knowledge continuously over time [7].

However, knowledge is not only present physically (i.e., in documents or books), it is also present in employees and the different processes carried out at organizations. Eraut [8] defines this type of knowledge as the cognitive resource which a person brings to a situation that enables them to think and perform. This knowledge may be either explicit or tacit, and it is the tacit form which is more difficult to codify and reproduce. This tacit knowledge includes experience, skills, insights, intuition, judgment and know-how. According to [6], knowledge management processes must be able to support the transfer of tacit knowledge to explicit knowledge and vice versa. This scattered nature of knowledge makes its management a complex and crucial task. Additionally, the increasing number of information sources and services, such as intelligent devices, sensors embedded in the environment and the Internet-of-Things (IoT), increase the amount of information available, and therefore there is a need to transform that information into knowledge.

Although knowledge management is one of the key priorities of institutions and companies, according to Davenport et al. [9], the implementation of knowledge management processes within an organization could be expensive, meaning that not all organizations have the capability and the resources to profit from their knowledge. Also, the different domains present different challenges. For example, in the healthcare domain, there is a trend in developing sensor-based applications for monitoring well-being or health conditions and to trigger alarms sent to relatives or care centers [10,11]. In particular, the development of ambient assisted living platforms and systems has increased in the last decade due to the alarming numbers of growing elderly population [11].

Over recent years, software ecosystems have emerged as a technological solution to support information and knowledge management in different contexts. Manikas and Hansen [12] define the software ecosystem as the interaction of a set of actors on a common technological platform that results in a series of software solutions and services. Other authors [13–15] refer to ecosystems that have a central software system or platform that provides a set of core functionality and offers the tools for users or developers to contribute services and software and hardware components to extend the functionality of the ecosystem. Institutions adopt a software ecosystem strategy to expand their organizational boundaries, share their platforms and resources with third parties, and define new business models [12,16].

There are many terms to describe this type of solution, though each one has distinctive characteristics [17]. In particular, the technological ecosystem concept emphasizes the human factor as an inherent part of the ecosystem in order to develop solutions with evolutive capabilities focused on supporting knowledge management in heterogeneous contexts. Technological ecosystems provide a general framework that allows defining and developing any type of technological solution, describing how data and information are shared between the ecosystem actors and how those actors interact with each other [18]. One of the main strengths of technological ecosystems is that when their components collaborate, they exploit all of their benefits, obtaining the most out of their functionalities to provide elaborate services. However, there are some challenges associated with the definition and development of these solutions. In particular, challenges associated with decision-making processes related to knowledge management, as well as the changes that continuously occur in any organization.

Furthermore, different terms are used to name technological ecosystems, depending on the domain they are focused. For example, if knowledge management is directed on supporting learning processes, the term learning ecosystem is used. A learning ecosystem is not only a technology, it is a community, with educational methods, policies, regulations, applications and work teams, which can coexist so that their processes are interrelated, and their application is based on the physical factors of the technological environment [19].

In the health sector, technological ecosystems in care and assistance share common characteristics in terms of providing technological means to a community of users (patients, relatives, caregivers, doctors, etc.) in order to involve all of them in providing better care and assistance-related services [20]. For this domain, information dashboards can be very powerful to exploit knowledge within healthcare technological ecosystems. Information dashboards allow through visual analysis [21,22] the identification of patterns, outliers, relationships, etc., within data, enabling the generation of knowledge.

However, dashboards need to be adapted to their audience [23], to specific data domains, and to the tasks that will be performed to analyze these data, among other factors. This is necessary because users have different mental models [24], goals, experience, literacy, domain knowledge, and so on. [25–34], making the design process of a dashboard a complex task where the elicitation of requirements can be seen as the backbone process, as it will drive the subsequent phases and decisions regarding the configuration and design of the tool.

A meta-modelling approach can be employed to ease this process. It is possible to obtain a general structure of dashboards that can be instantiated and adapted to any kind of contexts, data domains or audiences by abstracting the common and primitive elements of information dashboards. But not only dashboards' technical elements need to be taken into account during the meta-modelling process. As will be detailed, the inclusion of dashboard users and their requirements as elements of the meta-model enables the integration of this meta-model as a part of technological ecosystems, specifically, learning ecosystems, providing support to discover knowledge and support decision making processes [35].

This work provides two main contributions. First, the definition of a holistic meta-model to support decision-making processes in ecosystems focused on knowledge management, also called learning ecosystems. This meta-model integrates two meta-models defined in previous works: a learning ecosystem meta-model to support the definition of learning ecosystems based on open source software [36]; and a dashboard meta-model to support the analysis of information in order to transform implicit knowledge into tacit knowledge.

The second contribution of this work is an instantiation of the presented holistic meta-model in the healthcare domain. More specifically, an instantiation of a dashboard to support caregivers in order to be included in the eHealth technological ecosystem has been carried out [37]. This case study aims to present the holistic meta-model as a flexible solution able adapt to different actors and contexts. Knowledge management processes not only directly or indirectly affect patients and their relatives, but also healthcare professionals. In particular, there is a need to improve knowledge management processes related to dependent persons due to the aging of the population, with special emphasis in developed countries. The number of persons over 60 years is growing faster than all younger age groups [38], and these numbers have an impact on the cost of care and the resources needed for this population.

The rest of this paper is organized as follows. Section 2 presents related works. Section 3 outlines the methodology followed to develop the meta-models. Section 4 describes the learning ecosystem meta-model, the dashboard meta-model and their integration. Section 5 outlines the architecture of the technological ecosystem in which the dashboard will be included. Section 6 presents the dashboard instantiation. Finally, Sections 7 and 8 discuss the results and depict the conclusions derived from this work.

2. Related Work

According to a systematic review [20], proposed or developed health ecosystems tend to have problems when it comes to being deployed in the real world. Nevertheless, countries and different organizations have dedicated strong investment to the development of health ecosystems in the last decade, looking for new innovative solutions that could alleviate the increasing economic requirements of the health sector [39]. According to other studies about the trends in the development of technological ecosystems focused on health, it can be observed that web-based ecosystems are the most frequently developed, but sensors are present in most of the eHealth ecosystems analyzed [39,40]. Even though there are solutions for different domains inside the health sector, there is a lack of flexible solutions able to adapt to different actors and contexts. Furthermore, the trend in developing platforms and systems that integrate different sensors presents an additional challenge. As data is constantly streamed, the classic information pull (gather, analyze, decide) has to be complemented by a real-time business process [41].

In this context, dashboards are one of the most useful tools for generating knowledge about different data domains. Also, dashboards are popular solutions to exploit the data provided by sensors embedded in the environment, intelligent devices and the IoT [42,43].

The automatic generation of information displays, whether dashboards or single-visualizations, has become a research area of interest given the benefits that these generative pipelines could yield. In fact, different methods have been researched to achieve efficient development processes of these powerful tools, from configuration wizards to software engineering paradigms, among other methodologies [44].

In fact, different applications of the model-driven development to the dashboards' domain can be found in the literature. In [42], a meta-model is employed to visualize sensor data through a composition-based approach. On the other hand, in [45], two meta-models are presented to tackle the automatic generation of visualization and assist non-expert users during the selection process. One of the meta-models describes the user requirements through goals and tasks, and the other meta-model describes the structure of the potential data visualizations. Finally, in [46], a model-driven approach is employed to translate the users' requirements into a set of suitable visualizations based on a skyline-based technique and design guidelines.

As introduced in the first section, the holistic meta-model presented in this work merges two previously developed meta-models: a learning ecosystem meta-model [36] and a dashboard meta-model [47,48]. This dashboard meta-model has a finer grain than the aforementioned meta-models present in the literature, allowing more sophisticated combinations to obtain a wide variety of dashboard displays.

3. Materials and Methods

A model-driven development approach was employed to build the meta-models. Model-driven development (MDD) [49,50] allows separating the data and the operations specification of the system from lower-level details, like the technical aspects related to a specific program language or platforms. This methodology increases the reuse of components (thus decreasing the development time), but also the reuse of knowledge because the structures and relationships identified during the development of the meta-model can evolve to obtain better solutions.

The Object Management Group (OMG) proposal uses model-driven architecture (MDA) as a guideline to implement the MDD approach. This architecture provides a framework for software development in which the process is driven by models that describe and define the target system [51]. The main difference between MDD and MDA is that MDA determines a set of standards to develop the approach, such as meta-object facility (MOF), unified modeling language (UML), Extensible Markup Language (XML) metadata interchange (XMI) and query/view/transformation (QVT).

The dashboard meta-model is also part of the four-layer meta-model architecture proposed by the OMG, in which a model in one layer is used to specify models in the layer below [52]. In particular, the first version of the dashboard meta-model [47,48] was an instance of MOF (i.e., an M2-model), so it can be instantiated to obtain M1-models. This meta-model was transformed in an instance of

Ecore [36] using Graphical Modelling for Ecore included in the Eclipse Modeling Framework (EMF), in order to leverage the different features of this modeling framework (Figure 1).

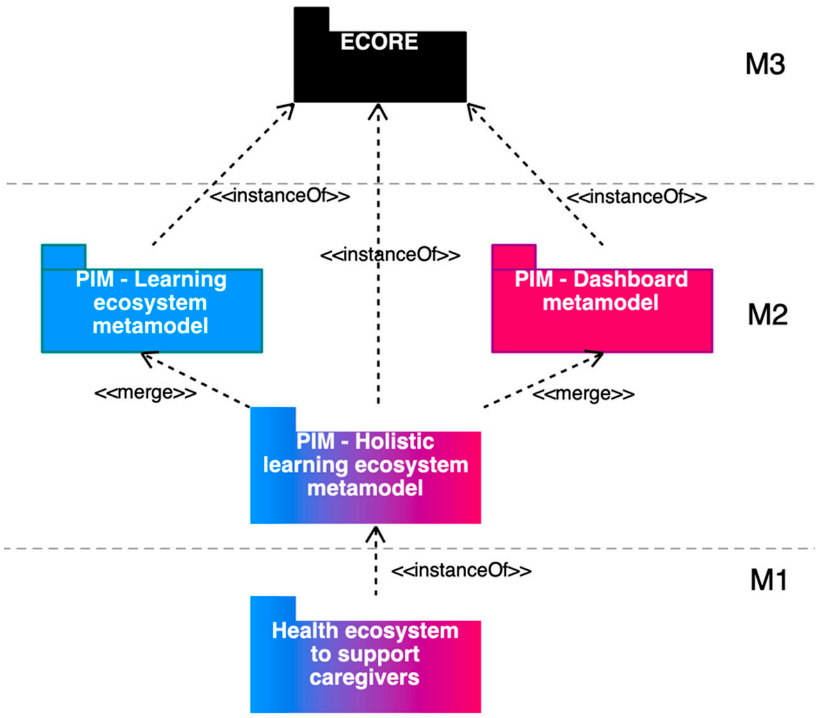


Figure 1. Methodology employed to integrate the learning ecosystem meta-model and the dashboard meta-model organized in the four-layer meta-model architecture of the Object Management Group (OMG).

In this case, the dashboard is a part of the learning ecosystem, which is based on a meta-model defined and validated in previous works. The first version of the learning ecosystem meta-model is based on MOF, and the last validated version is an instance of Ecore [36]. This meta-model was developed using a domain engineering approach [53,54], in which similarities and variability points were identified to obtain an abstract picture of the dashboards' domain in terms of these tools' elements and features.

The dashboard meta-model is also part of the four-layer meta-model architecture proposed by the OMG, in which a model in one layer is used to specify models in the layer below [52]. In particular, the dashboard meta-model is an instance of MOF (i.e., an M2-model), so it can be instantiated to obtain M1-models.

The integration of both meta-models is possible because of the fact that both are platform independent models (PIM) in the M2 layer, although one is instantiated from Ecore (learning ecosystem meta-model) and other from MOF (dashboard meta-model). To get the holistic meta-model, the dashboard meta-model was transformed in an instance of Ecore using Graphical Modelling for Ecore included in EMF. Likewise, the final meta-model can be instantiated to obtain a model in the M1 layer.

4. Holistic Meta-Model: An Integration of Meta-Models

This section presents the two meta-models that were integrated, the learning ecosystem meta-model and the dashboard meta-model.

4.1. Learning Ecosystem Meta-Model

The ecosystem meta-model provides a high-level view of the characteristics that a learning ecosystem has to fulfil to ensure the flexibility of the ecosystem and cover different needs detected in previous studies with real technological ecosystems. In particular, the meta-model definition is based on six real ecosystems for supporting learning processes and knowledge management in different domains (higher education, research, informal learning, youth participation, etc.). Furthermore, the meta-model was used to define real learning ecosystems for public administration, the health sector and knowledge management in PhD programmes.

The objective of the meta-model is to provide a set of guidelines to define learning ecosystems composed of three main elements: software components; a set of components that represent the human factor; and the information flows between the components. The components are black boxes; the learning ecosystem meta-model does not enable capture of the description of a specific component [55]. The meta-model includes Object Constraint Language (OCL) rules which ensure the correct instantiation of a model of the learning ecosystem. Moreover, the guided process is completed with another meta-model, a platform-specific meta-model that provides a pre-selection of open source tools to implement the learning ecosystem.

The learning ecosystem introduces the human factor as something tangible. The human factor is represented through users who have an influence in the other parts of the meta-model. In particular, the human factor is the management processes that define a set of objectives that apply an established methodology (Figure 2).

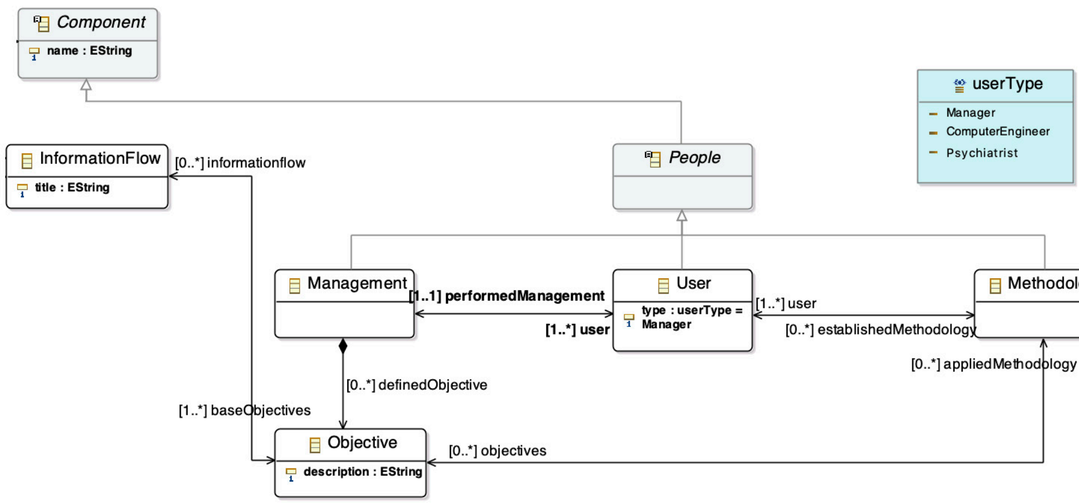


Figure 2. Human factor in the learning ecosystem meta-model. Based on [36].

The objectives are used to define the interaction between the different components of the ecosystem. This interaction is implemented through information flows (Figure 3). The ecosystem meta-model takes into account different ways to provide communication between two software tools. For this reason, each information flow has associated communication mechanisms. Moreover, the mechanisms are modelled as a hierarchy to support the evolution of the meta-model, so it is possible to extend it by adding new communication mechanisms.

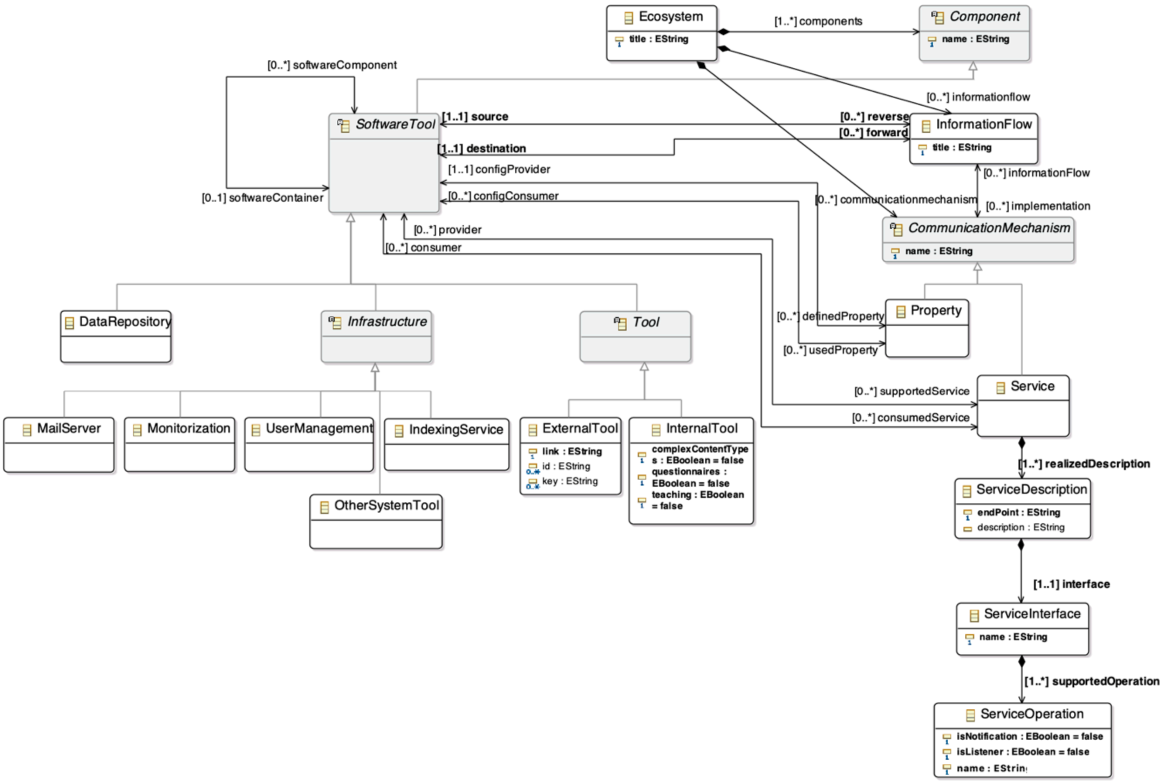


Figure 3. Software components and information flows in the learning ecosystem meta-model. Based on [36]. A high resolution version can be found at https://doi.org/10.5281/zenodo.1066369.

Finally, the last component of the meta-model is the software tools. They are modelled as a hierarchy organized in the layers of the architectural pattern for learning ecosystems, which was defined as a first step to build the meta-model. This architecture has four layers from bottom to top: infrastructure, static data, services, and presentation. Also, the hierarchy allows the addition of new software types.

4.2. Dashboard Meta-Model

The dashboard meta-model represents the high-level and abstracted pieces that compose different types of dashboards and information visualizations. The dashboard meta-model has three main factors that are present in any kind of variant of these tools: the user, the layout, and the components (referring to the content of the dashboard).

Figure 4 shows an excerpt of the dashboard meta-model containing the three sections mentioned. A detailed view of the components section can be consulted in Figure 5.

While the user section describes who will use the dashboard in terms of the user intent, goals, and profile, the layout and the components define more technical aspects of dashboards.

The meta-model supports the definition of different design questions that are common to any information visualization design process, e.g., how many visualizations will the dashboard hold? How these views will be arranged? What type of visualizations will the dashboard display? What type of interaction patterns will the dashboard support? Will the different views be linked? These are generic questions that will have specific answers depending on the data domain and the target audience of the dashboard.

Questions like the aforementioned condition crucial design decisions [56], and crucial design decisions need to be driven by the final consumers of dashboards, the users. That is why including the user in this meta-model is essential, because they will be using dashboards to reach insights, to support their decision-making processes or to exploit datasets.

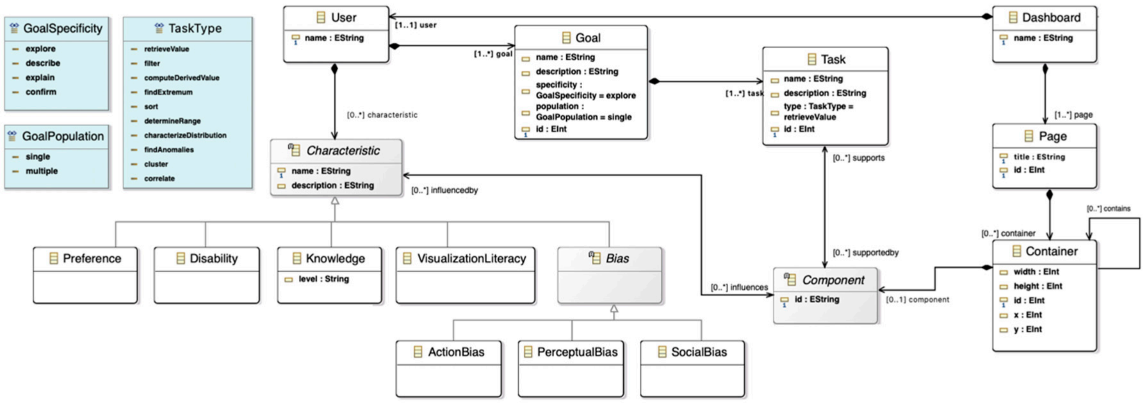


Figure 4. User and layout section of the dashboard meta-model.

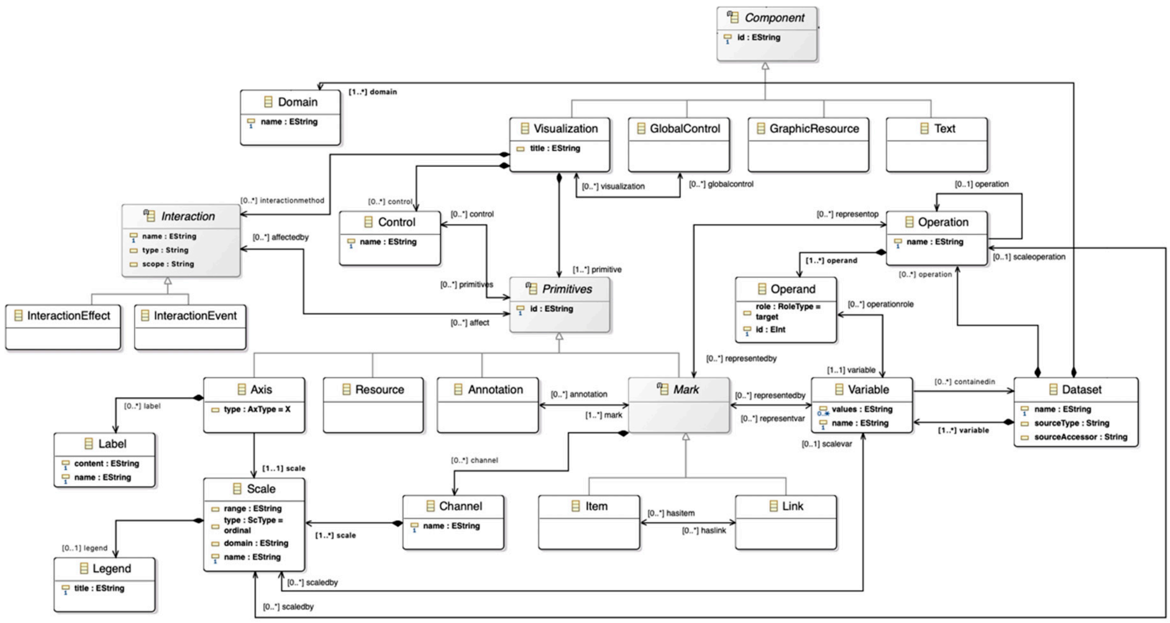


Figure 5. Components' section of the dashboard meta-model. A high resolution version can be found at https://doi.org/10.5281/zenodo.3625703.

The User class is defined in terms of a series of aspects that can be significant and influential within the design process of a dashboard [57]. Given that, the user entity is decomposed in terms of his or her goals and his or her characteristics.

Firstly, a crucial concept arises; goal. A user must have at least one goal for using a dashboard. Goals, in turn, can be broken down into individual and more specific, low-level tasks. Simple goals can be accomplished by performing a few tasks. However, more elaborated goals might involve several specific and chained tasks, which could involve different data dimensions to support complex decision-making processes [58–60].

Users can also have a set of identified characteristics. These characteristics can be diverse, but they need to be well-defined to take them into account while instantiating the meta-models. For example, preferences, disabilities, knowledge about different domains, visualization literacy, and bias are different kinds of characteristics. These characteristics can influence the design process; for example, a user with low visualization literacy might not perceive data correctly through complex visual encodings, so the visualization elements must be adapted properly to account for this user characteristic.

In terms of technical components of the dashboard, several elements are identified. The main components of dashboards are the information visualizations that display data, but also the interaction patterns, controls, graphic resources, or text that complement these visualizations (Figure 3). Information visualizations, in turn, are composed of primitives that encode data variables through channels (i.e., color, size, position, area, etc.). These primitives are the core of information visualizations, because they are the elements that encode the data to be displayed [61,62].

Regarding the data to analyze and visualize, the dataset concept represents the data input in the meta-model. This small piece inside the dashboard meta-model provides a high-level of adaptation. The dataset abstracts the dashboard from the data source. In this sense, the dashboard can integrate different sources such as the interaction generated inside an ecosystem, the information gathered through different sensors or a database provided by the users.

Figure 6 shows a practical example of the identification of different parts of information visualizations through the dashboard meta-model classes.

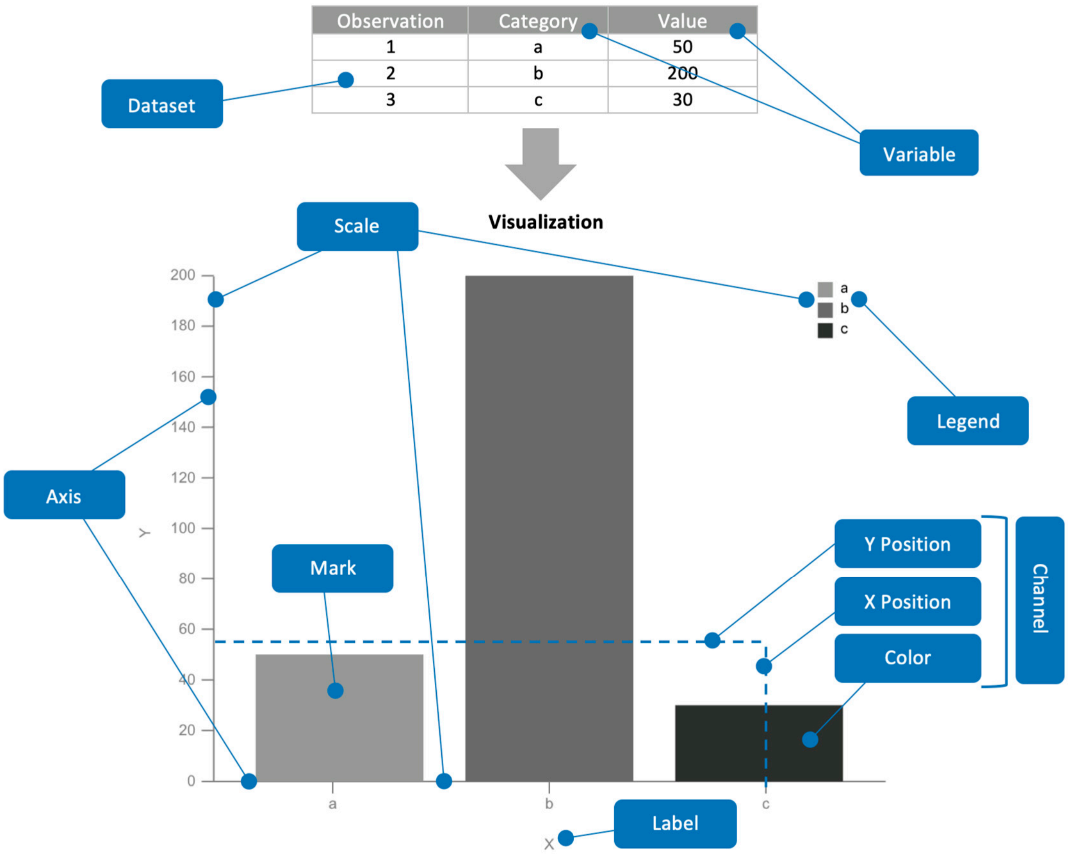


Figure 6. Example of the different elements that compose a visualization (in this case, a bar chart).

In this example, there are two scales that represent two variables (the "Category" variable through an ordinal scale, and the "Value" variable through a linear scale). The domain of these scales is the set of values from the variables. For instance, the domain of the scale that encodes the X position of the visual marks is the set of values retrieved from the "Category" variable (i.e., 'a', 'b', and 'c'). Axes, on the other hand, support the visualization of the scales' domains.

The visual marks of this visualization are bars with a specific position along both X and Y axes and a specific color based on a color scale that encodes the "Category" variable. Scales map the data values to another specific range of values in order to encode the information, that is why these entities are related both to the dataset variables' values (to obtain the domain) and to the visual channels or encodings (to encode these values using another specific range of values, like color codes or screen positions).

4.3. Meta-Model Integration

The previously presented meta-models have been combined to create a holistic meta-model. The idea behind this holistic meta-model is to leverage the connection and collaboration of the elements from both meta-models. The learning ecosystem meta-model has a lack of decision-making support. While monitorization tools, dashboard and other decision-making tools could be instantiated from *Tools* or *Infrastructure*, the main characteristics of the learning ecosystems would not be taken into account. The present proposal aims to solve this problem.

The dashboard meta-model provides the framework to define an infinite number of solutions based on data analysis and visualization. The main aim of a dashboard is to support the decision-making process, so the integration of the dashboard meta-model with the learning ecosystem will support the definition of learning ecosystems for knowledge management from a holistic point of view.

The integration is possible because booth meta-models are M2-level in the four-layer meta-model architecture of the OMG (Figure 1). Despite this, the granularity of each meta-model is different. In particular, the learning ecosystem meta-model has a high-level of abstraction, in which each software component is represented as a black box. The dashboard would be a black box in the ecosystem meta-model before the integration. On the other hand, the dashboard meta-model defines all the elements inside the dashboard component, the level of abstraction minor compared to the ecosystem meta-model.

The integration of the two M2-level metamodels is based on the connection between some elements present in both meta-models. It is important to highlight that the definition of both solutions was conducted independently in different periods of time, so the integration process implied a deep analysis of both solutions to find the connection elements that represent the same concept.

Despite the abstraction differences, the human factor plays a crucial role in both meta-models. The learning ecosystem meta-model introduces users and other elements, such as *Management* and *Methodology*, to represent the human factor at the same abstraction level as other elements in the ecosystem. Regarding the dashboard meta-model, it is necessary to include users in the proposal because they are the drivers and consumers of the displayed data. Moreover, the characteristics of the users, such as *Preference*, *Disability*, *Knowledge*, or *Bias* are used as an input to instantiate the dashboard.

On the other hand, there are two crucial elements shared in both meta-models too. The dashboard *Goals* (within the dashboard meta-model) are represented as *Objectives* within the learning ecosystem meta-model. These elements are represented by a set of *Tasks* and *Information Flows*, respectively. The relevance of these entities is that they are the core of the meta-model, because they frame the required components to achieve the goals or objectives set.

Figure 7 shows the connection between both meta-models. The dashboard *Goal* is merged with *Objective*. The connection between *Goal* and *User* in the dashboard meta-model is replaced by the association between *User* and *Objective* through *Management*. In this sense, the management decisions in the ecosystem define all the goals that support the definition of the dashboard.

Regarding the integration of *Dashboard*, the main class of the dashboard meta-model which contains all the elements in the meta-model, it is connected with the learning ecosystem meta-model through a subclass of *Tool*. Besides, the connection between *User* and *Dashboard*, which has a strong impact on the dashboard meta-model, is included in the proposal. The information flows and tasks are different concepts, so it is not possible to merge them. For this reason, the *Task* entity is included in Figure 7.

On the other hand, a new communication mechanism is included to implement the information flows, the *Dataset*, as a way to represent the integration between the dashboard and other software tools in the learning ecosystem. Also, the dashboard *Component* has been renamed as *Dashboard Component* to distinguish it from the learning ecosystem *Component*.

Finally, the connection between dashboard *Characteristic* and *User* appears in the new proposal. Tasks are supported by the dashboard's elements, that are also influenced by the user characteristics to match his or her information requirements.

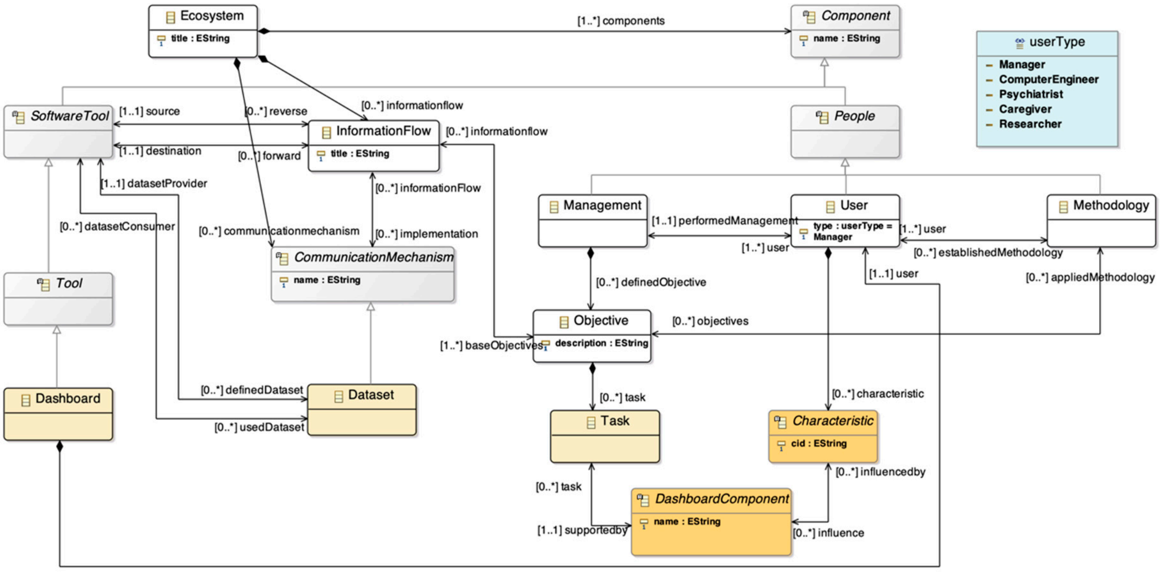


Figure 7. Holistic meta-model: the integration of the learning ecosystem meta-model with the dashboard meta-model.

5. Healthcare Ecosystem for Caregivers

The aim of the technological ecosystem for caregivers is to support the learning and knowledge management processes to develop and enhance the caregiving competences both at home and in the care environments of formal and informal caregivers [37]. In particular, the ecosystem allows psychoeducation [63] to be provided to dependent persons and informal caregivers in order to alleviate the physical and mental health problems that they suffer, such as work overload, depression or anxiety.

The ecosystem makes it possible to provide remote access to different services. It is composed of a set of software components (Figure 8) and based on a set of management and methodological input streams—a business plan, a training plan, and a medical protocol. First, it provides remote teaching-learning environments to support both informal and formal caregivers. Through *Discover*, psychoeducation is accessible to these profiles, so they can obtain psychological support and answers to the questions that arise daily during their care duties, as well as information, advice, and guidance.

Second, *SocialNet* is an online tool that provides a private social network composed of a set of private and safe areas, called walls, for each patient [64]. The main users are the relatives of the patients and their caregivers. In some cases, patients can also access to *SocialNet* to publish their activities and view the contents published by their caregivers and relatives. Finally, the caregivers' managers can access to the social network, but only to manage which caregivers control a patient's wall (this relationship is not represented in Figure 8 to avoid lines crossing the whole system).

Third, the *Dashboard* is a tool to support decision-making processes. In particular, it is focused on supporting caregivers' managers to make decisions about the workload of the caregivers, and the activity of the patients based on insights from the different components of the ecosystem.

Moreover, two software components provide support to other components, the *User Manager* that centralizes the users' data management and the access, and a tool to support the analysis of the data get from the *Discover*, *SocialNet*, and external database with medical and personal information about the patients. Moreover, the ecosystem is constantly evolving; the latest update has been the integration of patients' location data through a NoSQL (non-relational) database that stores them. The *Data Analysis Support* provides the datasets for the dashboard component.

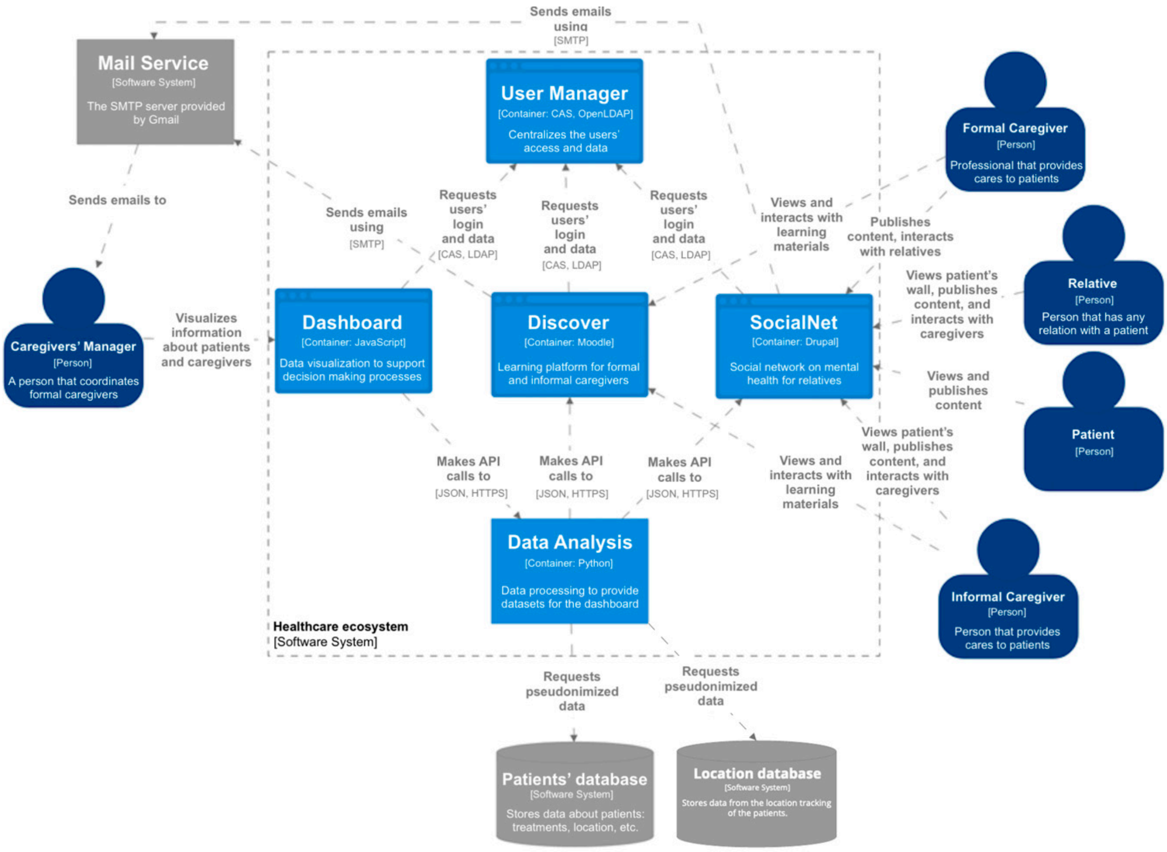


Figure 8. Healthcare ecosystem architecture in C4 Model.

6. Meta-Model Instantiation: A Dashboard for Supporting Caregivers

To illustrate the integration of the two meta-models, we present an instantiation of a dashboard for supporting caregivers based on the healthcare ecosystem described in the previous section. This dashboard gives support to achieve different information goals related to the management of the caregivers. Specifically, two main goals were identified:

- To analyze the relationship between the attention given by relatives and the patient's health;
- To gain insights about the workload of the caregivers.

Caregivers' managers will use this information to make decisions regarding the caregivers' organization to balance their workloads and to understand if some patients need more attention. Three visualizations were proposed to support these goals.

The component selected to support the first goal is a scatter chart that displays information regarding the patient's health and the attention given by their relatives. On the other hand, the caregivers' workload can be visualized in different ways and can involve different variables. In this case, two components have been selected to support the second goal: a heat map and a treemap to let managers identify patterns or relevant data points regarding the caregivers' distribution along time and among patients.

The instantiation of the dashboard was performed through EMF, obtaining a M1-model with specific values for the target dashboard. Figure 9 shows an excerpt of the XMI file containing the instance structure and values of the scatter chart visualization component.

```xml
<container
    name="LeftColumn">
  <component
      xsi:type="dashboardMetamodel:Visualization"
      supports="//@user/@goal.0/@task.0 //@user/@goal.0/@task.1"
      name="ScatterChart"
      title="">
    <primitive
        xsi:type="dashboardMetamodel:Axis"
        id="X">
      <label
          name="XLabel"/>
    </primitive>
    <primitive
        xsi:type="dashboardMetamodel:Axis"
        id="Y"
        type="Y">
      <label
          name="YLabel"/>
    </primitive>
    <primitive
        xsi:type="dashboardMetamodel:Item"
        id="Circle"
        representvar="RelativesAttention PatientsHealth PatientID">
      <channel
          name="YPosition">
        <scale
            scalevar="RelativesAttention"
            range=""
            type="linear"
            name="YScale"/>
      </channel>
      <channel
          name="XPosition">
        <scale
            scalevar="PatientsHealth"
            type="linear"
            name="XScale"/>
      </channel>
    </primitive>
  </component>
</container>
```

Figure 9. Fragment of the Extensible Markup Language (XML) metadata interchange (XMI) instance file containing the information regarding the scatter chart component of the dashboard.

This instance is then handed to a custom Python dashboard generator which "translates" the XMI structure into a set of software components. These components are tailored from a set of core assets developed following the SPL paradigm [65].

The dashboard generator takes the XMI files defining the dashboard as an input, and, depending on the definition of the different visualization, it will select the specific core assets, combining them to obtain the final visualization. For example, to build a bar chart, the generator would compose and configure two axes (X and Y) and visual marks with rectangular shape that encode some variable through their length. The different code fragments that hold the functionality of each primitive element are combined with the support of a template engine [66].

The result of the generation process is the front-end's React source code of a fully functional dashboard with three components that support the initially defined user goals: a scatter plot that shows potential relationships between the patients' given attention and their health; a heatmap that shows the caregivers' workload during different time intervals; and a treemap that shows the proportions of the caregivers' workload by patient (Figure 10). The dashboard receives the data through the configuration of the data sources that are available within the healthcare ecosystem.

Figure 10. Screenshot of the dashboard generated through the XMI instance. Three visualizations are shown: a scatter plot that display potential relationships between the patients' health and the attention given to them; a heatmap that display the aggregated workload of the caregivers during the different days and hours of the week; and a treemap that display the individual workload of each caregiver.

7. Discussion

Knowledge management is a complex task and requires good planning to obtain benefits. Gathering the best available knowledge is not always easy; organizations must understand who holds crucial knowledge, otherwise knowledge management loses all importance [67]. While information can generally be reproduced for minimal costs, knowledge reproduction requires training, apprenticeships, and other more costly forms of transmission [4]. The transformation of tacit knowledge into explicit knowledge implies not only an investment in technology but also the analysis and definition of the knowledge management processes that allow that transformation. According to [67], creating knowledge is a process of organizing data into information that can be analyzed and used to make educated decisions. In this context, the technological ecosystems, specifically the learning ecosystem that is directly focused on knowledge management, provide a framework to develop flexible solutions in which technology is used to establish information flows that support the transformation of knowledge into tangible information inside the organization. However, the main innovation of this type of technological solution is not the technology itself, but the introduction of the human factor at the same level as the software components. Despite this, the learning meta-model has a lack related to the decision-making processes, which are vital to support the transformation of tacit knowledge into explicit knowledge.

Including information dashboards as software tools within the learning ecosystem meta-model enables not only knowledge management but also knowledge discovery. Information dashboards provide different methods of presenting the generated data in an understandable manner, supporting users to identify patterns, relationships, and interesting data points, as well as to reach insights regarding these data.

The development of the presented holistic meta-model is supported by some conceptual classes present in both meta-models. One of these classes is the *User*. The user represents the human factor in both domains. This entity is crucial because it will define (through goals and objectives) the ecosystem's and dashboard's components. This is why the ecosystem's *User* and the dashboard's *User* are merged, obtaining a *User* entity that is related to different objectives (goals) that can be decomposed by different tasks that a series of dashboard components will support. The *User* also has characteristics that will influence these components (as presented in the dashboard meta-model).

On the other hand, dashboards are modeled as specializations of *SoftwareTool*, because the dashboard, in addition to supporting knowledge discovery, is also a tool that is part of the ecosystem.

To illustrate this approach, a dashboard has been generated within the healthcare context. The dashboard for supporting caregivers is part of the healthcare ecosystem. It aims to provide tools based on visual analysis to support decision-making processes related to the workload of caregivers and the patients' situation. The dashboard should be adapted to the different needs of its users. Furthermore, the medical contexts in which the ecosystem could be implemented are very different, so the ecosystem should be adapted to these different contexts.

The dashboard meta-model provides a "template" for generating concrete dashboard solutions and supports the instantiation of fine-grained features regarding visualizations by distinguishing basic primitives that can be combined to build any type of chart. This meta-model has been instantiated to obtain a concrete model for the presented healthcare context. Three visualizations have been selected and instantiated to support the caregivers' managers' goals and decisions with data. However, the dashboard meta-model provides freedom to modify the whole dashboard structure in order to adapt it to other data sources, contexts and audiences. In fact, given that the datasets (and thus, the data sources) are represented within the dashboard meta-model, it is possible to combine different data sources (even if heterogeneous) to unify scattered data into a single information display.

The integration of the meta-models presented not only gives relevance to knowledge discovery by including a dashboard as an important part of learning ecosystems, it also frames dashboards in a bigger picture in which they collaborate with different information flows to enable their users to gain insights.

8. Conclusions

This work proposes an integration of two meta-models to deal with issues related to knowledge generation and knowledge discovery in learning ecosystems. Specifically, a dashboard meta-model has been merged into a learning ecosystem meta-model to support decision-making processes related to the information flows and management objectives that are present within these ecosystems.

The dashboard meta-model provides a skeleton that can be adapted to instantiate concrete dashboard solutions that enable the analysis and visualization of information (datasets) from different sources, such as databases, users' interaction, IoT, sensors, etc. To illustrate the meta-model instantiation process, specific dashboard has been generated through instantiation within a healthcare ecosystem for caregivers, which is, in fact, an instantiation of the learning ecosystem meta-model.

On the other hand, the learning ecosystem meta-model solves several problems related to the definition and implementation of learning ecosystems. Learning ecosystems combine tools to support knowledge management. Including an information dashboard within the learning ecosystem address the improvement of knowledge discovery within the ecosystem by providing a tool to visually analyze information flows.

However, although the learning ecosystem was validated and its quality was checked through the framework defined by López-Fernández, et al. [68], it is necessary to validate and apply the same framework to the holistic meta-model proposed due to the fact that the dashboard meta-model has not been fully validated in previous works.

Furthermore, future research lines will involve the refinement of the meta-model through the addition of constraints, rules, and design guidelines to support a generative pipeline, easing the instantiation process to obtain specific products. Finally, it is also necessary to test the instantiated products with users to validate them and verify their usefulness in real contexts.

Author Contributions: Conceptualization, A.V.-I., A.G.-H., F.J.G.-P. and R.T.; Methodology, A.V.-I., A.G.-H., F.J.G.-P. and R.T.; Software, A.V.-I.; Validation, A.V.-I., A.G.-H., F.J.G.-P. and R.T.; Formal Analysis, A.V.-I., A.G.-H., F.J.G.-P. and R.T.; Investigation, A.V.-I., A.G.-H., F.J.G.-P. and R.T.; Writing—Original Draft Preparation, A.V.-I. and A.G.-H.; Writing—Review & Editing, F.J.G.-P. and R.T.; Supervision, F.J.G.-P. and R.T.; Project Administration, F.J.G.-P.; Funding Acquisition, F.J.G.-P. and A.V.-I. All authors have read and agreed to the published version of the manuscript.

Funding: This research was partially funded by the Spanish Government Ministry of Economy and Competitiveness throughout the DEFINES project grant number [TIN2016-80172-R]. This research was supported by the Spanish *Ministry of Education*, *Culture, and Sport* under a FPU fellowship (FPU17/03276).

Acknowledgments: The authors would like to thank the InterAction and eLearning Research Group (GRIAL) for its support to conduct the present research https://grial.usal.es.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Dufva, T.; Dufva, M. Grasping the future of the digital society. *Futures* **2019**, *107*, 17–28. [CrossRef]
2. Krüger, K. El concepto de 'Sociedad del Conocimiento'. *Rev. Bibliogr. Geogr. Cienc. Soc.* **2006**, *11*, 1–14.
3. UNESCO. *UNESCO World Report: Towards Knowledge Societies*; UNESCO Publishing: Paris, France, 2005.
4. Anderson, R.E. Implications of the Information and Knowledge Society for Education. In *International Handbook of Information Technology in Primary and Secondary Education*; Voogt, J., Knezek, G., Eds.; Springer: Boston, MA, USA, 2008; pp. 5–22. [CrossRef]
5. Castells, M. *La Era de la Información: Economía, Sociedad y Cultura*; Siglo XXI: Mexico City, Mexico, 1999; Volume 1.
6. Nonaka, I.; Takeuchi, H. *The Knowledge-Creating Company: How Japanese Companies Create the Dynamics of Innovation*; Oxford University Press: Oxford, UK, 1995.
7. Hargreaves, A. *Teaching in the Knowledge Society: Education in the Age of Insecurity*; Teachers College Press: New York, NY, USA, 2003.
8. Eraut, M. Non-formal learning and tacit knowledge in professional work. *Br. J. Educ. Psychol.* **2000**, *70*, 113–136. [CrossRef] [PubMed]
9. Davenport, T.H.; De Long, D.W.; Beers, M.C. Successful knowledge management projects. *Sloan Manag. Rev.* **1998**, *39*, 43–57.
10. Camarinha-Matos, L.M.; Rosas, J.; Oliveira, A.I.; Ferrada, F. Care services ecosystem for ambient assisted living. *Enterp. Inf. Syst.* **2013**, *9*, 607–633. [CrossRef]
11. Memon, M.; Wagner, S.R.; Pedersen, C.F.; Beevi, F.H.A.; Hansen, F.O. Ambient Assisted Living Healthcare Frameworks, Platforms, Standards, and Quality Attributes. *Sensors* **2014**, *14*, 4312–4341. [CrossRef]
12. Manikas, K.; Hansen, K.M. Software ecosystems—A systematic literature review. *J. Syst. Softw.* **2013**, *86*, 1294–1306. [CrossRef]
13. Robles, G.; Gonzalez-Barahona, J.M.; Herraiz, I. Evolution of the core team of developers in libre software projects. In Proceedings of the 6th IEEE International Working Conference on Mining Software Repositories(MSR), Vancouver, BC, Canada, 16–17 May 2009; pp. 167–170. [CrossRef]
14. Nakakoji, K.; Yamamoto, Y.; Nishinaka, Y.; Kishida, K.; Ye, Y. Evolution patterns of open-source software systems and communities. In Proceedings of the International Workshop on Principles of Software Evolution, Orlando, FL, USA, 25 May 2002; pp. 76–85.
15. Mens, T.; Claes, M.; Grosjean, P.; Serebrenik, A. Studying evolving software ecosystems based on ecological models. In *Evolving Software Systems*; Mens, T., Serebrenik, A., Cleve, A., Eds.; Springer: Berlin/Heidelberg, Germany, 2014; pp. 297–326. [CrossRef]
16. Weiblen, T.; Giessmann, A.; Bonakdar, A.; Eisert, U. Leveraging the software ecosystem: Towards a business model framework for marketplaces. In Proceedings of the International Conference on Data Communication Networking, e-Business and Optical Communication Systems, Rome, Italy, 24–27 July 2012; pp. 187–193.
17. García-Holgado, A.; García-Peñalvo, F.J. Mapping the systematic literature studies about software ecosystems. In Proceedings of the 6th International Conference on Technological Ecosystems for Enhancing Multiculturality (TEEM 2018), Salamanca, Spain, 24–26 October 2018; pp. 910–918. [CrossRef]

18. García-Peñalvo, F.J.; García-Holgado, A. Preface of Open Source Solutions for Knowledge Management and Technological Ecosystems. In *Open Source Solutions for Knowledge Management and Technological Ecosystems*; García-Peñalvo, F.J., García-Holgado, A., Eds.; IGI Global: Hershey, PA, USA, 2017; pp. 13–22.
19. Llorens-Largo, F.; Molina, R.; Compañ, P.; Satorre, R. Technological Ecosystem for Open education. In *Smart Digital Futures 2014*; Neves-Silva, R., Tsihrintzis, G.A., Uskov, V., Howlett, R.J., Jain, L.C., Eds.; IOS Press: Amsterdam, The Netherlands, 2014; Volume 262, pp. 706–715.
20. Marcos-Pablos, S.; García-Peñalvo, F.J. Technological Ecosystems in Care and Assistance: A Systematic Literature Review. *Sensors* **2019**, *19*, 708. [CrossRef]
21. Kehrer, J.; Hauser, H. Visualization and visual analysis of multifaceted scientific data: A survey. *IEEE Trans. Vis. Comput. Graph.* **2012**, *19*, 495–513. [CrossRef]
22. Keim, D.A.; Mansmann, F.; Schneidewind, J.; Ziegler, H. Challenges in visual data analysis. In Proceedings of the Tenth International Conference on Information Visualisation, London, UK, 5–7 July 2006; pp. 9–16.
23. Berinato, S. *Good Charts: The HBR Guide to Making Smarter, More Persuasive Data Visualizations*; Harvard Business Review Press: Brighton, MA, USA, 2016.
24. Padilla, L. How Do We Know When a Visualization is Good? Perspectives from a Cognitive Scientist. Available online: https://medium.com/multiple-views-visualization-research-explained/how-do-we-know-when-a-visualization-is-good-c894b5194b62 (accessed on 21 July 2020).
25. Hullman, J.; Adar, E.; Shah, P. The impact of social information on visual judgments. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, Vancouver, BC, Canada, 7–12 May 2011; pp. 1461–1470.
26. Kim, Y.-S.; Reinecke, K.; Hullman, J. Data through others' eyes: The impact of visualizing others' expectations on visualization interpretation. *IEEE Trans. Vis. Comput. Graph.* **2018**, *24*, 760–769. [CrossRef]
27. Bedek, M.A.; Nussbaumer, A.; Huszar, L.; Albert, D. Methods for Discovering Cognitive Biases in a Visual Analytics Environment. In *Cognitive Biases in Visualizations*; Springer: New York, NY, USA, 2018; pp. 61–73.
28. Valdez, A.C.; Ziefle, M.; Sedlmair, M. A framework for studying biases in visualization research. In Proceedings of the 2nd DECISIVe Workshop, Phoenix, AZ, USA, 1–6 October 2017.
29. Lee, S.; Kim, S.-H.; Kwon, B.C. Vlat: Development of a visualization literacy assessment test. *IEEE Trans. Vis. Comput. Graph.* **2017**, *23*, 551–560. [CrossRef] [PubMed]
30. Boy, J.; Rensink, R.A.; Bertini, E.; Fekete, J.-D. A principled way of assessing visualization literacy. *IEEE Trans. Vis. Comput. Graph.* **2014**, *20*, 1963–1972. [CrossRef] [PubMed]
31. Maltese, A.V.; Harsh, J.A.; Svetina, D. Data visualization literacy: Investigating data interpretation along the novice—Expert continuum. *J. Coll. Sci. Teach.* **2015**, *45*, 84–90. [CrossRef]
32. Toker, D.; Conati, C.; Carenini, G.; Haraty, M. Towards adaptive information visualization: On the influence of user characteristics. In Proceedings of the International Conference on User Modeling, Adaptation, and Personalization, Berlin, Germany, 23–27 July 2012; pp. 274–285.
33. Dillon, A. Spatial-semantics: How users derive shape from information space. *J. Am. Soc. Inf. Sci.* **2000**, *51*, 521–528. [CrossRef]
34. Conati, C.; Carenini, G.; Hoque, E.; Steichen, B.; Toker, D. Evaluating the impact of user characteristics and different layouts on an interactive visualization for decision making. *Comput. Graph. Forum* **2014**, *33*, 371–380. [CrossRef]
35. Patil, D.; Mason, H. *Data Driven*; O'Reilly Media, Inc.: Sebastopol, CA, USA, 2015.
36. García-Holgado, A.; García-Peñalvo, F.J. Validation of the learning ecosystem metamodel using transformation rules. *Future Gener. Comput. Syst.* **2019**, *91*, 300–310. [CrossRef]
37. García-Holgado, A.; Marcos-Pablos, S.; García-Peñalvo, F.J. A Model to Define an eHealth Technological Ecosystem for Caregivers. In *New Knowledge in Information Systems and Technologies. Advances in Intelligent Systems and Computing*; Rocha, Á., Adeli, H., Reis, L., Costanzo, S., Eds.; Springer: Cham, Switzerland, 2019; Volume 932, pp. 422–432.
38. *World Population Prospects: The 2017 Revision, Key Findings and Advance Tables*; ESA/P/WP/248; United Nations, Department of Economic and Social Affairs, Population Division: New York, NY, USA, 2017.
39. García-Holgado, A.; Marcos-Pablos, S.; Therón, R.; García-Peñalvo, F.J. Technological ecosystems in the health sector: A mapping study of European research projects. *J. Med. Syst.* **2019**, *43*. [CrossRef]

40. Marcos-Pablos, S.; García-Holgado, A.; García-Peñalvo, F.J. Trends in European research projects focused on technological ecosystems in the health sector. In Proceedings of the 6th International Conference on Technological Ecosystems for Enhancing Multiculturality (TEEM 2018), Salamanca, Spain, 24–26 October 2018; pp. 495–503. [CrossRef]
41. Jesse, N. Internet of Things and Big Data—The Disruption of the Value Chain and the Rise of New Software Ecosystems. *IFAC PapersOnLine* **2016**, *49*, 275–282. [CrossRef]
42. Logre, I.; Mosser, S.; Collet, P.; Riveill, M. Sensor data visualisation: A composition-based approach to support domain variability. In Proceedings of the European Conference on Modelling Foundations and Applications, York, UK, 21–25 July 2014; pp. 101–116.
43. Vanden Hautte, S.; Moens, P.; Van Herwegen, J.; De Paepe, D.; Steenwinckel, B.; Verstichel, S.; Ongenae, F.; Van Hoecke, S. A Dynamic Dashboarding Application for Fleet Monitoring Using Semantic Web of Things Technologies. Sensors. *Sensors* **2020**, *20*, 1152. [CrossRef]
44. Vázquez-Ingelmo, A.; García-Peñalvo, F.J.; Therón, R. Information Dashboards and Tailoring Capabilities—A Systematic Literature Review. *IEEE Access* **2019**, *7*, 109673–109688. [CrossRef]
45. Lavalle, A.; Maté, A.; Trujillo, J. Requirements-Driven Visualizations for Big Data Analytics: A Model-Driven Approach. In Proceedings of the Conceptual Modeling, Salvador, Brazil, 4–7 November 2019; pp. 78–92.
46. Golfarelli, M.; Rizzi, S. A model-driven approach to automate data visualization in big data analytics. *Inf. Vis.* **2019**, *19*, 24–47. [CrossRef]
47. Vázquez-Ingelmo, A.; García-Peñalvo, F.J.; Therón, R.; Conde González, M.Á. Extending a dashboard meta-model to account for users' characteristics and goals for enhancing personalization. In Proceedings of the Learning Analytics Summer Institute (LASI) Spain, Vigo, Spain, 27–28 June 2019.
48. Vázquez-Ingelmo, A.; García-Peñalvo, F.J.; Therón, R. Capturing high-level requirements of information dashboards' components through meta-modeling. In Proceedings of the 7th International Conference on Technological Ecosystems for Enhancing Multiculturality (TEEM 2019), León, Spain, 16–18 October 2019.
49. Pleuss, A.; Wollny, S.; Botterweck, G. Model-driven development and evolution of customized user interfaces. In Proceedings of the 5th ACM SIGCHI Symposium on Engineering Interactive Computing Systems, London, UK, 23–26 June 2013; pp. 13–22.
50. Kleppe, A.G.; Warmer, J.; Bast, W. *MDA Explained. The Model Driven Architecture: Practice and Promise*; Addison-Wesley Longman Publishing Co., Inc.: Boston, MA, USA, 2003.
51. Mellor, S.J.; Scott, K.; Uhl, A.; Weise, D. Model-Driven Architecture. In Proceedings of the Advances in Object-Oriented Information Systems: OOIS 2002 Workshops, Montpellier, France, 2 September 2002; pp. 290–297. [CrossRef]
52. Álvarez, J.M.; Evans, A.; Sammut, P. Mapping between Levels in the Metamodel Architecture. In *≪UML≫ 2001—The Unified Modeling Language. Modeling Languages, Concepts, and Tools. UML 2001. Lecture Notes in Computer Science*; Gogolla, M., Kobryn, C., Eds.; Springer: Berlin/Heidelberg, Germany, 2001; Volume 2185, pp. 34–46.
53. Kang, K.C.; Cohen, S.G.; Hess, J.A.; Novak, W.E.; Peterson, A.S. *Feature-Oriented Domain Analysis (FODA) Feasibility Study*; Carnegie-Mellon University, Software Engineering Institute: Pittsburgh, PA, USA, 1990.
54. Voelter, M.; Visser, E. Product line engineering using domain-specific languages. In Proceedings of the 15th International Software Product Line Conference (SPLC), Munich, Germany, 22–26 August 2011; pp. 70–79.
55. García-Holgado, A.; García-Peñalvo, F.J. A Metamodel Proposal for Developing Learning Ecosystems. In *Learning and Collaboration Technologies. Novel Learning Ecosystems. 4th International Conference, LCT 2017. Held as Part of HCI International, Vancouver, BC, Canada, 9–14 July 2017*; Zaphiris, P., Ioannou, A., Eds.; Springer International Publishing: Cham, Switzerland, 2017; Volume 10295, pp. 100–109.
56. Lavalle, A.; Maté, A.; Trujillo, J.; Rizzi, S. Visualization Requirements for Business Intelligence Analytics: A Goal-Based, Iterative Framework. In Proceedings of the 27th International Requirements Engineering Conference (RE), Lille, France, 23–27 September 2019; pp. 109–119.
57. Vázquez-Ingelmo, A.; García-Peñalvo, F.J.; Therón, R. Tailored information dashboards: A systematic mapping of the literature. In Proceedings of the Interacción 2019, Donostia, Spain, 25–28 June 2019.
58. Brehmer, M.; Munzner, T. A multi-level typology of abstract visualization tasks. *IEEE Trans. Vis. Comput. Graph.* **2013**, *19*, 2376–2385. [CrossRef] [PubMed]

59. Amar, R.; Eagan, J.; Stasko, J. Low-level components of analytic activity in information visualization. In Proceedings of the IEEE Symposium on Information Visualization, Minneapolis, MN, USA, 23–25 October 2005; pp. 111–117.
60. Lam, H.; Tory, M.; Munzner, T. Bridging from goals to tasks with design study analysis reports. *IEEE Trans. Vis. Comput. Graph.* **2017**, *24*, 435–445. [CrossRef] [PubMed]
61. Munzner, T. *Visualization Analysis and Design*; AK Peters/CRC Press: Boca Raton, FL, USA, 2014.
62. Ware, C. *Information Visualization: Perception for Design*; Elsevier: New York, NY, USA, 2012.
63. Bäuml, J.; Froböse, T.; Kraemer, S.; Rentrop, M.; Pitschel-Walz, G. Psychoeducation: A basic psychotherapeutic intervention for patients with schizophrenia and their families. *Schizophr. Bull.* **2006**, *32*, S1–S9. [CrossRef] [PubMed]
64. Toribio-Guzmán, J.M.; García-Holgado, A.; Soto Pérez, F.; García-Peñalvo, F.J.; Franco Martín, M.Á. Usability Evaluation of a Private Social Network on Mental Health for Relatives. *J. Med. Syst.* **2017**, *41*, 137. [CrossRef]
65. Vázquez-Ingelmo, A.; García-Peñalvo, F.J.; Therón, R. Taking advantage of the software product line paradigm to generate customized user interfaces for decision-making processes: A case study on university employability. *PeerJ Comput. Sci.* **2019**, *5*, e203. [CrossRef]
66. Vázquez-Ingelmo, A.; García-Peñalvo, F.J.; Therón, R. Addressing Fine-Grained Variability in User-Centered Software Product Lines: A Case Study on Dashboards. In Proceedings of the World Conference on Information Systems and Technologies, Galicia, Spain, 16–19 April 2019; pp. 855–864.
67. Stevens, R.H.; Millage, J.; Clark, S. Waves of Knowledge Management: The Flow between Explicit and Tacit Knowledge. *Am. J. Econ. Bus. Adm.* **2010**, *2*, 129–135. [CrossRef]
68. López-Fernández, J.J.; Guerra, E.; de Lara, J. Assessing the Quality of Meta-models. In *MoDeVVa*; Boulanger, F., Famelis, M., Ratiu, D., Eds.; CEUR Workshop Proceedings: Valencia, Spain, 2014; Volume 1235, pp. 3–22.

7.17 Appendix Q. Aggregation Bias: A Proposal to Raise Awareness Regarding Inclusion in Visual Analytics

Vázquez-Ingelmo, A., García-Peñalvo, F. J., & Therón, R. (2020). Aggregation Bias: A Proposal to Raise Awareness Regarding Inclusion in Visual Analytics. In Á. Rocha, H. Adeli, L. P. Reis, S. Costanzo, I. Orovic, & F. Moreira (Eds.), Trends and Innovations in Information Systems and Technologies, WorldCIST 2020 (Vol. 3, pp. 409-417). Springer Nature. https://doi.org/10.1007/978-3-030-45697-9_40

Aggregation Bias: A Proposal to Raise Awareness Regarding Inclusion in Visual Analytics

Andrea Vázquez-Ingelmo[1(✉)], Francisco J. García-Peñalvo[1], and Roberto Therón[1,2]

[1] GRIAL Research Group, Computer Sciences Department, Research Institute for Educational Sciences, University of Salamanca, Salamanca, Spain
{andreavazquez, fgarcia, theron}@usal.es
[2] VisUSAL Research Group, University of Salamanca, Salamanca, Spain

Abstract. Data is a powerful tool to make informed decisions. They can be used to design products, to segment the market, and to design policies. However, trusting so much in data can have its drawbacks. Sometimes a set of indicators can conceal the reality behind them, leading to biased decisions that could be very harmful to underrepresented individuals, for example. It is challenging to ensure unbiased decision-making processes because people have their own beliefs and characteristics and be unaware of them. However, visual tools can assist decision-making processes and raise awareness regarding potential data issues. This work describes a proposal to fight biases related to aggregated data by detecting issues during visual analysis and highlighting them, trying to avoid drawing inaccurate conclusions.

Keywords: Data bias · Information visualization · Data visualization · Inclusion awareness

1 Introduction

Information has grown in size and relevance over the last years; technology has not only increased the generation of data but also their accessibility. People with an Internet connection can consult a wide range of datasets about almost any topic: crime data, healthcare data, weather data, financial data, etc.

These data can be employed to make informed decisions regarding different domains. For example, businesses can employ demographic data to create personalized advertisements or to segment the market. Governments can employ their data to design new policies. Any person regularly uses data to make informed decisions. A simple question like "should I get a coat to go out today?" can be answered through data (made available by weather services) to make an informed decision that, in the end, seeks some kind of benefit (in this case, the benefit of avoiding hypothermia).

However, delegating decisions solely in data might turn out to be a two-edged sword. Data not only can be wrong or false, but it can also be incomplete, and making decisions using wrong data leads to wrong decisions. There are several cases in which relying on the wrong data has provoked undesired results, mostly because of data bias or even algorithmic bias [1–3].

So it seems clear that if the data that you are using to make decisions is not the best for your problem, you could end up with decisions that are also not the best for your problem. But how can people avoid such inconveniences with data? Bias is generally introduced unconsciously, and it can be hard to detect our own biases and be aware of them while collecting data. For these reasons, data should be thoroughly examined to identify gaps or inconsistencies before using them in decision-making processes.

One of the most used methods to ease the analysis and exploration of datasets is visual analytics [4, 5]; using information visualizations, users can interact and explore datasets through visual marks that encode certain information [6]. However, visualizations could hide data issues by lifting the attention from the analysis process carried out on the raw data to the discovered patterns. Patterns can be seen as shortcuts that tell us properties about the data, for example, if there are correlations among the visualized variables [7]. But visual analysis shouldn't be reduced to just the identification of patterns and to trust them blindly, because patterns can likewise lead to wrong conclusions [8].

This work describes a proposal for raising awareness during visual analysis, helping users to make informed decisions taking into account the flaws or potential issues of their datasets. Specifically, issues related to data aggregation, which can be very harmful in data-driven decision-making processes. The main goal is not only to improve decision-making, but to address inclusion problems when dealing with data, as data biases can lead to decisions that (involuntarily, or not) discriminate individuals.

The rest of this paper is organized as follows. Section 2 introduces some issues related to data analysis and data aggregation. Section 3 describes the methodology followed to design the proposal. Section 4 presents a proposal to raise awareness during visual data analysis. Section 5 discusses the proposal, following by Sect. 6, in which the conclusions derived from this work are outlined.

2 Background

The outcomes of decision-making processes are actions that affect the context in which decisions are being made. When deciding which action to take, the decision-maker will have an assumption on how the action's effect will affect the context, looking for a benefit or a pursued result. However, the critical fact is that assumptions can be very personal and could vary depending on the person's beliefs, background, domain knowledge, etc.

Even when the decision-maker support its decisions on data (embracing data-driven decision-making [9]), there are still problems. As introduced before, data is not the holy grail of decision-making, because as well as personal traits can influence the decision-maker, the collected data and performed analyses can be influenced by other harmful factors like data biases [10] or poor analysis.

There are specific fields of study, like uncertainty visualization, that try to find methods to visualize uncertain data, thus warning users regarding the uncertain nature of the results they are consuming through their displays [11, 12]. However, uncertainty visualization is complex, and several concepts could be difficult to understand by non-technical or non-statistical audiences, such as probabilities or densities, resulting in users ignoring or misinterpreting uncertainty [13].

On the other hand, the data that is being visualized can present issues that could be concealed and not considered through information visualizations, like excessive (or not appropriate) aggregation levels, which could result in wrong conclusions.

Summary statistics summarize a set of observations through a collection of values that simplify the comprehension of the datasets. But this simplification comes with a price; while performing these summaries, a lot of information can be lost. One of the most famous examples of this drawback is Anscombe's quartet [14], in which different datasets that tell very different stories have the same mean and variance. Anscombe highlighted the usefulness of graphics [14] to avoid these issues (Fig. 1).

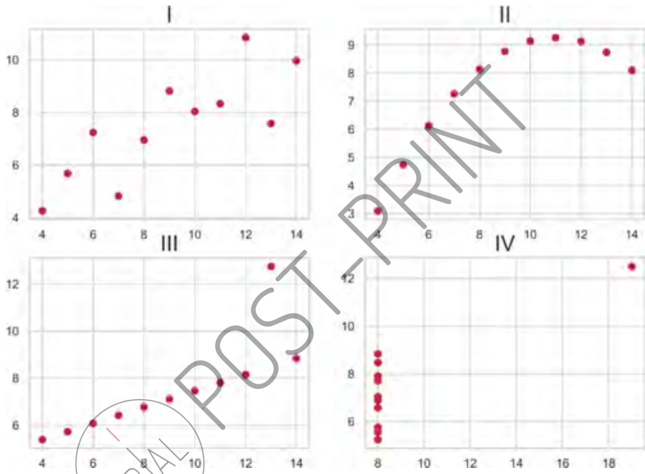


Fig. 1. The Anscombe's Quartet. The four datasets have the same mean and variance values on both variables represented on the X and Y axes.

So aggregated data ease the analysis process, but they can lead to a loss of information. Aggregated data can also be vulnerable to phenomena like the ecological fallacy and the Simpson's paradox [15].

Inferring individual behavior by using aggregated data is a common extrapolation mistake, where analysts might conclude that the behavior of a group is also accurate to explain the behavior of the individuals within that group [16, 17].

Simpson's paradox is also related to the data aggregation-level. In this case, there might exist lurking variables that could entirely "change" the conclusions derived from aggregated data [18, 19].

These aggregation-related issues can be very harmful if not taken into account [20], especially if the audience is biased or not statistically-trained (or both).

Some works have tried to address these aggregation drawbacks through detection algorithms [21, 22], but a few tried to address them during visual exploration [23].

3 Methodology

The proposal focuses on how to draw attention to potential aggregation biases and fallacies during visual analysis. A simple workflow has been considered to automatically seek for aggregation issues regarding the data being presented to the user. Specifically, issues involving the Simpson's paradox and underrepresentation of categories.

Each categorical variable is considered as a potentially influencing variable. Of course, as it will be discussed, this methodology is limited to the available variables within the dataset. If the whole dataset has a small set of categories, the results would not be as useful as it could be with a richer dataset.

The workflow follows a naïve approach to detect Simpson's paradoxes [23]:

1. Every possible grouping at any possible level is computed on categorical to obtain a set of potential disaggregation variables.
2. When the user visualizes data, the current aggregation level is retrieved (i.e., the categorical columns used to group the data)
3. These data are then grouped by the variables identified in the first step.
4. The results of the performed disaggregation are sorted and compared with the original scenario (i.e., the aggregated data values) trend.
5. If the disaggregation results differ from the originally aggregated results (a threshold can be defined to specify which proportion of values need differ from the original trend to consider the paradox), the Simpson's paradox is considered for the disaggregated attributes

However, even visualizing the disaggregated data by the identified attributes in the fifth step, there could still be aggregation issues if data are in turn aggregated by a function such as the mean, mode, ratios, etc. These functions can, in turn, distort the reality of data.

To avoid relying on aggregation functions, when the detected Simpson's paradoxes are inspected, a sunburst diagram complements the display to give information about the raw data sample sizes regarding the disaggregated values.

Sunburst diagrams are usually employed to represent hierarchies; in this context, they are useful to display how the number of observations of the variable being inspected varies its size among the different disaggregation levels.

The primary purpose is to have another perspective of data, drawing attention over potential underrepresentation or overrepresentation in datasets.

4 Proposal

A simple proof-of-concept has been developed to illustrate the proposal. The employed test data is from one of the most famous cases involving Simpson's paradox: the student admission at UC Berkeley in 1975 [24]. This dataset holds the following information about each student: gender, the department in which the application was issued, and the result of the application (admitted or rejected).

If the gender variable aggregates this data, the results yield a significant gender bias against women: only 35% of women were admitted, in contrast with the 44% of admitted males. This data could help the decision-makers to design new policies trying to address the discovered gender bias.

However, this high-level aggregation hides some parts of the picture. If data is, in turn, disaggregated using the department in which the application was issued, we see a different scenario: the majority of the departments shown higher admission rates for women than men. What was happening is that women applied to more competitive departments than men, who issued the majority of applications to departments with a high rate of admissions (resulting in higher admissions rates among male students).

This case is a famous example of Simpson's paradox, but misleading conclusions can be present in any context if these potential issues in data analysis are not accounted for. For this reason, the interface presented in Fig. 2 is proposed.

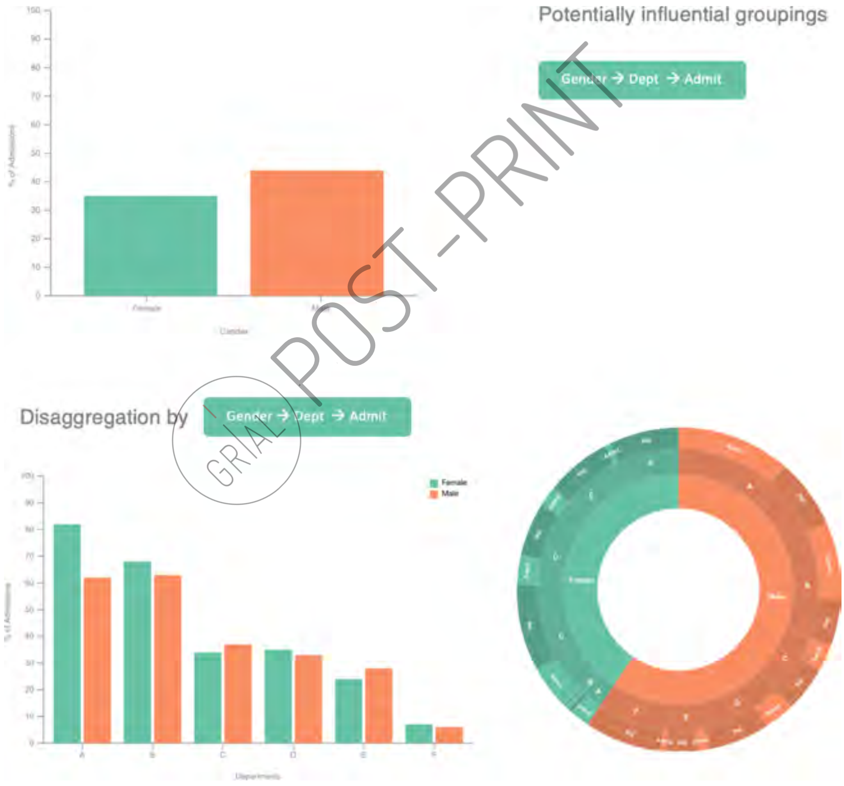


Fig. 2. Interface proposal for detecting aggregation issues.

When the user is exploring her dataset, Simpson's paradox detector starts searching for potentially influential groupings that change the trend of the currently displayed variables. If any grouping changes the trend, the categorical variables identified are displayed (top section of Fig. 2).

The user then can click on each detected grouping to explore how the disaggregation affects the value that she was examining, in addition to a sunburst diagram that shows the distribution of occurrences of each observation under the selected grouping (bottom section of the Fig. 2). In this specific example, the user can observe how women apply less to departments with high admission rates (like department A, for example) and issue more applications to more competitive departments, obtaining a complete view of the examined data.

5 Discussion

Aggregating data is useful to summarize observations, but it can overlook crucial aspects of data, like, for example, underrepresentation or overrepresentation of the samples. Raising attention over this matter is essential, especially when studying behavioral data or data that involve human beings.

How can this approach benefit decision-makers regarding inclusion-related issues? Our biases could blind ourselves and make us not prone to ask skeptical questions about the analyzed data. If data confirms something we believe, we might trust the results without carrying out further analyses [25].

This approach forces analysts (or any kind of audience) to have a more in-depth look at aggregated data, which sometimes can conceal underlying patterns or trends. Having a deeper look is crucial when analyzing data for inclusion-related research contexts because it is possible to visualize if aggregated results are due to the over-representation of certain categories and to identify if any category is missing or not represented at all.

Not considering aggregation issues can strengthen the belief that “one size fits all”, which can lead to (involuntary) discrimination. If you design a product (referring to an object, an algorithm, a policy, a treatment, etc.) for “people” and you use data that only represent a particular portion of people or don’t bring attention to their differing characteristics, you end up with a personalized product for a segment. There is nothing wrong with personalized products; what is wrong is to think that this unconsciously personalized product is universal and should fit every individual.

The underrepresentation of certain categories depends, of course, on the data context. For example, in the Berkeley dataset, the underrepresentation of women’s applications to some departments is due to the preference of the students to apply to specific departments. But there are other cases in which the underrepresentation is due to selection bias or a not representative sampling of the population. It is essential to take this into account to avoid data bias against minorities (or even against non-minorities, like women [20]).

A proposal for visually identifying aggregation issues (especially those related to the Simpson’s paradox) has been developed. Of course, this proposal does not try at all to replace statistical methods but to deliver a visual tool to understand better our datasets.

The proposal has been focused on raising awareness regarding how disaggregating data could change the patterns identified during the analysis of aggregated data. It also could be used as an informative tool to educate people through a friendly interface regarding the underlying issues of data aggregation and their dangerous effects on decision-making processes.

Educating people in data skepticism and regarding potential biases is important because data visualizations can be very persuasive and could influence people's beliefs. Relying on data visualizations tools to raise awareness can be powerful due to the possibility of presenting information in understandable manners and also to the possibility of enabling individuals to freely interact with data [26, 27].

The methodology seeks for sub-groups that "change" the original scenario (i.e., the trends identified on aggregated data). It is important to mention that, in this case, statistical significance has not been considered because the main goal was to draw attention to changes in visual patterns, no matter how small. However, complementing this methodology with the computation of statistical significance could be more powerful in some contexts [23].

Statistically-trained audiences might be aware of these issues. However, other audiences could reach wrong insights about data if attention is not raised regarding potential issues, thus distorting the decision-making process without even notice.

For example, when dealing with policies that affect individuals, it is crucial to rely on disaggregated data to avoid ignoring the necessities of minorities [28–30].

But when talking about disaggregated data, there are some limitations to take into account. Demographic variables are meaningful for inclusion-related research contexts, but also sensitive. Some of these variables can be difficult to collect because of privacy policies or privacy concerns.

In fact, for some activities as for example, hiring people, having such data available could introduce the risk of biasing the decisions made during some phases of the process [31, 32]. So analysts and decision-makers must understand the level of analysis and goals to anonymize or omit these attributes accordingly.

To sum up, it is important to foster critical thinking and some skepticism toward data. When dealing with information about individuals, accounting for data gaps is a responsibility, because the decisions made could have a high impact in the context of application, and sometimes, this impact is not beneficial for everyone.

6 Conclusions

This work presents a proposal for raising awareness in decision-making processes through visual analysis. Relying on inappropriate data could lead to wrong decisions. But identifying flaws in data is not a trivial task; bias, beliefs, and uncertainty can show up both at data collection time and analysis time, resulting in distorted insights.

Through the detection of existing Simpson's Paradox and the disaggregation of the displayed data, the presented proposal tries to draw attention to issues like excessive or inappropriate aggregation levels and potential overrepresentation or underrepresentation of data attributes or categories.

Future work will involve the evaluation and refinement of the proposal to improve its effectiveness to obtain a tool to raise awareness about inclusion in different fields.

Acknowledgments. This research work has been supported by the Spanish *Ministry of Education and Vocational Training* under an FPU fellowship (FPU17/03276). This work has been partially funded by the Spanish Government Ministry of Economy and Competitiveness throughout the DEFINES project (Ref. TIN2016-80172-R) and the Ministry of Education of the Junta de Castilla y León (Spain) throughout the T-CUIDA project (Ref. SA061P17).

References

1. Sweeney, L.: Discrimination in online ad delivery. arXiv preprint arXiv:1301.6822 (2013)
2. Garcia, M.: Racist in the machine: the disturbing implications of algorithmic bias. World Policy J. **33**, 111–117 (2016)
3. Hajian, S., Bonchi, F., Castillo, C.: Algorithmic bias: from discrimination discovery to fairness-aware data mining. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 2125–2126. ACM (2016)
4. Keim, D.A., Andrienko, G., Fekete, J., Görg, C., Kohlhammer, J., Melançon, G.: Visual analytics: definition, process, and challenges. In: Kerren, A., Stasko, J., Fekete, J., North, C. (eds.) Information Visualization, pp. 154–175. Springer, Heidelberg (2008)
5. Thomas, J.J., Cook, K.A.: Illuminating the path: the research and development agenda for visual analytics. National Visualization and Analytics Center, USA (2005)
6. Munzner, T.: Visualization Analysis and Design. AK Peters/CRC Press, Boca Raton (2014)
7. Harrison, L., Yang, F., Franconeri, S., Chang, R.: Ranking visualizations of correlation using weber's law. IEEE Trans. Visual Comput. Graph. **20**, 1943–1952 (2014)
8. O'Neil, C.: On Being a Data Skeptic. O'Reilly Media, Inc., Newton (2013)
9. Patil, D., Mason, H.: Data Driven. O'Reilly Media Inc, Newton (2015)
10. Shah, S., Horne, A., Capellá, J.: Good data won't guarantee good decisions. Harvard Bus. Rev. **90**, 23–25 (2012)
11. Bonneau, G.-P., Hege, H.-C., Johnson, C.R., Oliveira, M.M., Potter, K., Rheingans, P., Schultz, T.: Overview and state-of-the-art of uncertainty visualization. In: Scientific Visualization, pp. 3–27. Springer, Heidelberg (2014)
12. Brodlie, K., Osorio, R.A., Lopes, A.: A review of uncertainty in data visualization. In: Expanding the Frontiers of Visual Analytics and Visualization, pp. 81–109. Springer, Heidelberg (2012)
13. https://medium.com/multiple-views-visualization-research-explained/uncertainty-visualization-explained-67e7a73f031b
14. Anscombe, F.J.: graphs in statistical analysis. Am. Stat. **27**, 17–21 (1973)
15. Pollet, T.V., Stulp, G., Henzi, S.P., Barrett, L.: Taking the aggravation out of data aggregation: a conceptual guide to dealing with statistical issues related to the pooling of individual-level observational data. Am. J. Primatol. **77**, 727–740 (2015)
16. Kramer, G.H.: The ecological fallacy revisited: aggregate-versus individual-level findings on economics and elections, and sociotropic voting. Am. Polit. Sci. Rev. **77**, 92–111 (1983)
17. Piantadosi, S., Byar, D.P., Green, S.B.: The ecological fallacy. Am. J. Epidemiol. **127**, 893–904 (1988)
18. Blyth, C.R.: On Simpson's paradox and the sure-thing principle. J. Am. Stat. Assoc. **67**, 364–366 (1972)
19. Wagner, C.H.: Simpson's paradox in real life. Am. Stat. **36**, 46–48 (1982)

20. Perez, C.C.: Invisible Women: Exposing Data Bias in a World Designed for Men. Random House, New York (2019)
21. Alipourfard, N., Fennell, P.G., Lerman, K.: Can you trust the trend?: discovering Simpson's paradoxes in social data. In: Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining, pp. 19–27. ACM (2018)
22. Xu, C., Brown, S.M., Grant, C.: Detecting Simpson's paradox. In: The Thirty-First International Flairs Conference (2018)
23. Guo, Y., Binnig, C., Kraska, T.: What you see is not what you get!: detecting Smpson's paradoxes during data exploration. In: Proceedings of the 2nd Workshop on Human-In-the-Loop Data Analytics, p. 2. ACM (2017)
24. Bickel, P.J., Hammel, E.A., O'Connell, J.W.: Sex bias in graduate admissions: data from Berkeley. Science **187**, 398–404 (1975)
25. Nickerson, R.S.: Confirmation bias: a ubiquitous phenomenon in many guises. Review of general psychology **2**, 175–220 (1998)
26. Hullman, J., Adar, E., Shah, P.: The impact of social information on visual judgments. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, pp. 1461–1470. ACM (2011)
27. Kim, Y.-S., Reinecke, K., Hullman, J.: Data through others' eyes: the impact of visualizing others' expectations on visualization interpretation. IEEE Trans. Visual Comput. Graph. **24**, 760–769 (2018)
28. Mills, E.: 'Leave No One Behind': Gender, Sexuality and the Sustainable Development Goals. IDS (2015)
29. Stuart, E., Samman, E.: Defining "leave no one behind". ODI Briefing Note. London: ODI (www.odi.org/sites/odi.org.uk/files/resource-documents/11809.pdf) (2017)
30. Abualghaib, O., Groce, N., Simeu, N., Carew, M.T., Mont, D.: Making visible the invisible: why disability-disaggregated data is vital to "leave no-one behind". Sustainability **11**, 3091 (2019)
31. Rice, L., Barth, J.M.: Hiring decisions: the effect of evaluator gender and gender stereotype characteristics on the evaluation of job applicants. Gend. Issues **33**, 1–21 (2016)
32. Alford, H.L.: Gender bias in IT hiring practices: an ethical analysis (2016)

7.18 Appendix R. A meta-model to develop learning ecosystems with support for knowledge discovery and decision-making processes

Vázquez-Ingelmo, A., García-Holgado, A., García-Peñalvo, F. J., & Therón, R. (2020). A meta-model to develop learning ecosystems with support for knowledge discovery and decisionmaking processes. In A. Rocha, B. E. Pérez, F. J. García-Peñalvo, M. M. Miras, & R. Gonçalves (Eds.), Proceedings of the 2020 15th Iberian Conference on Information Systems and Technologies (CISTI) (Seville, Spain, 24-27 June 2020). IEEE. https://doi.org/10.23919/CISTI49556.2020.9140986

A meta-model to develop learning ecosystems with support for knowledge discovery and decision-making processes

Andrea Vázquez-Ingelmo, Alicia García-Holgado, Francisco J. García-Peñalvo, Roberto Therón
Grupo de Investigación GRIAL, Dpto. de Informática y Automática
University of Salamanca
Salamanca, Spain
{andreavazquez, aliciagh, fgarcia, theron}@usal.es

Abstract — **There are software solutions to solve most of the problems related to information management in any company or institutions, but still, there is a problem for transforming information into knowledge. Technological ecosystems emerge as a solution to combine existing tools and human resources to solve different problems of knowledge management. In particular, when the ecosystem is focused on learning processes associated with knowledge are named learning ecosystems. The learning ecosystem metamodel defined in previous works solves several problems related to the definition and implementation of these solutions. However, there are still challenges associated with improving the analysis and visualization of information as a way to discover knowledge and support decision making processes. On the other hand, there is a metamodel proposal to define customized dashboards for supporting decision-making processes. This proposal aims to integrate both metamodels as a way to improve the definition of learning ecosystems.**

Keywords – dashboard; software ecosystems; MDA; metamodel; software engineering; learning ecosystem; knowledge management; decision-makers.

I. INTRODUCTION

Two main concepts are used to refer to changes in nowadays society, information society and knowledge society. The notion of the 'information society' is used above all when dealing with technological aspects and their effects on economic growth and employment [1]. On the other hand, in the knowledge society, the core element is not the technology but the ability to identify, produce, process, transform, disseminate and use the information to build and apply knowledge for human development [2]. According to [1], Heidenreich [3] associates the knowledge society to the increasing relevance of education and training processes in the initial phase as well as in the whole life course is underlined, and the increasing weight of knowledge-intensive services and communication.

Knowledge has become the backbone for development; it is a strategic factor for creating new policies, to plan new actions, and to foster innovation within organizations. Knowledge management is considered a sustainable competitive advantage [4], so the organizations expend part of their resources on building their capacity to share, create, and apply new knowledge continuously over time [5].

However, knowledge is not only present physically (i.e., in documents or books), it is also present in employees and the different processes carried out at organizations. According to [4], knowledge management processes must be able to support the transfer of implicit knowledge to tacit knowledge. This scattered nature of knowledge makes its management a complex and crucial task.

Software ecosystems emerge as a technological solution to support information and knowledge management in different contexts. According to [6, 7], institutions adopt a software ecosystem strategy to expand their organizational boundaries, share their platforms and resources with third parties, and define new business models.

Although the term software ecosystem is the most used in the literature, there are other terms that have distinctive characteristics. This is the case of the technological ecosystems, solutions that propose a decentralized configuration of software tools and non-technological components (such as methodologies, management plans, or human resources). Technological ecosystems can be composed by several elements; elements that are connected to each other and have different functions within the ecosystem. One of the main strengths of the technological ecosystem is that when their components collaborate, they exploit all of their benefits, obtaining the most out of their functionalities to provide elaborate services.

Furthermore, when the technological ecosystem is focused on learning processes associated with knowledge are named learning ecosystems [8, 9]. The definition and development of these solutions have challenges associated with the evolution of its components and the whole ecosystem, as well as the need to adapt to the changes that constantly occur in any organization.

This work presents a holistic meta-model to support decision-making processes in learning ecosystems. This meta-model integrates two meta-models defined in previous works. First, a learning ecosystem meta-model to support the definition of learning ecosystems based on open source software [10]. On the other hand, a dashboard meta-model to support the analysis of information in order to transform implicit knowledge in tacit knowledge.

Information dashboards are powerful tools that allow the recognition of patterns and interesting data points through visual analysis . However, dashboards can be very diverse in terms of design, context, audience, pursued goals, supported tasks, etc., [11-14], which makes the development of these tools a complex activity. By abstracting the common elements of information dashboards through meta-modeling it is possible to obtain a general structure of dashboards that can be instantiated and adapted to any kind of contexts, data domains or audiences.

Furthermore, as it will be detailed, the inclusion of dashboard users and their requirements as elements of the meta-model enables the integration of this meta-model as a part of technological ecosystems, specifically, learning ecosystems, providing support to discover knowledge and support decision making processes [15].

A meta-model allows the definition of platform independent models, thus obtaining a high-level specification of rules, constraints, entities and relationships found within a specific domain. Meta-models can be subsequently instantiated to obtain particular models adapted to a specific problem.

The rest of this paper is organized as follows. Section 2 outlines the methodology followed to develop the meta-models. Section 3 describes the learning ecosystem meta-model, followed by section 4, in which the dashboard meta-model is presented. Section 4 discusses the integration of both meta-models. Finally, section 5, where the conclusions derived from this work are depicted.

II. Methodology

Model-driven development (MDD) [16] allows separating the data and the operations specification of the system from lower-level details, like the technical aspects related to a specific program language or platforms.

The Object Management Group (OMG) proposes the model-driven architecture (MDA) as a guideline to implement this approach. This architecture provides a framework for software development which employs models to describe and define the target system [17]. The main difference between MDD and MDA is that the OMG proposal uses a set of standards: meta-object facility (MOF), unified modeling language (UML), XML (Extensible Markup Language) metadata interchange (XMI), and query/view/transformation (QVT).

In this case, the dashboard is a part of the learning ecosystem, which is based on a meta-model defined and validated in previous works. The first version of the learning ecosystem meta-model is based on MOF, and the last validated version is an instance of Ecore [10]. Both versions are M2-models. The model has served as a map to develop and deploy the ecosystem in a real-world context.

The dashboard meta-model is also part of the four-layer meta-model architecture proposed by the OMG, in which a model at one layer is used to specify models in the layer below [18]. In particular, the dashboard meta-model is an instance of MOF (i.e., an M2-model), so it can be instantiated to obtain M1-models.

The integration of both meta-models is possible because of both are Platform Independent Models (PIM) at M2 layer, although one is instantiated from Ecore (learning ecosystem meta-model) and other from MOF (dashboard meta-model). To get the holistic meta-model, the dashboard meta-model was transformed in an instance of Ecore using Graphical Modelling for Ecore included in EMF.

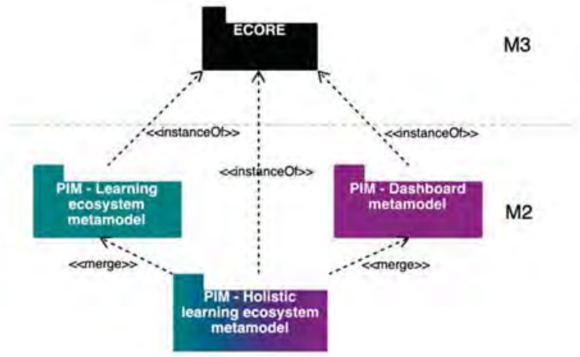


Figure 1. Meta-models organized in the four-layer metamodel architecture.

III. Learning Ecosystem Meta-model

There are a large number of open source tools that allow knowledge management in different ways, with particular emphasis on content managers and document repositories. On the other hand, from the point of view of learning management, there is a wide variety of learning platforms (Learning Management System, or LMS [19]) and tools that allow the definition of Personal Learning Environments (PLE) [20, 21]. Technological ecosystems for learning must be able to combine some of these tools to support knowledge and learning processes in heterogeneous contexts, from institutional environments to private enterprises. Besides, they must be able to incorporate emerging tools, as well as to remove those that become obsolete or that users do not use, in such a way that the system must be in continuous evolution.

Despite the advantages offered by technological ecosystems, the development of such solutions is more complex than traditional information systems. The definition of a particular ecosystem requires knowledge and selection of appropriate systems and services to meet the needs of a particular context. Likewise, the ecosystem should be prepared to evolve and adapt to the changing needs of the environment and users; meanwhile, interoperability between the different components must ensure a high degree of integration and cohesion [9].

The learning ecosystem meta-model is proposed as a solution to improve the processes of definition and development of technological ecosystems, in order to solve the different challenges and problems identified through the analysis of a set of learning ecosystems deployed in different contexts, and with very diverse objectives.

In particular, the meta-model [22] is a Platform-Independent Model (PIM) to define learning ecosystems based on Open Source software (https://doi.org/10.5281/zenodo.1066369) [9]. It is an instance of Ecore with a set of constraints defined with Object Constraint Language (OCL).

The meta-model represents the three main elements of a learning ecosystem. First, the different software tools that compose the ecosystem: data repositories, monitoring tools, user management systems, indexing services, decision-making tools, etc.

Second, the proposed meta-model includes the human factor at the same level as the software because these human resources (management definition, methodologies, and users) are the key elements to ensure the evolution of the ecosystem.

Finally, the third element is the information flows used to support the interaction between the other elements in the ecosystem. The interaction between the software tools is implemented by services and properties files.

On the other hand, the interaction between software tools and users has a substantial impact on the ecosystem; for this reason, these interactions are also represented through information flows. All information flows are based on the objectives defined by management.

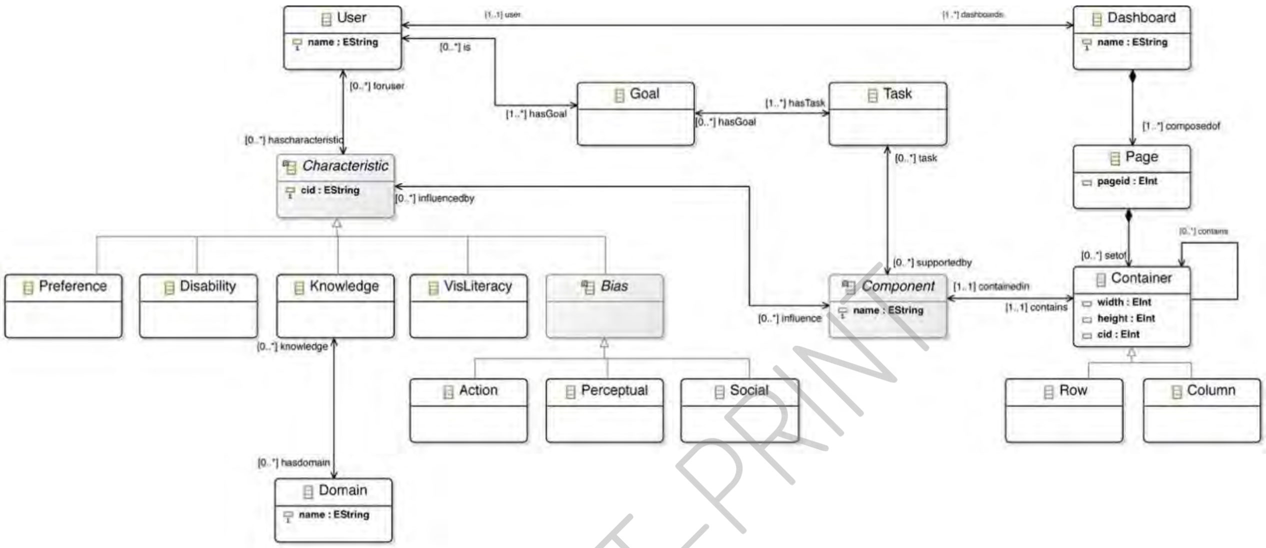


Figure 2. User, layout and components section of the dashboard meta-model proposal.

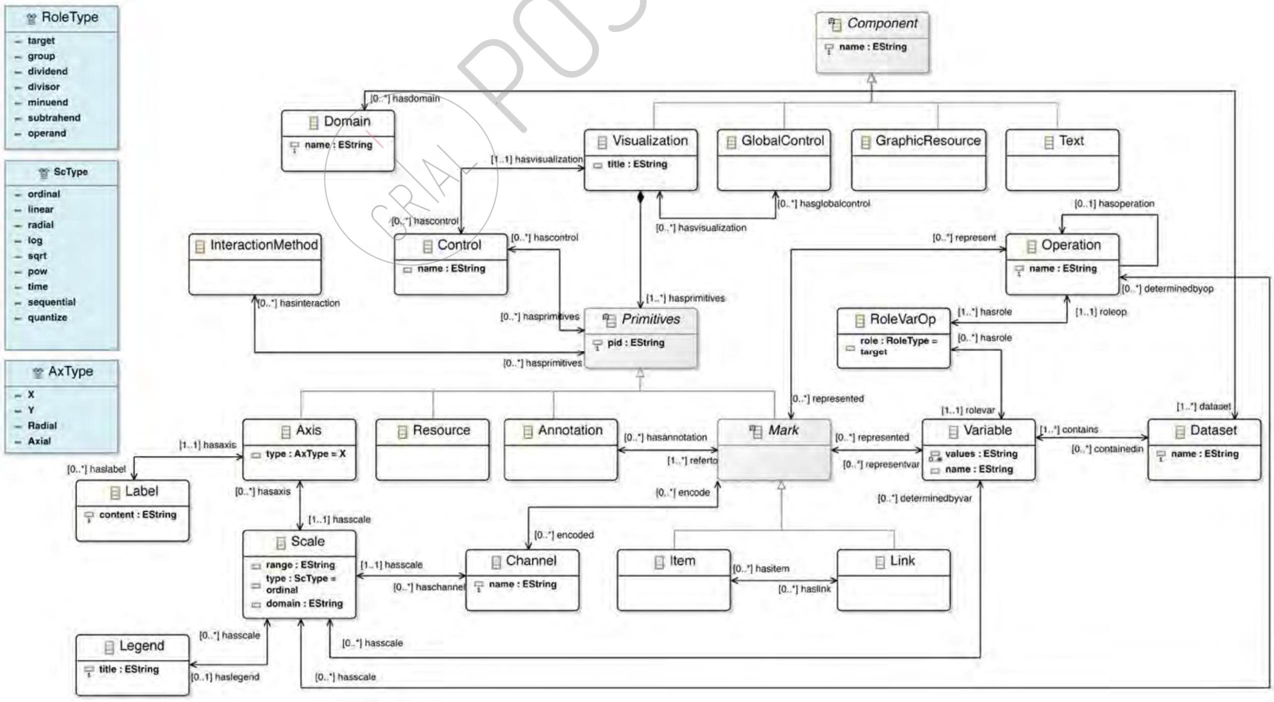


Figure 3. Detailed view of the components section of the dashboard meta-model proposal.

IV. DASHBOARD ECOSYSTEM META-MODEL

The dashboard meta-model is also composed by a variety of pieces that allow the definition of different types of dashboard and information visualizations. The dashboard meta-model can be divided into three main sections: the user, the layout and the components.

Figure 2 shows an excerpt of the dashboard meta-model containing the mentioned three sections. A detailed view of the components section can be consulted in Figure 3. The layout and the components are more technical aspects of dashboards, as they will compose the final display.

However, how many visualizations will the dashboard hold? How these views will be arranged? What type of visualizations will the dashboard display? What type of interaction methods will the dashboard support? Will the different views be linked? These questions cannot be answered in an arbitrary manner. These are design decisions, and they need to be driven by the final consumers of dashboards: the users.

Including the user in this meta-model is essential, because they will be using the delivered dashboards to reach insights, to support their decision-making processes or to explore certain datasets. The user is defined in terms of significant and influential aspects to support a personalized dashboard design, i.e., the factors that influence the design process of a dashboard [12]. Given that, the user entity is decomposed in terms of his or her goals and his or her characteristics.

Firstly, a crucial concept arises; Goal. A user must have at least one goal for using a dashboard, however implicit. Goals, in turn, can be broken down into individual and more specific, low-level tasks. Simple goals can be accomplished by performing a few tasks. However, more elaborated goals might involve several specific and chained tasks, which could involve different data dimensions to support complex decision-making processes [13, 14, 23].

Finally, a user can have a set of identified characteristics. Characteristics can be diverse. For example, preferences, disabilities, knowledge about different domains, visualization literacy, and bias (action, perceptual, or social bias) are different kind of characteristics. These characteristics can influence the design process, thus needing to adapt the dashboard's components to match the identified user aspects.

In terms of the components of the dashboard, several elements are identified. The main components of dashboards are the information visualizations that display data, but also the controls (handlers, filters, and so on), graphic resources, or text that complement these visualizations (Figure 3).

Information visualizations, in turn, are composed of primitives, which can be different visual marks that encode data variables through channels (i.e., color, size, position, etc.). These primitives are the core of information visualizations, because they are the elements that hold the actual data [24, 25].

The dashboard metamodel was an instance of MOF, but it was transformed into an instance of Ecore using Graphical Modelling for Ecore to enable its the connection to the learning ecosystem meta-model, as mentioned in the methodology section.

During the transformation, some changes were introduced to compile with the Ecore rules, so the dashboard meta-model described in this section differs in some details from the previous version [26, 27]. The following changes only address modeling issues to enable the instantiation of Ecore:

- Renaming some classes to remove white spaces and introducing CamelCase notation: *VisLiteracy*, *GlobalControl*, *GraphicResource*, and *InteractionMethod.*
- Introduction of id attributes in each class in order to allow the instantiation of the meta-model in a M1-model.
- Introduction of enumeration classes to enclose the values for some attributes.
- Review of the associations' navigability because in Ecore the navigability is always represented.
- Introduction of names for each relationship.
- Translation of the *RoleVarOp* association class into binary associations (association class is not supported by Ecore).
- Transformation of the reflexive composition association of the *Container* class into a reflexive binary association.
- Transformation of the aggregations into binary associations (aggregation is not supported by Ecore).

The mentioned modifications are crucial for the next steps. The introduction of identifiers and the explicit navigability of the relationships are necessary to instantiate the metamodel and to introduce constraints through OCL, for example.

V. A HOLISTIC ECOSYSTEM METAMODEL

As introduced at the beginning, ecosystems are composed by different elements with different functions and goals. However, these elements are more powerful when connected and when they collaborate with each other through information flows. That is why a holistic solution is proposed, in which each part of it is strengthened when collaborating among them. The ecosystem is seen as a whole, and not only as individual parts with no relationships between one another.

Although the learning ecosystem meta-model proposed solve most of the problems associated to the definition and development of theses technological solutions, there are some issues related to the analysis of the information flows and the support to decision-making processes that should be improved.

The dashboard meta-model presented in Figures 3 and 4 is connected to the ecosystem meta-model (available at https://doi.org/10.5281/zenodo.3561320 [28]).

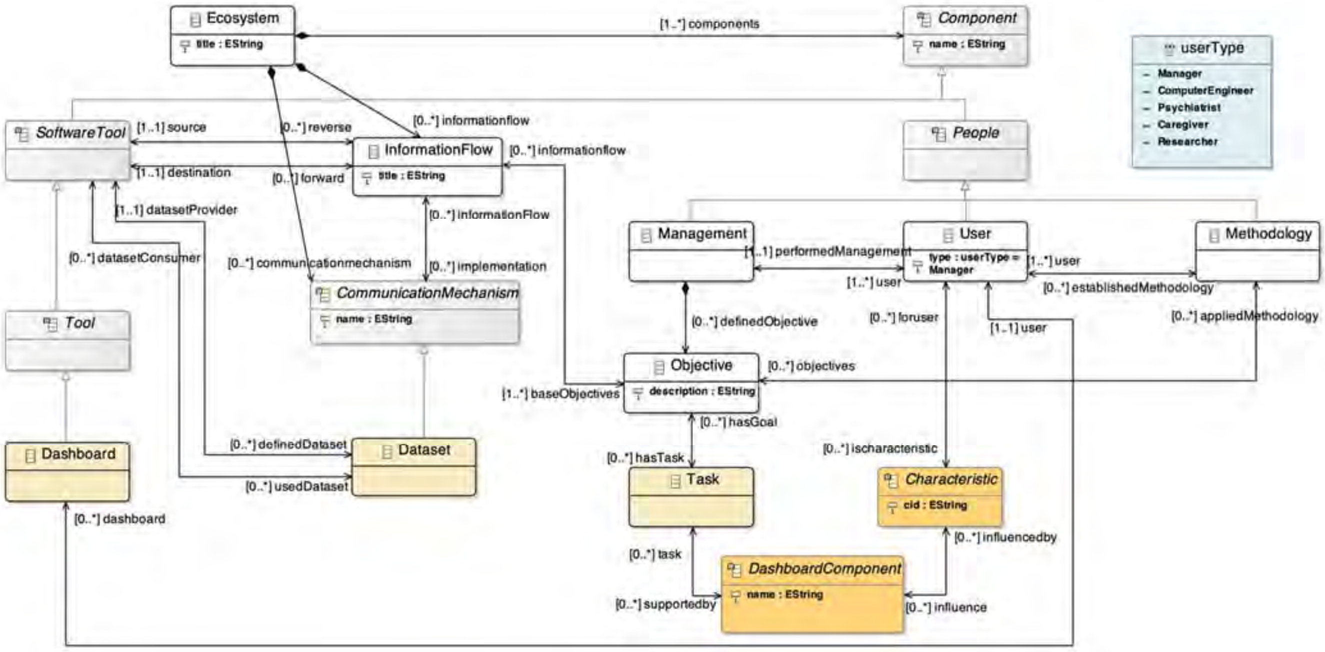


Figure 4. Connection between both meta-models.

The ecosystem meta-model represents each software component as a black box, to provide a high-level abstraction of its structure. The dashboard meta-model provides detail to model these tools as a part of the ecosystem. The complexity of dashboards requires an in-depth analysis of the domain to identify their main commonalities and features, and how these features relate with each other.

Although the dashboard meta-model includes more details regarding these tools' structure and elements, the meta-model is still at a M2-level in terms of the OMG's four-layer architecture. The presented dashboard meta-model is an instance of Ecore, as well as the learning ecosystem meta-model.

These two M2-level meta-models are connected by some elements present both in the dashboard meta-model and the ecosystem meta-model. On the one hand, it has been justified the necessity of including users in the dashboard meta-model because they are the drivers and consumers of the displayed data. The human factor also plays a crucial role in the learning ecosystem meta-model because the technology is defined and evolved to support the users' needs.

On the other hand, there are two relevant elements shared in both meta-models too. The dashboard *Goals* (within the dashboard meta-model) are represented as *Objectives* within the learning ecosystem meta-model. These elements are represented by a set of *Tasks*, and *Information Flows*, respectively. The relevance of these entities is that they are the core of the meta-model, because they frame the required components to achieve the goals or objectives set.

Figure 4 shows the connection between both meta-models. The dashboard *Goal* is merged with *Objective. T*he connection between *Goal* and *User* in the dashboard meta-model is replaced by the association between *User* and *Objective* through the *Management*. In this sense, all the goals that support the definition of the dashboard are connected to the management decisions in the ecosystem.

Regarding the *Dashboard*, the main class to instantiate the dashboard meta-model is connected with the learning ecosystem as a *Tool*. Besides, the connection between *User* and *Dashboard*, which has a particular impact on the dashboard meta-model, is included in the proposal. The information flows and tasks are different concepts, so it is not possible to merge them. For this reason, the *Task* entity is included in Figure 4.

On the other hand, a new communication mechanism is included to implement the information flows, the *Dataset*, as a way to represent the integration between the dashboard and other software tools in the learning ecosystem. Also, the dashboard *Component* has been renamed as *DashboardComponent* to distinguish it from the learning ecosystem *Component*.

Finally, the connection between dashboard *Characteristic* and *User* appears in the new proposal. Tasks are supported by the dashboard's elements, that are also influenced by the user characteristics to match his or her information requirements.

VI. Conclusions

An integration of two meta-models has been proposed. Specifically, a dashboard meta-model has been included within a learning ecosystem meta-model to solve some issues related to knowledge discovery and decision-making processes in the learning ecosystems.

The dashboard meta-model provides a skeleton that can be adapted to instantiate concrete dashboard solutions. The role of the dashboard is to support decision-making processes through visual analysis. Including the user within the meta-model is crucial, as their goals and data requirements are the drivers of the dashboard configuration process.

On the other hand, the learning ecosystem meta-model solves several problems related to the definition and implementation of learning ecosystems. Learning ecosystems combine tools to support knowledge management. Including an information dashboard within the learning ecosystem address the improvement of knowledge discovery within the ecosystem, by providing a tool to analyze information flows.

However, although the learning ecosystem is validated and its quality was checked through the framework defined by López-Fernández, et al. [29]. it is necessary to validate and apply the same framework to the meta-model proposed due to the dashboard meta-model is not fully validated in previous works. Furthermore, future research lines will involve the refinement of the meta-model through the addition of constraints, rules, and design guidelines, in addition to the testing of products instantiated from these meta-models.

ACKNOWLEDGMENT

This research was partially funded by the Spanish Government Ministry of Economy and Competitiveness throughout the DEFINES project grant number (TIN2016-80172-R) and the Ministry of Education of the Junta de Castilla y León (Spain) throughout the T-CUIDA project (SA061P17).

This research was supported by the Spanish Ministry of Education, Culture and Sport under a FPU fellowship (FPU17/03276).

REFERENCES

[1] K. Krüger, "El concepto de 'Sociedad del Conocimiento'," *Revista bibliográfica de geografía y ciencias sociales,* vol. XI, no. 683, 2006.

[2] UNESCO, *UNESCO World Report: Towards Knowledge Societies*. Paris: UNESCO Publishing, 2005.

[3] M. Heidenreich, "Die Debatte um die Wissensgesellschaft," in *Wissenschaft in der Wissensgesellschaft*, S. Böschen and I. Schulz-Schaeffer, Eds. Wiesbaden: VS Verlag für Sozialwissenschaften, 2003.

[4] I. Nonaka and H. Takeuchi, *The knowledge-creating company: How Japanese companies create the dynamics of innovation*. Oxford, UK: Oxford university press, 1995.

[5] A. Hargreaves, *Teaching in the knowledge society: Education in the age of insecurity*. Teachers College Press, 2003.

[6] K. Manikas and K. M. Hansen, "Software ecosystems – A systematic literature review," *Journal of Systems and Software,* vol. 86, no. 5, pp. 1294-1306, 05/05/2013 2013.

[7] T. Weiblen, A. Giessmann, A. Bonakdar, and U. Eisert, "Leveraging the software ecosystem:Towards a business model framework for marketplaces," in *Proceedings of the International Conference on Data Communication Networking, e-Business and Optical Communication Systems*: SciTePress, 2012, pp. 187-193.

[8] F. Llorens-Largo, R. Molina, P. Compañ, and R. Satorre, "Technological Ecosystem for Open education," in *Smart Digital Futures 2014*, vol. 262, R. Neves-Silva, G. A. Tsihrintzis, V. Uskov, R. J. Howlett, and L. C. Jain, Eds.: IOS Press, 2014, pp. 706-715.

[9] A. García-Holgado, "Análisis de integración de soluciones basadas en software como servicio para la implantación de ecosistemas tecnológicos educativos," PhD, Programa de Doctorado en Formación en la Sociedad del Conocimiento, University of Salamanca, Salamanca, Spain, 2018.

[10] A. García-Holgado and F. J. García-Peñalvo, "Validation of the learning ecosystem metamodel using transformation rules," *Future Generation Computer Systems,* vol. 91, pp. 300-310, 2019.

[11] A. Sarikaya, M. Correll, L. Bartram, M. Tory, and D. Fisher, "What Do We Talk About When We Talk About Dashboards?," *IEEE Transactions on Visualization Computer Graphics,* vol. 25, no. 1, pp. 682 - 692, 2018.

[12] A. Vázquez-Ingelmo, F. J. García-Peñalvo, and R. Therón, "Information Dashboards and Tailoring Capabilities - A Systematic Literature Review," *IEEE Access,* vol. 7, pp. 109673-109688, 2019.

[13] R. Amar, J. Eagan, and J. Stasko, "Low-level components of analytic activity in information visualization," in *IEEE Symposium on Information Visualization, 2005. INFOVIS 2005.*, 2005, pp. 111-117: IEEE.

[14] M. Brehmer and T. Munzner, "A multi-level typology of abstract visualization tasks," *IEEE transactions on visualization and computer graphics,* vol. 19, no. 12, pp. 2376-2385, 2013.

[15] D. Patil and H. Mason, *Data Driven*. " O'Reilly Media, Inc.", 2015.

[16] A. Pleuss, S. Wollny, and G. Botterweck, "Model-driven development and evolution of customized user interfaces," in *Proceedings of the 5th ACM SIGCHI symposium on Engineering interactive computing systems*, 2013, pp. 13-22: ACM.

[17] S. J. Mellor, K. Scott, A. Uhl, and D. Weise, "Model-Driven Architecture," in *Advances in Object-Oriented Information Systems: OOIS 2002 Workshops Montpellier, France, September 2, 2002 Proceedings*, J.-M. Bruel and Z. Bellahsene, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2002, pp. 290-297.

[18] J. M. Álvarez, A. Evans, and P. Sammut, "Mapping between Levels in the Metamodel Architecture," in *《UML》 2001 — The Unified Modeling Language. Modeling Languages, Concepts, and Tools. UML 2001. Lecture Notes in Computer Science*, vol. 2185, M. Gogolla and C. Kobryn, Eds. Berlin, Heidelberg: Springer, 2001, pp. 34-46.

[19] F. J. García-Peñalvo and B. Gros, "Future trends in the design strategies and technological affordances of e-learning," in *Learning, Design, and Technology. An International Compendium of Theory, Research, Practice, and Policy*, M. Spector, B. B. Lockee, and M. D. Childress, Eds. Switzerland: Springer International Publishing, 2016, pp. 1-23.

[20] S. Wilson, O. Liber, M. Johnson, P. Beauvoir, P. Sharples, and C. Milligan, "Personal Learning Environments: Challenging the dominant design of educational systems," *Journal of e-Learning and Knowledge Society,* vol. 3, no. 3, pp. 27-38, 2007.

[21] G. Attwell, "The personal learning environments–The future of eLearning?," *eLearning Papers,* vol. 2, no. 1, 2007.

[22] A. García-Holgado and F. J. García-Peñalvo, "A metamodel proposal for developing learning ecosystems," in *Learning and Collaboration Technologies. Novel Learning Ecosystems. 4th International Conference, LCT 2017. Held as Part of HCI International 2017, Vancouver, BC, Canada, July 9–14, 2017. Proceedings, Part I*, P. Zaphiris and A. Ioannou, Eds. (Lecture Notes in Computer Science, no. 10295) Switzerland: Springer International Publishing, 2017, pp. 100-109.

[23] H. Lam, M. Tory, and T. Munzner, "Bridging from goals to tasks with design study analysis reports," *IEEE transactions on visualization and computer graphics,* vol. 24, no. 1, pp. 435-445, 2017.

[24] T. Munzner, *Visualization analysis and design*. Boca Raton, FL, USA: AK Peters/CRC Press, 2014.

[25] C. Ware, *Information visualization: perception for design*. Elsevier, 2012.

[26] A. Vázquez-Ingelmo, F. J. García-Peñalvo, R. Therón, and M. Á. Conde González, "Extending a dashboard meta-model to account for users' characteristics and goals for enhancing personalization," presented at the Learning Analytics Summer Institute (LASI) Spain 2019, Vigo, Spain, 2019.

[27] A. Vázquez-Ingelmo, F. J. García-Peñalvo, and R. Therón, "Capturing high-level requirements of information dashboards' components through meta-modeling," presented at the 7th International Conference on Technological Ecosystems for Enhancing Multiculturality (TEEM 2019), León, Spain, 2019.

[28] A. Vázquez-Ingelmo, A. García-Holgado, F. J. García-Peñalvo, and R. Therón. (2019). *Ecore version of the metamodel for information dashboards*. Available: https://doi.org/10.5281/zenodo.3561320

[29] J. J. López-Fernández, E. Guerra, and J. de Lara, "Assessing the Quality of Meta-models," in *MoDeVVa*, vol. 1235, F. Boulanger, M. Famelis, and D. Ratiu, Eds. Valencia, Spain: CEUR Workshop Proceedings, 2014, pp. 3-22.

7.19 Appendix S. Generating Dashboards Using Fine-Grained Components: A Case Study for a PhD Programme

Vázquez-Ingelmo, A., García-Peñalvo, F. J., & Therón, R. (2020). Generating Dashboards Using Fine-Grained Components: A Case Study for a PhD Programmes. In P. Zaphiris & A. Ioannou (Eds.), Learning and Collaboration Technologies. Design, Experiences. 7th International Conference, LCT 2020, Held as Part of the 22nd HCI International Conference, HCII 2020, Copenhagen, Denmark, July 19–24, 2020, Proceedings, Part I (pp. 303-314). Springer Nature. https://doi.org/10.1007/978-3-030-50513-4_23

Generating Dashboards Using Fine-Grained Components: A Case Study for a PhD Programme

Andrea Vázquez-Ingelmo[1(✉)], Francisco J. García-Peñalvo[1], and Roberto Therón[1,2]

[1] GRIAL Research Group, Computer Sciences Department, Research Institute for Educational Sciences, University of Salamanca, Salamanca, Spain
{andreavazquez, fgarcia, theron}@usal.es
[2] VisUSAL Research Group, University of Salamanca, Salamanca, Spain

Abstract. Developing dashboards is a complex domain, especially when several stakeholders are involved; while some users could demand certain indicators, other users could demand specific visualizations or design features. Creating individual dashboards for each potential need would consume several resources and time, being an unfeasible approach. Also, user requirements must be thoroughly analyzed to understand their goals regarding the data to be explored, and other characteristics that could affect their user experience. All these necessities ask for a paradigm to foster reusability not only at development level but also at knowledge level. Some methodologies, like the Software Product Line paradigm, leverage domain knowledge and apply it to create a series of assets that can be composed, parameterized, or combined to obtain fully functional systems. This work presents an application of the SPL paradigm to the domain of information dashboards, with the goal of reducing their development time and increasing their effectiveness and user experience. Different dashboard configurations have been suggested to test the proposed approach in the context of the Education in the Knowledge Society PhD programme of the University of Salamanca.

Keywords: Domain engineering · SPL · Information dashboards · Information systems · Educational dashboards

1 Introduction

Data visualization is gaining relevance as a method to understand and generate knowledge [1] from datasets. However, the exponential growth in data generation due to the widespread of data-driven technologies [2] asks for new methodologies to support informed decision-making even if relying on complex and large datasets.

Visual analytics [3, 4] is a popular methodology to foster the comprehension of large quantities of data. This field focuses on analytical reasoning and how interactive tools and visual interfaces can support it.

Visualizing data and generating knowledge from them through information visualizations or information dashboards [5–7] require proper representations, that is, appropriate encodings and visual metaphors to convey the information contained (and mainly hidden) within large datasets.

However, although practitioners can follow general guidelines to design information visualizations [8], the process of building these tools is costly, because it is important not only to take into account but also to deeply understand the context in which these tools will be employed.

The audience and the context of application are essential to build an effective dashboard. One of the challenges in this field is the adaptation of visualizations and dashboards based on their context [9]. But the adaptation of these tools is complex; it requires several resources to design and implement an adapted version of the same tool.

That is why some approaches that leverage reusability are extremely useful in these domains [10]. Especially, approaches like the Software Product Line (SPL) [11, 12] paradigm and the Model-Driven Development (MDD) [13] foster the reusability of components and knowledge by abstracting common features of a concrete domain.

A dashboard meta-model has been previously developed [14–16] to account for not only for technical features of these tools but also to account for their users. Involving the end-users' characteristics and goals regarding information is crucial to obtain a dashboard adapted to the context and audience.

This work focuses on the application of the developed dashboard meta-model to generate information dashboards in the context of a PhD programme. The goal of this application is to test the suitability of a generative pipeline to tailor information dashboards in a real-world scenario.

A requirement elicitation process has been carried to identify the information requirements and goals of each involved user profile to achieve the mentioned goal. With this information, it has been possible to map the goals into concrete dashboard features.

The rest of this paper is organized as follows. The next section contextualizes the PhD Programme on Education in the Knowledge Society. Section 3 explains the different materials and methods employed to carry out this work. Section 4 presents the results of the requirement elicitation process, as well as the results regarding the generation of customized dashboards based on the previously identified requirements. Finally, Sect. 5 discusses the obtained results, concluding with Sect. 6, in which the conclusions derived from this work are described.

2 PhD Programme on Education in the Knowledge Society

The PhD Program on Education in the Knowledge Society emerges from the Research Institute for Educational Sciences (IUCE – http://iuce.usal.es) at the University of Salamanca (Spain), following the Spanish Royal Decree 99/2011. The primary motivation behind this PhD Program is to feature the teaching-learning forms as main impetus of the Knowledge Society, so as to examine and create new information about the learning as a key component of the Knowledge Society, including both the Social Sciences considers and the new mechanical advances yet inside a synergic and harmonious methodology [17, 18].

The focus of this program is completely interdisciplinary, mainly upheld by the Recognized Research Groups at the University of Salamanca: GRIAL (http://grial.usal.es) [19], GITE (http://gite213.usal.es), OCA (http://campus.usal.es/~oca/), VISUALMED (http://visualmed.usal.es), Robotics and Society Group (http://gro.usal.es) y E-LECTRA (http://electra.usal.es).

The PhD Program gives a context where knowledge generation and its visibility and dissemination are primary objectives. In order to reach them, the scientific knowledge management of the Programme is supported by a technological ecosystem that fusion methodology and technology to provide tools for both PhD candidates and researchers. The fundamental parts of the ecosystem are the PhD portal (http://knowledgesociety.usal.es) and a set of social instruments, such as SlideShare to share presentations (http://www.slideshare.net/knowedgesociety) or a YouTube channel to share seminars and meetings (http://youtube.com/knowledgesocietyphd).

3 Materials and Methods

3.1 Dashboard Meta-model

Meta-models are the starting point of the Model-Driven Development paradigm [13]. These artifacts allow the high-level definition of entities and relationships, without focusing on the technical details of the domain.

In this case, a dashboard meta-model has been developed to capture the abstract characteristics that define dashboards. As can be seen in Fig. 1, the user is also part of the meta-model because they are the drivers of the design process and the final consumers of the information displayed through these tools, so their characteristics must be accounted for to deliver a properly designed dashboard.

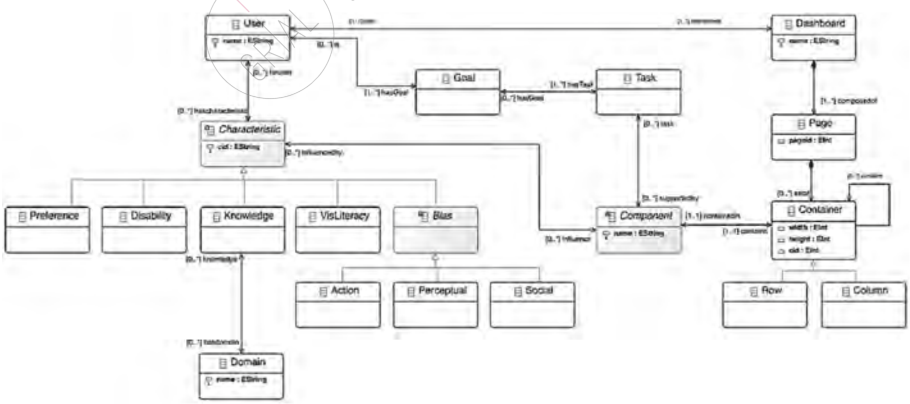


Fig. 1. Overview of the dashboard meta-model. The component definition has been omitted for legibility reasons, but it can be consulted at [20].

The entire meta-model design process is out of the scope of this paper, but it can be consulted in [15, 16].

3.2 Feature Model

The goal of software product lines (SPLs) is to derive final products from different core assets and software components [11, 12], and to allow the adaptation of products to match specific requirements without consuming significant time and resources.

This paradigm has two main phases: the domain engineering phase and the application engineering phase. The mentioned core assets are developed during the first phase by identifying commonalities and variability points within the products' domain. This phase supports the implementation of base components that hold the common logic of the domain as well as the variability points.

Variability points are sections of the logic that can be modified, parameterized, or configured to change the functionality of the products during the second phase (application engineering phase). In SPLs, products' functionalities are seen as features. Stakeholders can select different features for their products to be injected in the base logic, thus obtaining personalized systems that fit their specific requirements. These personalized products are built by reusing and assembling the core assets developed during the domain engineering phase, which reduces the time-to-market of tailored software systems as well as their development efforts.

The features of software product lines can be specified by different means. One of the most popular methods to identify and arrange SPL features are feature models [21]. Feature models are useful for documentation purposes, but also essential artifacts for guiding the development process of the product line. These models provide a skeleton for designing the core assets and for materializing the variability points at code level.

In this specific domain (i.e., the dashboards domain), the feature model captures the dashboards' visualizations' low-level characteristics, corresponding with visual encodings, visual marks, etc. [8]. Figure 2 shows an excerpt from the feature model employed to design the core assets for the dashboards product line.

The hierarchical structure of the feature model enables the definition of high-level characteristics and their refinement to reach finer-grained features [22].

3.3 Requirements Elicitation

The requirements elicitation phase is an essential phase in any software development process. In this case, it is crucial to understand the context of application (the PhD Programme on Education in the Knowledge Society) to generate adapted information dashboards.

Firstly, there is a variety of profiles to take into account involved in the programme: from PhD candidates to their advisors and managers. These users will be the drivers of the dashboards' design and generation processes because the outcome must match their information needs and functional requirements.

A questionnaire is proposed to gather the information needs of each involved profile. However, to ensure the proper design of this questionnaire, an initial interview was conducted. The goal of this interview was to collect information regarding the business processes of the PhD programme, as well as the available data and their structure.

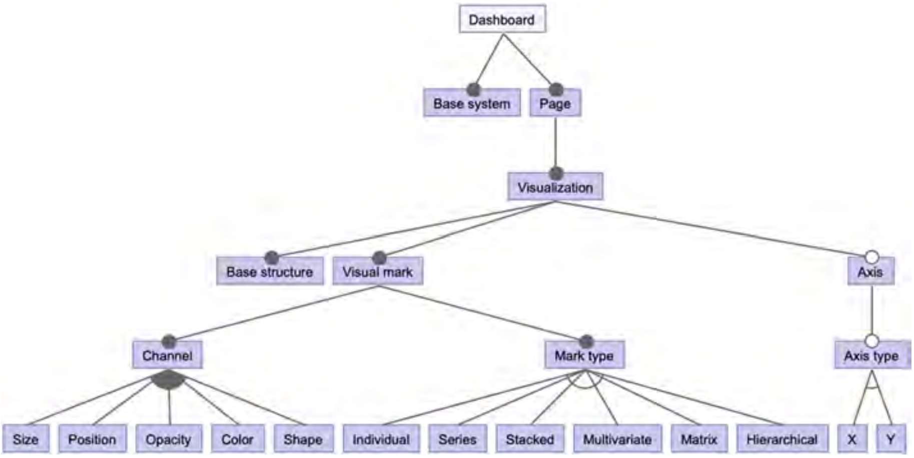


Fig. 2. Simplified feature model of a dashboards product line.

A few questions were asked to a member of the PhD programme's quality committee to understand which data could be displayed in a potential information dashboard. The PhD portal gathers information about the milestones achieved by the members of the PhD programme. For example, data about publications, research visits, conferences, reports, awards, seminars, patents, etc., are collected in a semi-structured way.

On the other hand, the PhD programme on Education in the Knowledge Society has a series of minimum requirements to be able to defend the thesis or to obtain some recognitions like an international PhD mention. These requirements are covered in [23].

In addition, other aspects of the PhD candidates are stored, such as their academic year, associated research lines, PhD advisors, scholarships, contracts, PhD modality (full- or part-time), etc.

Finally, the users of the portal can have different roles: students (including current PhD candidates or post-doctoral students), PhD advisors, and managers.

This interview supported not only the development of the questionnaire, but also the comprehension of the variables available to be displayed in potential PhD information dashboards.

3.4 Instrumentation

Once the interview was carried out, an instrument to collect information regarding the users' information requirements was designed. First, some demographic variables are collected to contextualize the sample: age, gender, and birthplace.

The next section focuses on the collection of the user situation within the PhD programme: role, research lines, PhD modality, and academic year (in the case of PhD students) and the number of PhD thesis being directed (in the case of PhD advisors). Questions regarding the usage of the PhD portal were also included in this section to understand how users employ this platform.

Finally, the last section included questions regarding users' past experiences with information visualization and regarding the users' information requirements for a hypothetical PhD programme dashboard.

The instrument to collect these requirements was implemented employing a customized version of LimeSurvey (https://www.limesurvey.org), an Open Source online statistical survey web application. The instrument was applied in Spanish because the PhD portal users were Spanish speakers.

4 Results

4.1 Requirements Elicitation Results

The questionnaire was answered by 24 participants. The distribution of the participants in terms of their role is summarized in Fig. 2. All the participants stated that they employ the PhD portal, and the majority had experience using dashboards or visualization tools (70.83%).

Regarding how the PhD students save their milestones, an interesting pattern of answers were found. The information regarding their milestones is scattered among different research profiles (Mendeley, Scopus, WoS, Google Scholar, etc.) and/or the PhD programme portal, in personal Microsoft Word files created by themselves and even some students state that they don't save their milestones at all. These aspects are summarized in Fig. 3.

Finally, the collected requirements for a hypothetical personal dashboard within the PhD programme portal suggest that users have different information goals. On the one hand, PhD advisors were very interested in obtaining information regarding the publications (count, typology, and the number of citations), conference attendance/participation, and research profiles of their PhD students (Fig. 4).

In relation to managers or academic committee members, in addition to the information related to the progress of all the PhD students and their deadlines, some pointed out as an information requirement the number of PhD students associated to each research group, as well as metrics regarding the interactions between their PhD advisors. Another participant asked for information about publications by research line and the number of publications during specific periods of time.

As will be discussed, the most diverse requirements were found within the PhD students' answers. The majority share the necessity of displaying information related to their PhD progress and their achieved milestones (publications, conferences, seminars, etc.), but other requirements were also mentioned:

- Remaining required activities.
- Recommended activities vs. required activities.
- Comparison with other PhD students.
- Distribution of activities/milestones (publications, seminars, etc.) by type.
- Percentage of progress based on the requirements of the PhD programme.
- Status of each research milestone uploaded to the portal.
- Deadlines and enrollment dates.

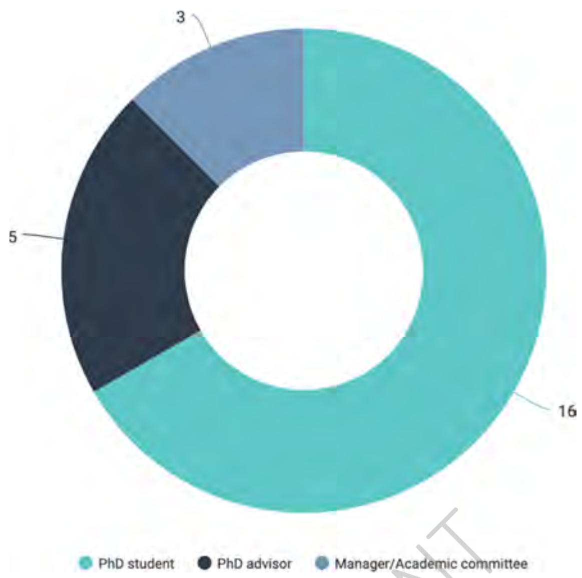


Fig. 3. Distribution of participants regarding their role in the PhD programme (n = 24).

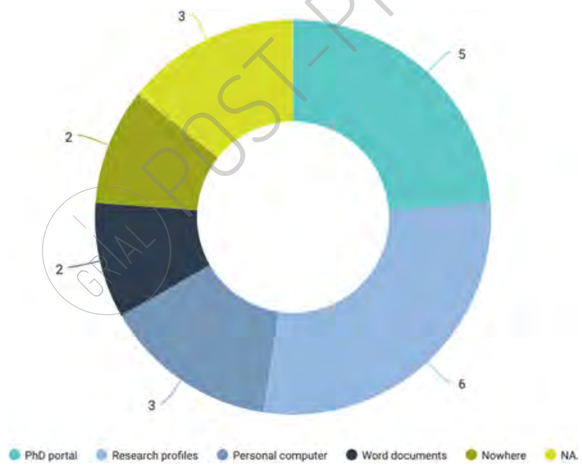


Fig. 4. Distribution of the methods employed by PhD students to save their progress/milestones obtained within the PhD programme (n = 16; some participants pointed out more than one method to save their milestones).

4.2 Dashboard Generation Results

Based on some of the requirements identified through the questionnaire, three dashboard prototypes for each major PhD role have been generated by using the software product line paradigm. These proposals can be seen in Figs. 5, 6, and 7.

For example, based on the collected requirements, a student dashboard might be composed of views that show their achievements classified by type and reference marks to show if they have reached the minimum requirements of the PhD programme. On the other hand, these achievements can also be displayed through time, allowing students to inspect their productivity (Fig. 5).

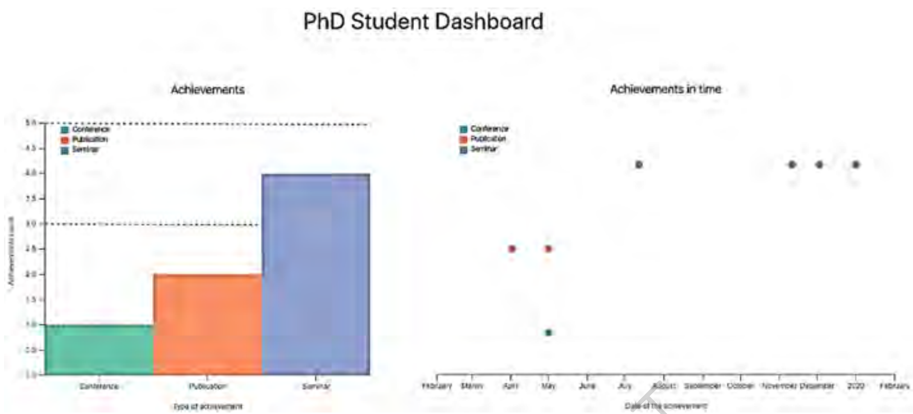


Fig. 5. PhD student dashboard proposal.

On the other hand, the PhD advisor dashboard (Fig. 6) could show each student individual progress as well as a general overview through the time of the advisors' PhD candidates. The flexibility of this approach allow the adaptation of this view to the specific number of students of each PhD advisor.

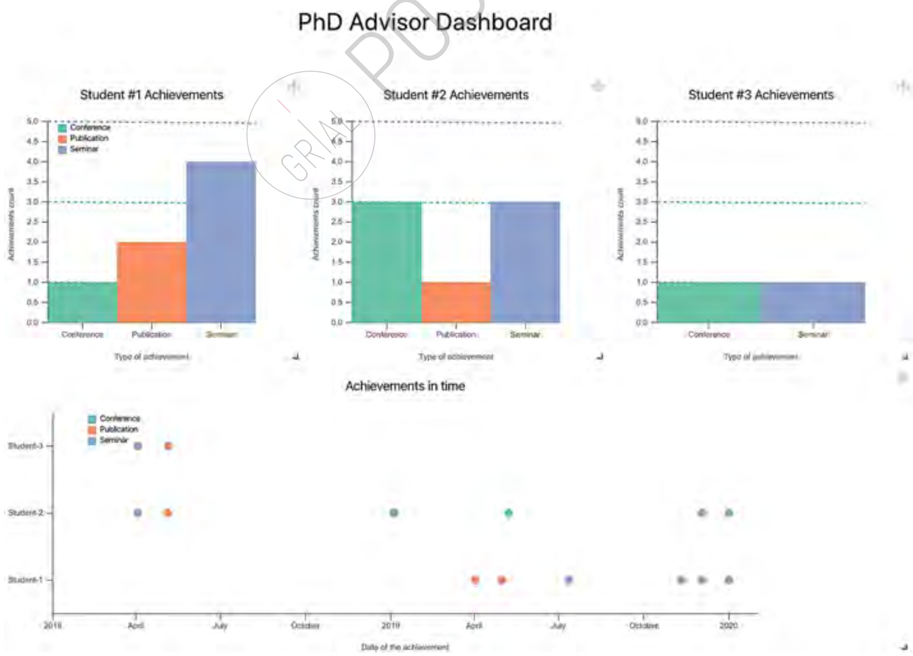


Fig. 6. PhD advisor dashboard proposal.

Finally, the PhD manager dashboard (Fig. 7) could hold information regarding the workload (in terms of directed thesis) of each PhD advisor and research group involved in the programme. Also, following the collected requirements, this dashboard can also have a view displaying the achieved publications through time by research line.

Fig. 7. PhD manager dashboard proposal.

5 Discussion

The questionnaire yielded very interesting results and confirmed that dashboards must take into account different user roles and user requirements. There is not a “universal” dashboard that fits for any user; that is why dashboards should be tailored depending on their context.

Three main roles have been identified within the PhD programme through an interview with a member of the quality committee: PhD students, PhD advisors, and the programme managers. The questionnaire asked different questions depending on the role of the participant. While students asked for information about their progress or publications, PhD advisors wanted to gain insights regarding their own students’ process. On the other hand, managers mostly asked to gain overall insights regarding all the students enrolled in the programme.

The greatest diversity of requirements was found among PhD students; although these users share the same role, significant differences were encountered regarding their information goals. In this work, a single dashboard proposal was presented for PhD students, but different dashboards could be developed for this role; for example, one user asked for comparisons with other students’ progress.

While this could be a beneficial feature for this user, it could be counterproductive for other users (for example, showing their progress compared with other pre-doctoral users could have a negative impact on some students’ motivation [24]).

The three dashboards were generated using a DSL based on the feature tree and meta-model presented in Sect. 3. This approach not only improved the development time of the dashboards, as the tools were automatically generated through code templates (the core assets of the SPL, in this case) but also improved the requirements management of the users.

A requirement file using the DSL could be maintained for each user, supporting fine-grained management of requirements and controlling changes in a straightforward manner.

The main effort, however, is still in the requirements' elicitation process. If users' information requirements are taken for granted or are given low relevance, the approach would be useless, because the output would be an inefficient and ineffective dashboard. That is why current research is focused on how to automatize or infer users' requirements from their characteristics or from the data to be displayed [25, 26].

This approach is yet to be integrated and tested within the Education in the Knowledge Society PhD programme context, but the viability results of using this approach to generate information dashboards seem promising.

6 Conclusions

The SPL paradigm has been applied to generate dashboards in an educative context; specifically, in a PhD programme context.

The diversity of information goals and requirements ask for flexible design and development process to build these tools in order to accelerate the delivery time and leverage the information displayed.

The application of the SPL paradigm requires initial efforts to identify the domain's abstract features and to develop the different core assets, but relying on these artifacts have subsequently reduced the required time to develop tailored dashboards.

The presented dashboard generative process has been driven by the PhD programme involved actors and their information goals, which were gathered through an online questionnaire.

Future research lines will involve the improvement of the dashboard product line to automatize the generation process, as well as in-depth user testing to validate the utility of this approach to integrate tailored dashboards in learning contexts.

Acknowledgments. This research work has been supported by the Spanish *Ministry of Education and Vocational Training* under an FPU fellowship (FPU17/03276). This work has been partially funded by the Spanish Government Ministry of Economy and Competitiveness throughout the DEFINES project (Ref. TIN2016-80172-R).

References

1. Zeleny, M.: Management support systems: Towards integrated knowledge management. Hum. Syst. Manag. **7**, 59–70 (1987)
2. Patil, D., Mason, H.: Data Driven. O'Reilly Media Inc., Sebastopol (2015)
3. Keim, D., Andrienko, G., Fekete, J.-D., Görg, C., Kohlhammer, J., Melançon, G.: Visual analytics: definition, process, and challenges. In: Kerren, A., Stasko, J.T., Fekete, J.-D., North, C. (eds.) Information Visualization. LNCS, vol. 4950, pp. 154–175. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-70956-5_7

4. Keim, D., Kohlhammer, J., Ellis, G., Mansmann, F.: Mastering the Information Age Solving Problems with Visual Analytics. Eurographics Association, Goslar (2010)
5. Tufte, E., Graves-Morris, P.: The Visual Display of Quantitative Information. Graphics Press, Cheshire (1983, 2014)
6. Ware, C.: Information Visualization: Perception for Design. Elsevier, Amsterdam (2012)
7. Few, S.: Information Dashboard Design. O'Reilly Media Inc., Sebastopol (2006)
8. Munzner, T.: Visualization Analysis and Design. AK Peters/CRC Press, Boca Raton (2014)
9. Sarikaya, A., Correll, M., Bartram, L., Tory, M., Fisher, D.: What do we talk about when we talk about dashboards? IEEE Trans. Vis. Comput. Graph. **25**, 682–692 (2018)
10. Vázquez-Ingelmo, A., García-Peñalvo, F.J., Therón, R.: Information dashboards and tailoring capabilities - a systematic literature review. IEEE Access **7**, 109673–109688 (2019)
11. Clements, P., Northrop, L.: Software Product Lines. Addison-Wesley, Boston (2002)
12. Gomaa, H.: Designing Software Product Lines with UML: From Use Cases to Pattern-Based Software Architectures. Addison Wesley Longman Publishing Co., Inc., Boston (2004)
13. Kleppe, A.G., Warmer, J., Bast, W.: MDA Explained. The Model Driven Architecture: Practice and Promise. Addison-Wesley Longman Publishing Co., Inc., Boston (2003)
14. Vázquez-Ingelmo, A., García-Holgado, A., García-Peñalvo, F.J., Therón, R.: Dashboard meta-model for knowledge management in technological ecosystem: a case study in healthcare. In: UCAmI 2019. MDPI (2019)
15. Vázquez-Ingelmo, A., García-Peñalvo, F.J., Therón, R.: Capturing high-level requirements of information dashboards' components through meta-modeling. In: 7th International Conference on Technological Ecosystems for Enhancing Multiculturality (TEEM 2019), León, Spain (2019)
16. Vázquez-Ingelmo, A., García-Peñalvo, F.J., Therón, R., Conde González, M.Á.: Extending a dashboard meta-model to account for users' characteristics and goals for enhancing personalization. In: Learning Analytics Summer Institute (LASI) Spain 2019, Vigo, Spain (2019)
17. García-Holgado, A., García-Peñalvo, F.J., Rodríguez Conde, M.J.: Definition of a technological ecosystem for scientific knowledge management in a PhD programme. In: Proceedings of the Third International Conference on Technological Ecosystems for Enhancing Multiculturality (TEEM 2015), Porto, Portugal, 7–9 October 2015, pp. 695–700. ACM, New York (2015)
18. García-Peñalvo, F.J.: Engineering contributions to a knowledge society multicultural perspective. IEEE Revista Iberoamericana de Tecnologías del Aprendizaje (IEEE RITA) **10**, 17–18 (2015)
19. García-Peñalvo, F.J., Rodríguez-Conde, M.J., Therón, R., García-Holgado, A., Martínez-Abad, F., Benito-Santos, A.: Grupo GRIAL. IE Comunicaciones. Revista Iberoamericana de Informática Educativa, 33–48 (2019)
20. Vázquez-Ingelmo, A., García-Holgado, A., García-Peñalvo, F.J., Therón, R.: Ecore version of the metamodel for information dashboards (2019). https://doi.org/10.5281/zenodo.3561320
21. Kang, K.C., Cohen, S.G., Hess, J.A., Novak, W.E., Peterson, A.S.: Feature-oriented domain analysis (FODA) feasibility study. Carnegie-Mellon University, Software Engineering Institute (1990)
22. Vázquez-Ingelmo, A., García-Peñalvo, F.J., Therón, R.: Addressing fine-grained variability in user-centered software product lines: a case study on dashboards. In: Rocha, Á., Adeli, H., Reis, L.P., Costanzo, S. (eds.) WorldCIST'19 2019. AISC, vol. 930, pp. 855–864. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-16181-1_80
23. Programa de Doctorado Formación en la Sociedad del Conocimiento. https://www.slideshare.net/knowedgesociety/kickoff-phd-programme-20192020

24. Teasley, S.D.: Student facing dashboards: One size fits all? Technol. Knowl. Learn. **22**, 377–384 (2017). https://doi.org/10.1007/s10758-017-9314-3
25. Hu, K., Bakker, M.A., Li, S., Kraska, T., Hidalgo, C.: VizML: a machine learning approach to visualization recommendation. In: Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems, p. 128. ACM (2019)
26. Dibia, V., Demiralp, Ç.: Data2Vis: automatic generation of data visualizations using sequence to sequence recurrent neural networks. IEEE Comput. Graph. Appl. **39**, 33–46 (2019)

7.20 Appendix T. Specifying information dashboards' interactive features through meta-model instantiation

Specifying information dashboards' interactive features through meta-model instantiation

Andrea Vázquez-Ingelmo[1][0000-0002-7284-5593], Francisco J. García-Peñalvo[1][0000-0001-9987-5584], Roberto Therón[1,2][0000-0001-6739-8875] and Alicia García-Holgado[1][0000-0001-5881-7775]

[1] GRIAL Research Group, Computer Sciences Department, Research Institute for Educational Sciences, University of Salamanca, Salamanca, Spain
[2]VisUSAL Research Group. University of Salamanca. Salamanca, Spain

{andreavazquez,fgarcia,theron,aliciagh}@usal.es

Abstract. Information dashboards[1] can be leveraged to make informed decisions with the goal of improving policies, processes, and results in different contexts. However, the design process of these tools can be convoluted, given the variety of profiles that can be involved in decision-making processes. The educative context is one of the contexts that can benefit from the use of information dashboards, but given the diversity of actors within this area (teachers, managers, students, researchers, etc.), it is necessary to take into account different factors to deliver useful and effective tools. This work describes an approach to generate information dashboards with interactivity capabilities in different contexts through meta-modeling. Having the possibility of specifying interaction patterns within the generative workflow makes the personalization process more fine-grained, allowing to match very specific requirements from the user. An example of application within the context of Learning Analytics is presented to demonstrate the viability of this approach.

Keywords: Information Dashboards, Meta-model, Information Visualization, Interactions, MDA, SPL.

1 Introduction

Information dashboards have increased in popularity and relevance in several fields. They foster knowledge generation by presenting complex datasets through visual arrangements and encodings. These tools support informed decision-making processes, and data-driven approaches to carry out complex decision flow [1, 2].

However, carrying out data-driven decision making is not a trivial task. First, because a significant amount of data is needed to generate knowledge. And second, because the processes of analyzing such data require that the person leading the decision making or analysis be able to understand and interpret data sets that are often complex and extensive.

But thanks to the evolution of technologies, these analysis tasks are now available to less technical profiles. There are tools that facilitate analysis and knowledge generation from data sets. One of the most popular tools are dashboards [3].

Having a dashboard does not guarantee knowledge generation, though. It is necessary to take into account the audience and the profile of the users who will use these tools. There may be users who can understand complex visualizations, while others will need other visual metaphors to correctly understand their data sets [4].

Especially, there are some contexts in which putting the focus on the audience is crucial. The educative context is one of them. In this context, there are several actors that play crucial roles: teachers, managers, students, etc.

Learning Analytics (LA) dashboards provide a display in which different indicators regarding learners, learning processes, and/or learning contexts are arranged into a set of visualizations [5]. However, the design process of LA dashboards is crucial to leverage their capabilities [6]. It is necessary to take into account the data, the user, and the goals of the dashboard to get the most out of these tools.

In previous works, a dashboard meta-model has been developed [7-9] to specify and instantiate dashboards within any context. The dashboard meta-model defines high-level classes that depict abstract concepts from the domain, such as the elements that compose information visualizations (channels, visual marks, axes, users' goals, variables, datasets, etc.).

The dashboard meta-model can be derived into concrete models to build specific dashboards through a generator by using a high-level syntax. However, there is a dimension that must also be taken into account: the potential user interactions with the dashboard. Interaction patterns help users to interact directly with the data displayed on their screens. Users could highlight data points, select them, filter them, etc., through different events (clicking, hovering, brushing, etc.). These patterns need to be represented within the meta-model at a high-level to allow their instantiation and, subsequently, to obtain fully interactive dashboards.

This work describes the introduction of interaction patterns within a dashboard meta-model and presents an example of application in the context of LA through a generative pipeline and a DSL (Domain Specific Language).

The rest of this paper is organized as follows. Section 2 provides a background on educational and learning analytics dashboards. Section 3 depicts the methodology employed to carry out the research. Section 4 describes the meta-model structure regarding interaction patterns. Section 5 presents a dashboard instantiation framed in the Learning Analytics context. Finally, sections 6 and 7 discuss the results and conclude the work, respectively.

2 Background

As mentioned in the introduction, dashboards are increasingly popular tools because of their usefulness in supporting the visual analysis of complex datasets. The educational context is one of the contexts in which these tools can bring significant benefits since the use of data to make decisions regarding educational processes can improve learning outcomes [10-12].

Educational dashboards [13] are instruments that allow their users to identify patterns, relationships, relevant data, etc., among a set of learning variables. However, the diversity of roles in this context makes the design of educational dashboards a challenge.

In [5], it was found that the majority of users are usually teachers, but students, administrators, and researchers are also among the main users of these tools. Educational dashboards are also diverse in terms of their objectives; self-monitoring, monitoring of other students, and administrative monitoring [5]. This type of research allows us to observe that dashboards are very diverse in the educational context, both in their functionalities and in their design.

Due to these factors, some methods have been sought to design educational and learning analytics dashboards, so that they can be adapted according to their purposes and audience because there is no one-size-fits-all approach [14].

Dashboards should be customized to provide the necessary information in the most effective way. In fact, a study by Roberts et al. confirmed the widespread desire of students for control panels that can be customized, giving them the option of configuring them to display the information that interests them most or that they find most useful [15].

Thus, it is not only the variety of user roles in the educational context but the variety of objectives and profiles among users with the same role, which makes the development of learning analytics dashboards an elaborate activity. In addition, the amount of data generated and its complex structure can make the process of knowledge discovery even more difficult for less technical profiles.

For these reasons, models have been proposed to try to adapt these tools using conceptual models that take into account the indicators, the description, and needs of the users, their preferences, their knowledge of the domain, etc. In [16], a generator of learning analytics dashboards is presented. This generator takes into account the above-mentioned information by structuring it in models that feed a dashboard generator.

As can be seen, control panels in the educational context have increased in popularity due to the benefits that the use of data can bring to decision making. However, to take advantage of them, it is necessary to take into account the users and the context in which they will be employed.

3 Methodology

The methodology employed relies on two paradigms: model-driven development [17, 18] and the software product line paradigm [19, 20].

Model-driven development leverages high-level models (meta-models) to obtain an abstract representation of systems. Meta-models do not only help in the conceptualization of information systems but are also powerful artifacts that support a whole pipeline for developing such systems. These abstract models can be mapped to concrete products, according to the OMG four-layer meta-model architecture [21]: meta-meta model layer (M3), meta-model layer (M2), user model layer (M1) and user object layer (M0).

Concepts of the domain are captured through meta-models and arranged as a set of classes and relationships, yielding a simplified representation of the problem's domain. This representation is structured, thus supporting the processing of meta-models through computational methods.

In previous works, a dashboard meta-model has been presented [7-9, 22]. The dashboard meta-model includes three main parts involved in the development of information dashboards: the user, the layout, and the components of the dashboard.

This work is focused on addressing the specification and instantiation of interaction patterns among dashboard components to obtain interactive and functional information dashboards. A set of core assets have been developed following the SPL paradigm to support a connection between meta-model instantiations and final products. The SPL provides a framework to create components that can be configured and customized with almost no effort (the main effort is made during the development of these core assets) to meet specific requirements.

4 Modeling interaction patterns

Interaction patterns are highly diverse. They can involve the user clicking some parts of the dashboard. They can also involve hovering, brushing, etc. And, on the other hand, they can provoke different effects, such as highlighting some point, showing a tooltip, filter the data, etc.

All these possibilities must be captured through the meta-model in an abstract and coherent manner. Following a domain engineering approach [23, 24], a set of conceptual classes have been identified across dashboards from different domains to model interaction patterns. These classes are depicted in Fig. 1.

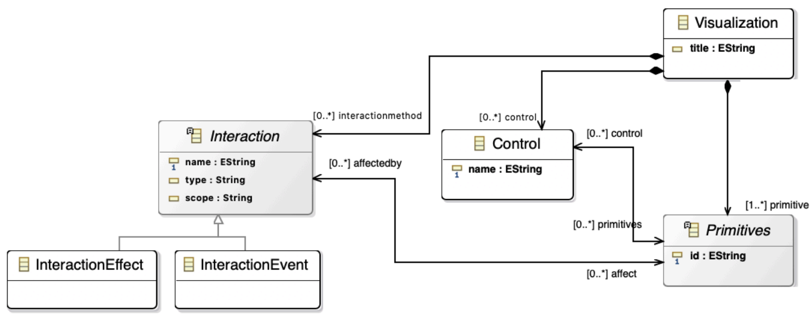


Fig. 1. Meta-model section regarding interaction patterns.

Information visualizations are composed of different elements. Mainly, these visualizations are composed of basic primitives, like visual marks, axes, scales, visual channels, etc. When interacting with a visualization, these primitives will be affected, for example, by changing their colors to highlight them or by showing a tooltip.

Three classes have been identified to reflect interactions in the meta-model. The *Interaction* class, which represents the interaction pattern to be applied to a specific primitive of the visualization. This class is abstract and can be of two types: an event or an effect. This distinction is necessary to represent which events to capture and which effects to apply to the visualization's primitives. For example, clicking in one of the bars from a bar chart is an event, and highlighting that bar (by varying its style) when selected is an effect. With these conceptual classes, it is possible to combine different specifications to obtain fully functional and interactive dashboards.

5 Learning Analytics Dashboard Example

A simple Learning Analytics information dashboard has been instantiated to demonstrate the viability of the generative workflow in this context. To do so, an instance of the Ecore meta-model was developed through EMF (Eclipse Modeling Framework, https://www.eclipse.org/modeling/emf/). This framework provides several features to support model-driven approaches: from meta-model editors to code-generation facilities. In this case, two visualizations are specified: a scatter plot and a parallel coordinates plot (Fig. 2).

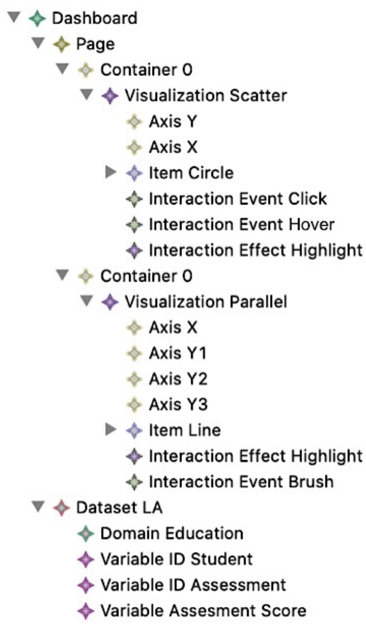


Fig. 2. An excerpt from the meta-model instance.

The elements of these plots encode the values of the input dataset through different channels. These channels are based on scales that map the dataset variables to values that are encoded through the position, color, size, etc. of the visual marks.

The employed dataset to test this application is the Open University Learning Analytics Dataset [25], which contains data from courses presented at the Open University (OU): assessments, scores, students' clickstreams, etc. In this case, the dashboard instance will employ the student ID, assessment ID, and assessment score variables from the dataset.

The instance is then handed to a Python generator [26], which "translates" the structure of the XMI (XML Metadata Interchange) into React code through a set of custom and low-level components.

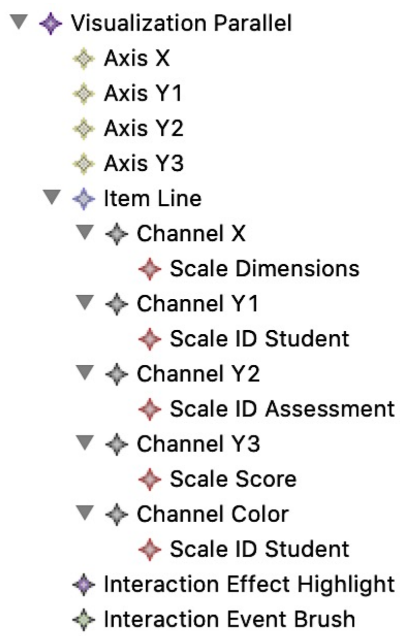


Fig. 3. An excerpt from a visualization's channels and scales.

For example, Fig. 4 presents a generated code excerpt from the dashboard. This code fragment is part of the props that are passed down to a specific React component. In this case, these props specify that the component will attend hovering and clicking events, and, if a data point is selected, it will be affected by highlighting that data point through a custom style.

```
interactions: {
    events: {
        hover: {
            type: 'individual',
            scope: 'global'
        },

        click: {
            type: 'individual',
            scope: 'global'
        }
    },
    effects: {
        highlight: {
            type: 'individual',
            selected_style: {
                ...this.props.dashboardStyle.interactions.hover_highlight
            },
            unselected_style: {
                ...this.props.dashboardStyle.interactions.unhover_highlight
            }
        },
    }
},
```

Fig. 4. React code fragment from the generated dashboard.

The outcome of this process is a React application that hosts the dashboard. The instantiated dashboard shown in Fig. 2 is presented in Fig 5. This dashboard holds two information visualizations. A scatter plot that represents each student on the y-axis and their scores in different assessments on the x-axis. On the other hand, a parallel coordinates plot shows the relationship between students, assessments, and scores, allowing them to detect patterns regarding these variables.

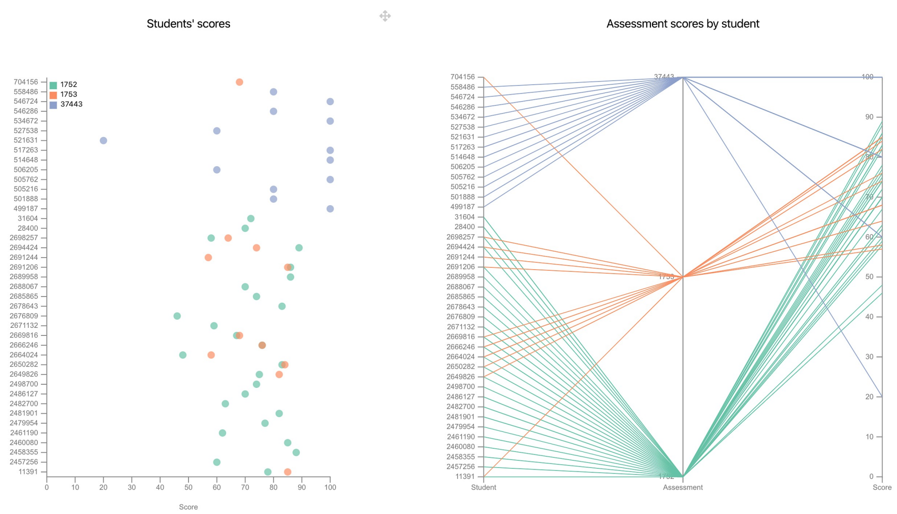


Fig. 5. Screenshot of the generated dashboard.

As shown in Fig. 4, some interaction patterns have been included in the generated dashboard. For example, the scatter plot component lets users hover on data points, provoking a highlight effect (as previously defined in the dashboard instance). Due to this configuration, when a user hovers on a data point in the first visualization, that data point is highlighted throughout the dashboard (Fig. 6).

This approach also allows the combination of different interaction patterns. For example, users can use a brush to select points on the parallel coordinates plot, affecting the scatter plot (Fig. 7).

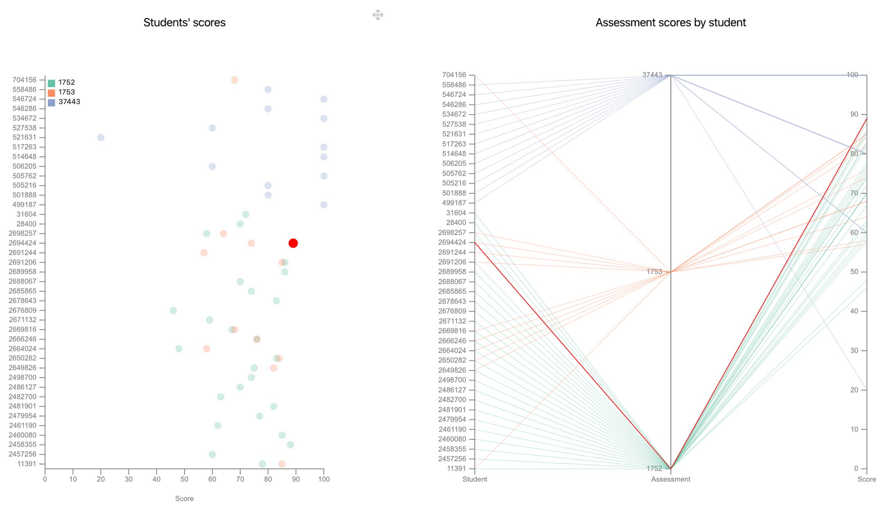


Fig. 6. Example of the addition of interaction patterns: hovering.

Fig. 7. Example of the addition of interaction patterns: brushing.

6 Discussion

Creating an information dashboard is not a trivial task; it involves the study of the domain, of the users, of the data, as well as the subsequent development process of the designed display. Given the current necessity to rely on data to make better decisions, it is important to have tools that ease knowledge discovery and insight delivery.

A dashboard meta-model has been developed to tackle the personalization of these tools. The meta-model could be seen as a conceptual artifact that supports the design and development processes of dashboards. However, in this case, the meta-model is used as an input for an automatic generative process of information dashboards.

The meta-model can be instantiated through the Eclipse Modeling Framework (EMF), thus obtaining a concrete model of a concrete Learning Analytics dashboard (an M1 model, Fig. 8).

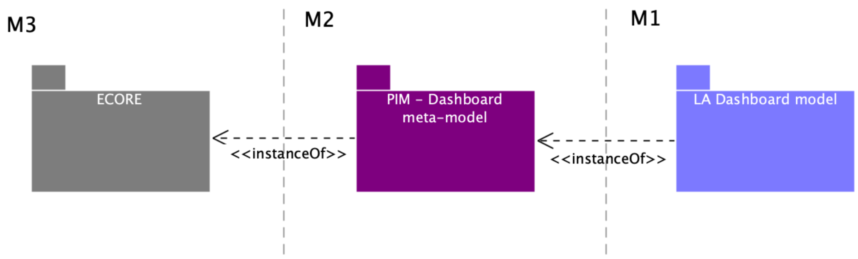


Fig. 8. Meta-model organization following the OMG architecture.

The obtained instance is the input for a dashboard generator, which arranges a set of software assets to compose a dashboard that matches the instance characteristics and features.

This work is focused on the definition of interaction patterns through the meta-model to obtain fully interactive dashboards. Interaction patterns are necessary to deliver good levels of user experience, as well as to improve the visual analysis process [27].

Interactions have been included in the meta-model through two conceptual classes, Event, and Effect, which define the events that the component will be listening to and the effects that the data point selections will have on their visual representation.

Adding interaction patterns to the instantiation process allows a more fine-grained definition of the dashboard features, which can be modified depending on different factors, such as the user expertise [28-30], the data domain [31], the analytical tasks and goals [32, 33], etc.

In this case, a Learning Analytics dashboard has been instantiated. This dashboard takes information regarding students' assessments and their scores. However, as stated in previous sections, it is straightforward to adapt the dashboard instance to other datasets with different variables. This is possible because the meta-model includes entities related to the datasets, their variables and potential operations that could transform data, meaning that they can be configured to support and generate indicators that allow dashboard users to reach meaningful insights related to their information goals.

Learning Analytics dashboards are very diverse, and the context and actors involved in each particular situation are crucial to building useful tools. Having the possibility of configuring information dashboards in a straightforward way allows for shifting the focus from the development process and giving more relevance to the design phases of the dashboard; Which analytic tasks will allow the user to reach his or her information goals? Which data variables should the dashboard display? How many views should the dashboard present? Which visual encodings allow better understanding given the target user expertise? Which interaction patterns would be more useful to support the user's analytical tasks?

Information dashboards are everywhere, but that does not mean that they are useful for everyone. It is crucial to find the best visual encodings, view arrangements, and interaction patterns based on each user's goals, characteristics, and datasets.

Designing a meta-model is a preliminary step to identify which features make a dashboard useful, effective or efficient. The abstract classes and structures of the meta-model can be used as inputs or outputs of external algorithms (for example, machine learning algorithms [31, 34]). Subsequent research will use the identified structures to build algorithms that maximize the usability, reliability, efficiency, etc., of these tools.

The model-driven development paradigm has been combined with a software product line approach to obtain a complete generative pipeline: starting from a conceptual phase (meta-model), software assets (core assets of the product line) were created to support the automatic generation of dashboards.

7 Limitations

Information dashboards usually have several intertwined features, elements and interaction patterns involved. This meta-model tries to capture the majority of them. However, dashboards are indeed very diverse, hampering the abstraction process. The current version of the meta-model supports dashboards based on different views and focused on structured data, but the visualization realm can also involve infographics, reports and other analysis assets that are not, at this time, supported by the presented meta-model.

8 Conclusions

A meta-model for information dashboards has been presented. This meta-model not only includes the visual elements of dashboards (visual marks, channels, axes, etc.) but also intangible components such as interaction patterns.

Including interaction patterns in the meta-model allows a more fine-grained specification and configuration of tailored information dashboards, meaning that not only the visual display can be customized, but also the methods in which users interact with datasets.

An example of application in the context of Learning Analytics has been carried out. This context involves very different actors and stakeholders, which might need different features from dashboards. Relying on the dashboard meta-model can make the development of these tools a more straightforward task.

Future research lines will involve the refinement of the meta-model to include rules and constraints, as well as in-depth user testing to test the usability of different interaction patterns and designs.

Acknowledgments

This research work has been supported by the Spanish *Ministry of Education and Vocational Training* under a FPU fellowship (FPU17/03276). This work has been partially funded by the Spanish Government Ministry of Economy and Competitiveness throughout the DEFINES project (Ref. TIN2016-80172-R).

References

1. Patil, D., Mason, H.: Data Driven. " O'Reilly Media, Inc." (2015)
2. Albright, S.C., Winston, W., Zappe, C.: Data analysis and decision making. Cengage Learning, Mason, OH, USA (2010)
3. Sarikaya, A., Correll, M., Bartram, L., Tory, M., Fisher, D.: What Do We Talk About When We Talk About Dashboards? IEEE Transactions on Visualization Computer Graphics 25, 682 - 692 (2018)
4. Aldrich, F., Sheppard, L.: Graphicacy-the fourth'R'? Primary Science Review 64, 8-11 (2000)
5. Schwendimann, B.A., Rodriguez-Triana, M.J., Vozniuk, A., Prieto, L.P., Boroujeni, M.S., Holzer, A., Gillet, D., Dillenbourg, P.: Perceiving learning at a glance: A systematic literature review of learning dashboard research. IEEE Transactions on Learning Technologies 10, 30-41 (2017)
6. Jivet, I., Scheffel, M., Drachsler, H., Specht, M.: Awareness is not enough: pitfalls of learning analytics dashboards in the educational practice. In: European Conference on Technology Enhanced Learning, pp. 82-96. Springer, (2017)
7. Vázquez-Ingelmo, A., García-Holgado, A., García-Peñalvo, F.J., Therón, R.: Dashboard Meta-Model for Knowledge Management in Technological Ecosystem: A Case Study in Healthcare. In: UCAmI 2019. MDPI, (2019)
8. Vázquez-Ingelmo, A., García-Peñalvo, F.J., Therón, R.: Capturing high-level requirements of information dashboards' components through meta-modeling. 7th International Conference on Technological Ecosystems for Enhancing Multiculturality (TEEM 2019), León, Spain (2019)
9. Vázquez-Ingelmo, A., García-Peñalvo, F.J., Therón, R., Conde González, M.Á.: Extending a dashboard meta-model to account for users' characteristics and goals for enhancing personalization. Learning Analytics Summer Institute (LASI) Spain 2019, Vigo, Spain (2019)
10. Cooper, M.M.: Data-driven education research. Science 317, 1171-1171 (2007)

11. Mandinach, E.B., Honey, M.: Data-driven school improvement: Linking data and learning. Teachers College Press (2008)
12. Custer, S., King, E.M., Atinc, T.M., Read, L., Sethi, T.: Toward Data-Driven Education Systems: Insights into Using Information to Measure Results and Manage Change. Center for Universal Education at The Brookings Institution (2018)
13. Yoo, Y., Lee, H., Jo, I.-H., Park, Y.: Educational dashboards for smart learning: Review of case studies. Emerging issues in smart learning, pp. 145-155. Springer (2015)
14. Teasley, S.D.: Student facing dashboards: One size fits all? Technology, Knowledge and Learning 22, 377-384 (2017)
15. Roberts, L.D., Howell, J.A., Seaman, K.: Give me a customizable dashboard: Personalized learning analytics dashboards in higher education. Technology, Knowledge and Learning 22, 317-333 (2017)
16. Dabbebi, I., Iksal, S., Gilliot, J.-M., May, M., Garlatti, S.: Towards Adaptive Dashboards for Learning Analytic: An Approach for Conceptual Design and implementation. In: 9th International Conference on Computer Supported Education (CSEDU 2017), pp. 120-131. SCITEPRESS, (2017)
17. Pleuss, A., Wollny, S., Botterweck, G.: Model-driven development and evolution of customized user interfaces. In: Proceedings of the 5th ACM SIGCHI symposium on Engineering interactive computing systems, pp. 13-22. ACM, (2013)
18. Kleppe, A.G., Warmer, J., Bast, W.: MDA Explained. The Model Driven Architecture: Practice and Promise. Addison-Wesley Longman Publishing Co., Inc., Boston, MA (2003)
19. Clements, P., Northrop, L.: Software product lines. Addison-Wesley, Boston, MA, USA (2002)
20. Gomaa, H.: Designing Software Product Lines with UML: From Use Cases to Pattern-Based Software Architectures. Addison Wesley Longman Publishing Co., Inc., Boston, MA, USA (2004)
21. Álvarez, J.M., Evans, A., Sammut, P.: Mapping between levels in the metamodel architecture. In: International Conference on the Unified Modeling Language, pp. 34-46. Springer, (2001)
22. Vázquez-Ingelmo, A., García-Peñalvo, F.J., Therón, R., Conde, M.Á.: Representing Data Visualization Goals and Tasks Through Meta-Modeling to Tailor Information Dashboards. Applied Sciences 10, 2306 (2020)
23. Harsu, M.: A survey on domain engineering. Citeseer (2002)
24. Kang, K.C., Cohen, S.G., Hess, J.A., Novak, W.E., Peterson, A.S.: Feature-oriented domain analysis (FODA) feasibility study. Carnegie-Mellon University, Software Engineering Institute (1990)
25. Kuzilek, J., Hlosta, M., Zdrahal, Z.: Open university learning analytics dataset. Scientific data 4, 170171 (2017)
26. Vázquez-Ingelmo, A., García-Peñalvo, F.J., Therón, R.: Taking advantage of the software product line paradigm to generate customized user interfaces for decision-making processes: a case study on university employability. PeerJ Computer Science 5, e203 (2019)

27. Shneiderman, B.: The eyes have it: A task by data type taxonomy for information visualizations. The Craft of Information Visualization, pp. 364-371. Elsevier (2003)
28. Lee, S., Kim, S.-H., Kwon, B.C.: Vlat: Development of a visualization literacy assessment test. IEEE transactions on visualization and computer graphics 23, 551-560 (2017)
29. Maltese, A.V., Harsh, J.A., Svetina, D.: Data visualization literacy: Investigating data interpretation along the novice—expert continuum. Journal of College Science Teaching 45, 84-90 (2015)
30. Toker, D., Conati, C., Carenini, G., Haraty, M.: Towards adaptive information visualization: on the influence of user characteristics. In: International conference on user modeling, adaptation, and personalization, pp. 274-285. Springer, (2012)
31. Dibia, V., Demiralp, Ç.: Data2Vis: Automatic generation of data visualizations using sequence to sequence recurrent neural networks. IEEE computer graphics and applications (2019)
32. Amar, R., Eagan, J., Stasko, J.: Low-level components of analytic activity in information visualization. In: IEEE Symposium on Information Visualization, 2005. INFOVIS 2005., pp. 111-117. IEEE, (2005)
33. Lam, H., Tory, M., Munzner, T.: Bridging from goals to tasks with design study analysis reports. IEEE transactions on visualization and computer graphics 24, 435-445 (2017)
34. Hu, K., Bakker, M.A., Li, S., Kraska, T., Hidalgo, C.: VizML: A Machine Learning Approach to Visualization Recommendation. In: Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems, pp. 128. ACM, (2019)

7.21 Appendix U. A Dashboard to Support Decision-Making Processes in Learning Ecosystems: A Metamodel Integration

Vázquez-Ingelmo, A., García-Holgado, A., García-Peñalvo, F. J., & Therón, R. (2020). A Dashboard to Support Decision-Making Processes in Learning Ecosystems: A Metamodel Integration. In Proceedings of the 2020 European Symposium on Software Engineering - ESSE 2020 (November 6-8, 2020, Rome, Italy) (pp. 80-87). ACM. https://doi.org/10.1145/3393822.3432326

A Dashboard to Support Decision-Making Processes in Learning Ecosystems: A Metamodel Integration

Andrea Vázquez-Ingelmo
GRIAL Research Group
Research Institute for Educational Sciences,
University of Salamanca
Salamanca, Spain
andreavazquez@usal.es

Alicia García-Holgado
GRIAL Research Group
Research Institute for Educational Sciences,
University of Salamanca
Salamanca, Spain
aliciagh@usal.es

Francisco José García-Peñalvo
GRIAL Research Group
Research Institute for Educational Sciences,
University of Salamanca
Salamanca, Spain
fgarcia@usal.es

Roberto Therón
GRIAL Research Group
Research Institute for Educational Sciences,
University of Salamanca
Salamanca, Spain
theron@usal.es

ABSTRACT
There are software solutions to solve most of the problems related to information management in any company or institutions, but still, there is a problem for transforming information into knowledge. Technological ecosystems emerge as a solution to combine existing tools and human resources to solve different problems of knowledge management. In particular, when the ecosystem is focused on learning processes associated with knowledge are named learning ecosystems. The learning ecosystem metamodel defined in previous works solves several problems related to the definition and implementation of these solutions. However, there are still challenges associated with improving the analysis and visualization of information as a way to discover knowledge and support decision making processes. On the other hand, there is a metamodel proposal to define customized dashboards for supporting decision-making processes. This proposal aims to integrate both metamodels as a way to improve the definition of learning ecosystems.

CCS Concepts
• Information systems → Open source software • Software and its engineering → Software design engineering • Software and its engineering → Software development techniques.

Keywords
Technological ecosystems; software ecosystems; MDA; metamodel; software engineering; learning ecosystem; knowledge management; decision-makers.

1. INTRODUCTION
Two main concepts are used to refer to changes in nowadays society, information society and knowledge society. The notion of the 'information society' is used above all when dealing with technological aspects and their effects on economic growth and employment [1]. On the other hand, in the knowledge society, the core element is not the technology but the ability to identify, produce, process, transform, disseminate and use the information to build and apply knowledge for human development [2]. According to [1], Heidenreich [3] associates the knowledge society to the increasing relevance of education and training processes in the initial phase as well as in the whole life course is underlined, and the increasing weight of knowledge-intensive services and communication.

Knowledge has become the backbone for development; it is a strategic factor for creating new policies, to plan new actions, and to foster innovation within organizations. Knowledge management is considered a sustainable competitive advantage [4], so the organizations expend part of their resources on building their capacity to share, create, and apply new knowledge continuously over time [5].

However, knowledge is not only present physically (i.e., in documents or books), it is also present in employees and the different processes carried out at organizations. According to [4], knowledge management processes must be able to support the transfer of implicit knowledge to tacit knowledge. This scattered nature of knowledge makes its management a complex and crucial task.

Software ecosystems emerge as a technological solution to support information and knowledge management in different contexts. According to [6, 7], institutions adopt a software ecosystem strategy to expand their organizational boundaries, share their platforms and resources with third parties, and define new business models.

Although the term software ecosystem is the most used in the literature, there are other terms that have distinctive characteristics. This is the case of the technological ecosystems, solutions that propose a decentralized configuration of software tools and non-technological components (such as methodologies, management plans, or human resources). Technological ecosystems can be composed by several elements; elements that are connected to each other and have different functions within the ecosystem. One of the main strengths of the technological ecosystem is that when their components collaborate, they exploit all of their benefits, obtaining the most out of their functionalities to provide elaborate services.

Furthermore, when the technological ecosystem is focused on learning processes associated with knowledge are named learning ecosystems [8, 9]. The definition and development of these solutions have challenges associated with the evolution of its components and the whole ecosystem, as well as the need to adapt to the changes that constantly occur in any organization.

This work presents a holistic meta-model to support decision-making processes in learning ecosystems. This meta-model integrates two meta-models defined in previous works. First, a learning ecosystem meta-model to support the definition of learning ecosystems based on open source software [10]. On the other hand, a dashboard meta-model to support the analysis of information in order to transform implicit knowledge in tacit knowledge.

Information dashboards are powerful tools that allow the recognition of patterns and interesting data points through visual analysis [11, 12]. However, dashboards can be very diverse in terms of design, context, audience, pursued goals, supported tasks, etc., [13-18], which makes the development of these tools a complex activity.

By abstracting the common elements of information dashboards through meta-modeling it is possible to obtain a general structure of dashboards that can be instantiated and adapted to any kind of contexts, data domains or audiences.

Furthermore, as it will be detailed, the inclusion of dashboard users and their requirements as elements of the meta-model enables the integration of this meta-model as a part of technological ecosystems, specifically, learning ecosystems, providing support to discover knowledge and support decision making processes [19].

The rest of this paper is organized as follows. Section 2 outlines the methodology followed to develop the meta-models. Section 3 describes the learning ecosystem meta-model, followed by section 4, in which the dashboard meta-model is presented. Section 4 discusses the integration of both meta-models. Finally, section 5, where the conclusions derived from this work are depicted.

2. METHODOLOGY

Model-driven development (MDD) [20, 21] allows separating the data and the operations specification of the system from lower-level details, like the technical aspects related to a specific program language or platforms.

The Object Management Group (OMG) proposes the model-driven architecture (MDA) as a guideline to implement this approach. This architecture provides a framework for software development which employs models to describe and define the target system [22]. The main difference between MDD and MDA is that the OMG proposal uses a set of standards: meta-object facility (MOF), unified modeling language (UML), XML (Extensible Markup Language) metadata interchange (XMI), and query/view/transformation (QVT).

In this case, the dashboard is a part of the learning ecosystem, which is based on a meta-model defined and validated in previous works. The first version of the learning ecosystem meta-model is based on MOF, and the last validated version is an instance of Ecore [10]. Both versions are M2-models. The model has served as a map to develop and deploy the ecosystem in a real-world context.

The dashboard meta-model is also part of the four-layer meta-model architecture proposed by the OMG, in which a model at one layer is used to specify models in the layer below [23]. In particular, the dashboard meta-model is an instance of MOF (i.e., an M2-model), so it can be instantiated to obtain M1-models.

The integration of both meta-models is possible because of both are Platform Independent Models (PIM) at M2 layer, although one is instantiated from Ecore (learning ecosystem meta-model) and other from MOF (dashboard meta-model). To get the holistic meta-model, the dashboard meta-model was transformed in an instance of Ecore using Graphical Modelling for Ecore included in EMF.

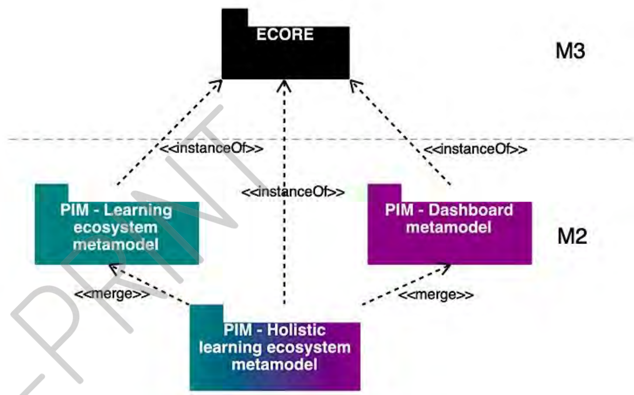


Figure 1. Meta-models organized in the four-layer metamodel architecture.

3. LEARNING ECOSYSTEM META-MODEL

There are a large number of open source tools that allow knowledge management in different ways, with particular emphasis on content managers and document repositories. On the other hand, from the point of view of learning management, there is a wide variety of learning platforms (Learning Management System, or LMS [24, 25]) and tools that allow the definition of Personal Learning Environments (PLE) [26, 27]. Technological ecosystems for learning must be able to combine some of these tools to support knowledge and learning processes in heterogeneous contexts, from institutional environments to private enterprises. Besides, they must be able to incorporate emerging tools, as well as to remove those that become obsolete or that users do not use, in such a way that the system must be in continuous evolution.

Despite the advantages offered by technological ecosystems, the development of such solutions is more complex than traditional information systems. The definition of a particular ecosystem requires knowledge and selection of appropriate systems and services to meet the needs of a particular context. Likewise, the ecosystem should be prepared to evolve and adapt to the changing needs of the environment and users; meanwhile, interoperability between the different components must ensure a high degree of integration and cohesion [9].

The learning ecosystem meta-model is proposed as a solution to improve the processes of definition and development of technological ecosystems, in order to solve the different challenges and problems identified through the analysis of a set of learning ecosystems deployed in different contexts, and with very diverse objectives.

In particular, the meta-model [28] is a Platform-Independent Model (PIM) to define learning ecosystems based on Open Source software (Figure 2). It is an instance of Ecore with a set of constraints defined with Object Constraint Language (OCL).

The meta-model represents the three main elements of a learning ecosystem.

First, the different software tools that compose the ecosystem: data repositories, monitoring tools, user management systems, indexing services, decision-making tools, etc.

Second, the proposed meta-model includes the human factor at the same level as the software because these human resources (management definition, methodologies, and users) are the key elements to ensure the evolution of the ecosystem.

Finally, the third element is the information flows used to support the interaction between the other elements in the ecosystem.

The interaction between the software tools is implemented by services and properties files. On the other hand, the interaction between software tools and users has a substantial impact on the ecosystem; for this reason, these interactions are also represented through information flows. All information flows are based on the objectives defined by management.

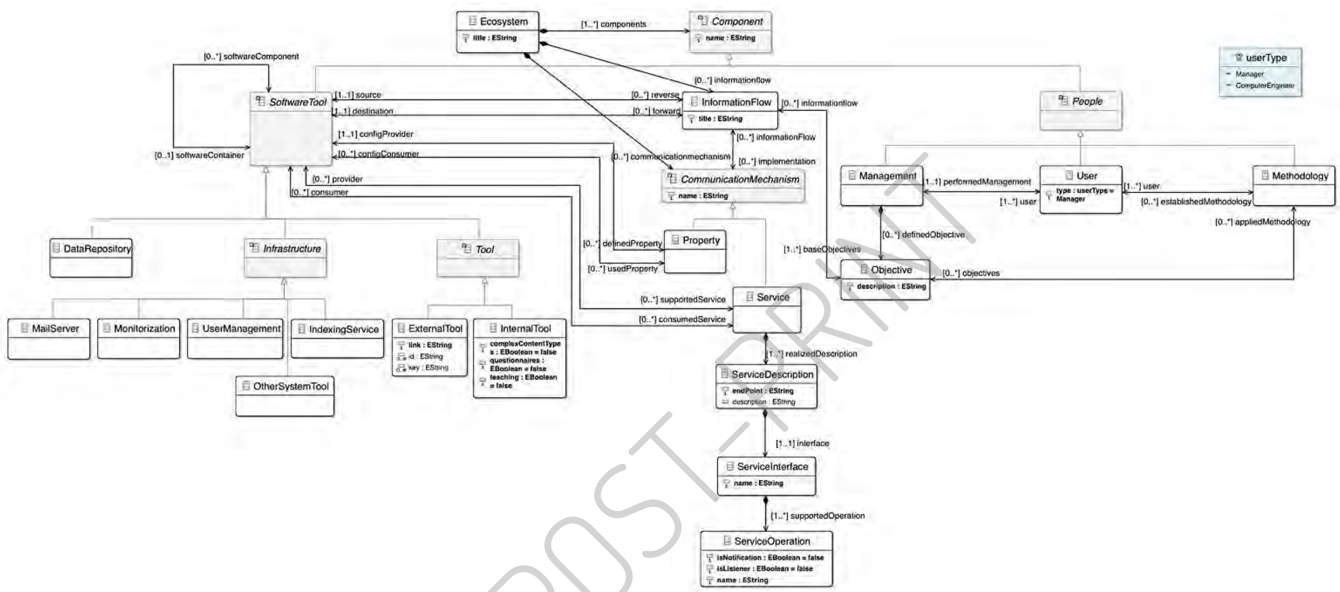


Figure 2. Learning ecosystem meta-model in Ecore. This image is available in high resolution at https://doi.org/10.5281/zenodo.1066369. Source: [9].

4. DASHBOARD ECOSYSTEM META-MODEL

The dashboard meta-model is also composed by a variety of pieces that allow the definition of different types of dashboard and information visualizations. The dashboard meta-model can be divided into three main sections: the user, the layout and the components.

Figure 3 shows an excerpt of the dashboard meta-model containing the mentioned three sections. A detailed view of the components section can be consulted in Figure 4.

The layout and the components are more technical aspects of dashboards, as they will compose the final display.

However, how many visualizations will the dashboard hold? How these views will be arranged? What type of visualizations will the dashboard display? What type of interaction patterns will the dashboard support? Will the different views be linked?

These questions cannot be answered in an arbitrary manner. These are design decisions, and they need to be driven by the final consumers of dashboards: the users.

Including the user in this meta-model is essential, because they will be using the delivered dashboards to reach insights, to support their decision-making processes or to explore certain datasets.

The user is defined in terms of significant and influential aspects to support a personalized dashboard design, i.e., the factors that influence the design process of a dashboard [29]. Given that, the user entity is decomposed in terms of his or her goals and his or her characteristics.

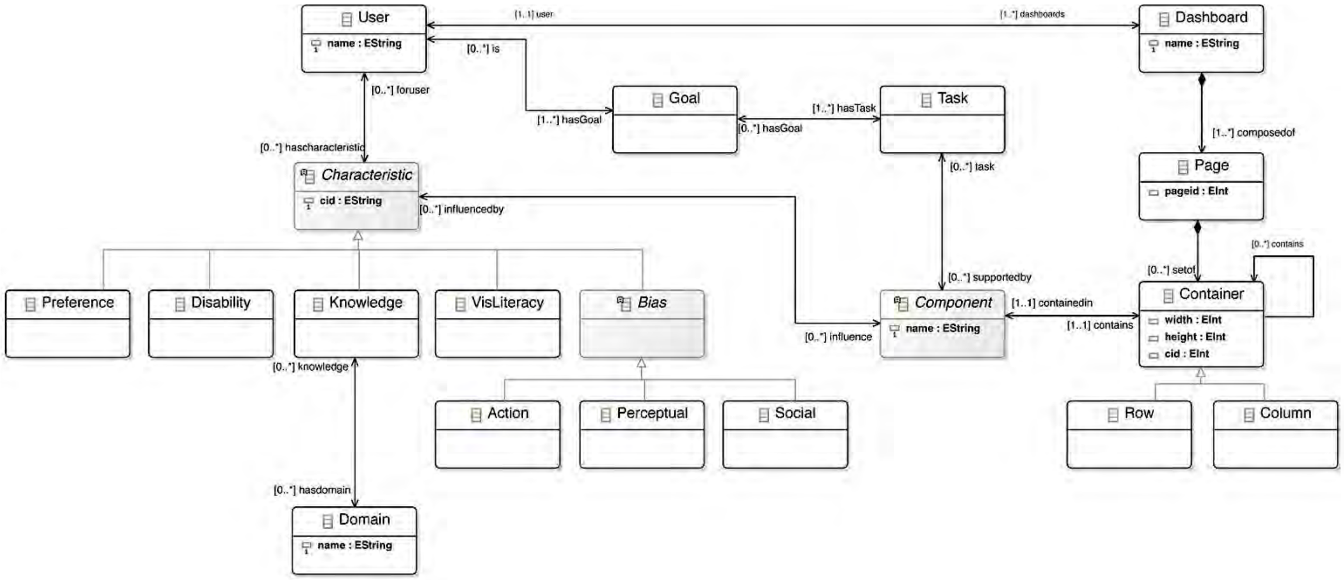


Figure 3. User, layout and components section of the dashboard meta-model proposal.

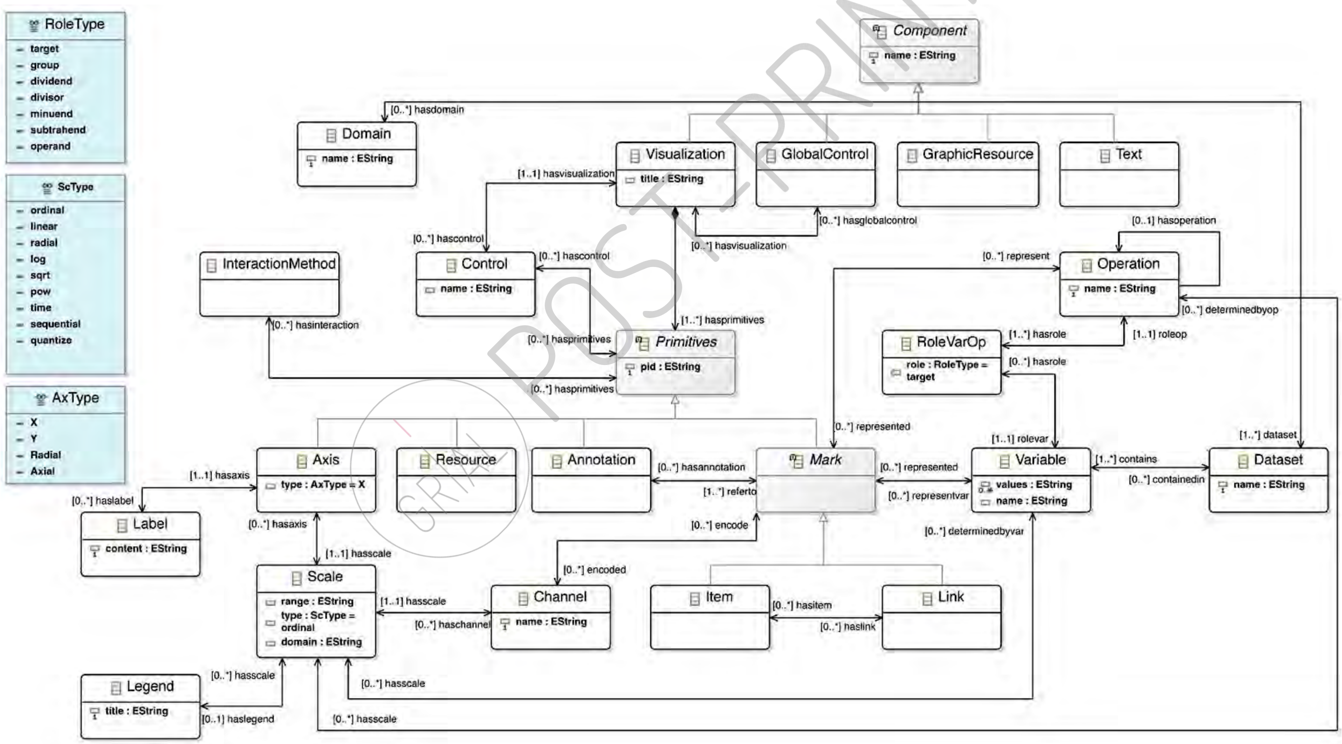


Figure 4. Detailed view of the components section of the dashboard meta-model proposal.

Firstly, a crucial concept arises; Goal. A user must have at least one goal for using a dashboard, however implicit.

Goals, in turn, can be broken down into individual and more specific, low-level tasks. Simple goals can be accomplished by performing a few tasks.

However, more elaborated goals might involve several specific and chained tasks, which could involve different data dimensions to support complex decision-making processes [15, 17, 30].

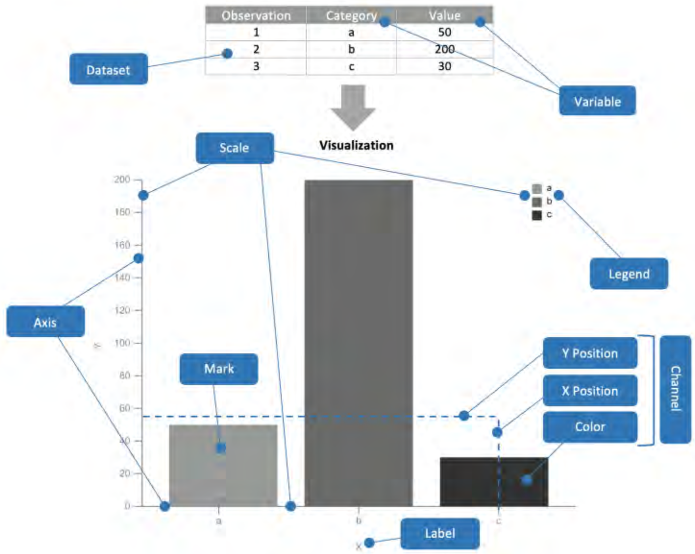


Figure 5. Example of the different elements that compose a visualization (in this case, a bar chart).

Finally, a user can have a set of identified characteristics. Characteristics can be diverse. For example, preferences, disabilities, knowledge about different domains, visualization literacy, and bias (action, perceptual, or social bias) are different kind of characteristics. These characteristics can influence the design process, thus needing to adapt the dashboard's components to match the identified user aspects.

Regarding the more technical details of the dashboards (i.e., the layout and components), the purpose of the layout section of the dashboard is to model the generic structure of a dashboard, which can be composed of different containers (rows or columns) that hold different components.

In terms of the components of the dashboard, several elements are identified. The main components of dashboards are the information visualizations that display data, but also the controls (handlers, filters, and so on), graphic resources, or text that complement these visualizations (Figure 4).

Information visualizations, in turn, are composed of primitives, which can be different visual marks that encode data variables through channels (i.e., color, size, position, etc.). These primitives are the core of information visualizations, because they are the elements that hold the actual data [12, 31].

Figure 5 shows a practical example of the identification of different parts of information visualizations through the dashboard meta-model classes.

In this example, there are two scales that represent two variables (the "Category" variable through an ordinal scale, and the "Value" variable through a linear scale). The domain of these scales are the set of values from the variables. For instance, the domain of the scale that encode the X position of the visual marks is the set of values retrieved from the "Category" variable (i.e., 'a', 'b', and 'c'). Axes, on the other hand, support the visualization of the scales' domains.

The visual marks of this visualization are bars with a specific position along both X and Y axes and a specific color based on a color scale that encodes the "Category" variable. Scales map the data values to another specific range of values in order to encode the information, that is why these entities are related both to the dataset variables' values (to obtain the domain) and to the visual channels or encodings (to encode these values using another specific range of values, like color codes or screen positions).

The dashboard metamodel was an instance of MOF, but it was transformed into an instance of Ecore using Graphical Modelling for Ecore to enable its the connection to the learning ecosystem meta-model, as mentioned in the methodology section.

During the transformation, some changes were introduced to compile with the Ecore rules, so the dashboard meta-model described in this section differs in some details from the previous version [32, 33].

These changes only address modeling issues to enable the instantiation of Ecore. These are the modified aspects of the meta-model:

- Renaming some classes to remove white spaces and introducing CamelCase notation: *VisLiteracy*, *GlobalControl*, *GraphicResource*, and *InteractionMethod*.
- Introduction of id attributes in each class in order to allow the instantiation of the meta-model in a M1-model.
- Introduction of enumeration classes to enclose the values for some attributes: *RoleType*, *AxType*, and *ScType*.
- Review of the associations' navigability because in Ecore the navigability is always represented.
- Introduction of names for each relationship.
- Translation of the *RoleVarOp* association class into binary associations (association class is not supported by Ecore).
- Transformation of the reflexive composition association of the *Container* class into a reflexive binary association.
- Transformation of the aggregations into binary associations (aggregation is not supported by Ecore).

The mentioned modifications are crucial for the next steps. The introduction of identifiers and the explicit navigability of the relationships are necessary to instantiate the metamodel and to introduce constraints through OCL, for example.

5. A HOLISTIC ECOSYSTEM METAMODEL

As introduced at the beginning, ecosystems are composed by different elements with different functions and goals. However, these elements are more powerful when connected and when they collaborate with each other through information flows.

That is why a holistic solution is proposed, in which each part of it is strengthened when collaborating among them. The ecosystem is seen as a whole, and not only as individual parts with no relationships between one another.

Although the learning ecosystem meta-model proposed solve most of the problems associated to the definition and development of theses technological solutions, there are some issues related to the analysis of the information flows and the support to decision-making processes that should be improved.

The dashboard meta-model presented in Figures 3 and 4 is connected to the ecosystem meta-model (available at https://doi.org/10.5281/zenodo.3561320 [34]).

The ecosystem meta-model represents each software component as a black box, to provide a high-level abstraction of its structure. The dashboard meta-model provides detail to model these tools as a part of the ecosystem. The complexity of dashboards requires an in-depth analysis of the domain to identify their main commonalities and features, and how these features relate with each other.

Although the dashboard meta-model includes more details regarding these tools' structure and elements, the meta-model is still at a M2-level in terms of the OMG's four-layer architecture. The presented dashboard meta-model is an instance of Ecore, as well as the learning ecosystem meta-model.

These two M2-level meta-models are connected by some elements present both in the dashboard meta-model and the ecosystem meta-model. On the one hand, it has been justified the necessity of including users in the dashboard meta-model because they are the drivers and consumers of the displayed data.

The human factor also plays a crucial role in the learning ecosystem meta-model because the technology is defined and evolved to support the users' needs.

On the other hand, there are two relevant elements shared in both meta-models too. The dashboard *Goals* (within the dashboard meta-model) are represented as *Objectives* within the learning ecosystem meta-model.

These elements are represented by a set of *Tasks*, and *Information Flows*, respectively. The relevance of these entities is that they are the core of the meta-model, because they frame the required components to achieve the goals or objectives set.

Figure 6 shows the connection between both meta-models. The dashboard *Goal* is merged with *Objective. T*he connection between *Goal* and *User* in the dashboard meta-model is replaced by the association between *User* and *Objective* through the *Management*. In this sense, all the goals that support the definition of the dashboard are connected to the management decisions in the ecosystem.

Regarding the *Dashboard*, the main class to instantiate the dashboard meta-model is connected with the learning ecosystem as a *Tool*. Besides, the connection between *User* and *Dashboard*, which has a particular impact on the dashboard meta-model, is included in the proposal.

The information flows and tasks are different concepts, so it is not possible to merge them. For this reason, the dashboard *Task* is included in Figure 6.

On the other hand, a new communication mechanism is included to implement the information flows, the *Dataset*, as a way to represent the integration between the dashboard and other software tools in the learning ecosystem.

Regarding the dashboard *Component*, it is renamed as *DashboardComponent* to distinguish it from the learning ecosystem *Component*.

Finally, the connection between dashboard *Characteristic* and *User* appears in the new proposal. Tasks are supported by the dashboard's elements, that are also influenced by the user characteristics to match his or her information requirements.

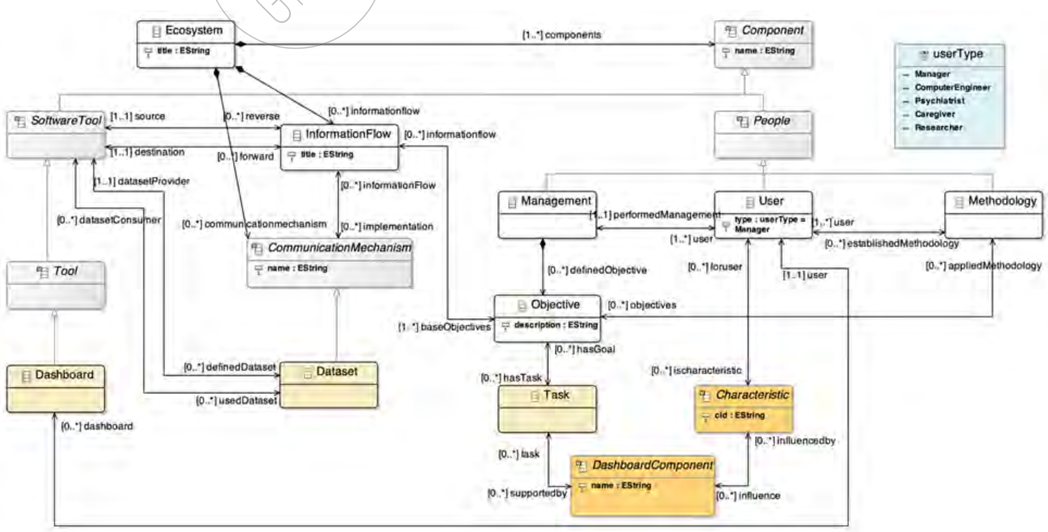


Figure 6. Connection between both meta-models.

6. CONCLUSIONS

An integration of two meta-models has been proposed. Specifically, a dashboard meta-model has been included within a learning ecosystem meta-model to solve some issues related to knowledge discovery and decision-making processes in the learning ecosystems.

The dashboard meta-model provides a skeleton that can be adapted to instantiate concrete dashboard solutions. The role of the dashboard is to support decision-making processes through visual analysis.

Including the user within the meta-model is crucial, as their goals and data requirements are the drivers of the dashboard configuration process.

On the other hand, the learning ecosystem meta-model solves several problems related to the definition and implementation of learning ecosystems. Learning ecosystems combine tools to support knowledge management. Including an information dashboard within the learning ecosystem address the improvement of knowledge discovery within the ecosystem, by providing a tool to visually analyze information flows.

However, although the learning ecosystem is validated and its quality was checked through the framework defined by López-Fernández et al. [35]. it is necessary to validate and apply the same framework to the meta-model proposed due to the dashboard meta-model is not fully validated in previous works.

Furthermore, future research lines will involve the refinement of the meta-model through the addition of constraints, rules, and design guidelines, in addition to the testing of products instantiated from these meta-models. The goal is to obtain a tool to automatically generate information dashboards based on the context of application using, for example, machine learning algorithms [36-38].

7. ACKNOWLEDGMENTS

This research was partially funded by the Spanish Government Ministry of Economy and Competitiveness throughout the DEFINES project grant number (TIN2016-80172-R) and the Ministry of Education of the Junta de Castilla y León (Spain) throughout the T-CUIDA project (SA061P17).

This research was supported by the Spanish Ministry of Education, Culture and Sport under a FPU fellowship (FPU17/03276).

8. REFERENCES

[1] KRÜGER, K., 2006. El concepto de 'Sociedad del Conocimiento'. Revista bibliográfica de geografía y ciencias sociales XI, 683.

[2] UNESCO, 2005. UNESCO World Report: Towards Knowledge Societies. UNESCO Publishing, Paris.

[3] HEIDENREICH, M., 2003. Die Debatte um die Wissensgesellschaft. In Wissenschaft in der Wissensgesellschaft, S. BÖSCHEN and I. SCHULZ-SCHAEFFER Eds. VS Verlag für Sozialwissenschaften, Wiesbaden. DOI= http://dx.doi.org/10.1007/978-3-663-07783-1_2.

[4] NONAKA, I. and TAKEUCHI, H., 1995. The knowledge-creating company: How Japanese companies create the dynamics of innovation. Oxford university press, Oxford, UK.

[5] HARGREAVES, A., 2003. Teaching in the knowledge society: Education in the age of insecurity. Teachers College Press.

[6] MANIKAS, K. and HANSEN, K.M., 2013. Software ecosystems – A systematic literature review. Journal of Systems and Software 86, 5 (05/05/2013), 1294-1306. DOI= http://dx.doi.org/10.1016/j.jss.2012.12.026.

[7] WEIBLEN, T., GIESSMANN, A., BONAKDAR, A., and EISERT, U., 2012. Leveraging the software ecosystem:Towards a business model framework for marketplaces. In Proceedings of the International Conference on Data Communication Networking, e-Business and Optical Communication Systems SciTePress, 187-193.

[8] LLORENS-LARGO, F., MOLINA, R., COMPAÑ, P., and SATORRE, R., 2014. Technological Ecosystem for Open education. In Smart Digital Futures 2014, R. NEVES-SILVA, G.A. TSIHRINTZIS, V. USKOV, R.J. HOWLETT and L.C. JAIN Eds. IOS Press, 706-715. DOI= http://dx.doi.org/10.3233/978-1-61499-405-3-706.

[9] GARCÍA-HOLGADO, A., 2018. Análisis de integración de soluciones basadas en software como servicio para la implantación de ecosistemas tecnológicos educativos. In Programa de Doctorado en Formación en la Sociedad del Conocimiento University of Salamanca, Salamanca, Spain.

[10] GARCÍA-HOLGADO, A. and GARCÍA-PEÑALVO, F.J., 2019. Validation of the learning ecosystem metamodel using transformation rules. Future Generation Computer Systems 91, 300-310. DOI= http://dx.doi.org/10.1016/j.future.2018.09.011.

[11] FEW, S., 2006. Information dashboard design.

[12] WARE, C., 2012. Information visualization: perception for design. Elsevier.

[13] SARIKAYA, A., CORRELL, M., BARTRAM, L., TORY, M., and FISHER, D., 2018. What Do We Talk About When We Talk About Dashboards? IEEE Transactions on Visualization Computer Graphics 25, 1, 682 - 692.

[14] VÁZQUEZ-INGELMO, A., GARCIA-PEÑALVO, F.J., and THERÓN, R., 2019. Information Dashboards and Tailoring Capabilities - A Systematic Literature Review. IEEE Access 7, 109673-109688. DOI= http://dx.doi.org/10.1109/ACCESS.2019.2933472.

[15] AMAR, R., EAGAN, J., and STASKO, J., 2005. Low-level components of analytic activity in information visualization. In IEEE Symposium on Information Visualization, 2005. INFOVIS 2005. IEEE, 111-117.

[16] BERINATO, S., 2016. Good charts: The HBR guide to making smarter, more persuasive data visualizations. Harvard Business Review Press, Brighton, MA, US.

[17] BREHMER, M. and MUNZNER, T., 2013. A multi-level typology of abstract visualization tasks. IEEE transactions on visualization and computer graphics 19, 12, 2376-2385.

[18] CHI, E.H.-H., 2000. A taxonomy of visualization techniques using the data state reference model. In IEEE Symposium on Information Visualization 2000. INFOVIS 2000. Proceedings IEEE, 69-75.

[19] PATIL, D. and MASON, H., 2015. Data Driven. " O'Reilly Media, Inc.".

[20] PLEUSS, A., WOLLNY, S., and BOTTERWECK, G., 2013. Model-driven development and evolution of customized user interfaces. In Proceedings of the 5th ACM SIGCHI symposium on Engineering interactive computing systems ACM, 13-22.

[21] KLEPPE, A.G., WARMER, J., and BAST, W., 2003. MDA Explained. The Model Driven Architecture: Practice and Promise Addison-Wesley Longman Publishing Co., Inc., Boston, MA.

[22] MELLOR, S.J., SCOTT, K., UHL, A., and WEISE, D., 2002. Model-Driven Architecture. In Advances in Object-Oriented Information Systems: OOIS 2002 Workshops Montpellier, France, September 2, 2002 Proceedings, J.-M. BRUEL and Z. BELLAHSENE Eds. Springer Berlin Heidelberg, Berlin, Heidelberg, 290-297. DOI= http://dx.doi.org/10.1007/3-540-46105-1_33.

[23] ÁLVAREZ, J.M., EVANS, A., and SAMMUT, P., 2001. Mapping between Levels in the Metamodel Architecture. In ≪UML≫ 2001 — The Unified Modeling Language. Modeling Languages, Concepts, and Tools. UML 2001. Lecture Notes in Computer Science, M. GOGOLLA and C. KOBRYN Eds. Springer, Berlin, Heidelberg, 34-46. DOI= http://dx.doi.org/10.1007/3-540-45441-1_4.

[24] GARCÍA-PEÑALVO, F.J. and SEOANE-PARDO, A.M., 2015. Una revisión actualizada del concepto de eLearning. Décimo Aniversario. Education in the Knowledge Society 16, 1, 119-144. DOI= http://dx.doi.org/10.14201/eks2015161119144.

[25] GARCÍA-PEÑALVO, F.J. and GROS, B., 2016. Future trends in the design strategies and technological affordances of e-learning. In Learning, Design, and Technology. An International Compendium of Theory, Research, Practice, and Policy, M. SPECTOR, B.B. LOCKEE and M.D. CHILDRESS Eds. Springer International Publishing, Switzerland, 1-23. DOI= http://dx.doi.org/10.1007/978-3-319-17727-4_67-1.

[26] WILSON, S., LIBER, O., JOHNSON, M., BEAUVOIR, P., SHARPLES, P., and MILLIGAN, C., 2007. Personal Learning Environments: Challenging the dominant design of educational systems. Journal of e-Learning and Knowledge Society 3, 3, 27-38.

[27] ATTWELL, G., 2007. The personal learning environments– The future of eLearning? eLearning Papers 2, 1.

[28] GARCÍA-HOLGADO, A. and GARCÍA-PEÑALVO, F.J., 2017. A metamodel proposal for developing learning ecosystems. In Learning and Collaboration Technologies. Novel Learning Ecosystems. 4th International Conference, LCT 2017. Held as Part of HCI International 2017, Vancouver, BC, Canada, July 9–14, 2017. Proceedings, Part I, P. ZAPHIRIS and A. IOANNOU Eds. Springer International Publishing, Switzerland, 100-109. DOI= http://dx.doi.org/10.1007/978-3-319-58509-3_10.

[29] VÁZQUEZ INGELMO, A., GARCÍA-PEÑALVO, F.J., and THERÓN, R., 2019. Tailored information dashboards: A systematic mapping of the literature. In Proceedings of the Interacción2019 (Donostia, Spain2019), ACM.

[30] LAM, H., TORY, M., and MUNZNER, T., 2017. Bridging from goals to tasks with design study analysis reports. IEEE transactions on visualization and computer graphics 24, 1, 435-445.

[31] MUNZNER, T., 2014. Visualization analysis and design. AK Peters/CRC Press, Boca Raton, FL, USA.

[32] VÁZQUEZ INGELMO, A., GARCÍA-PEÑALVO, F.J., THERÓN, R., and CONDE GONZÁLEZ, M.Á., 2019. Extending a dashboard meta-model to account for users' characteristics and goals for enhancing personalization. In Proceedings of the Learning Analytics Summer Institute (LASI) Spain 2019 (Vigo, Spain2019).

[33] VÁZQUEZ-INGELMO, A., GARCÍA-PEÑALVO, F.J., and THERÓN, R., Article In Press. Capturing high-level requirements of information dashboards' components through meta-modeling. In Proceedings of the 7th International Conference on Technological Ecosystems for Enhancing Multiculturality (TEEM 2019) (León, SpainArticle In Press).

[34] VÁZQUEZ-INGELMO, A., GARCÍA-HOLGADO, A., GARCÍA-PEÑALVO, F.J., and THERÓN, R., 2019. Ecore version of the metamodel for information dashboards. DOI= http://dx.doi.org/10.5281/zenodo.3561320.

[35] LÓPEZ-FERNÁNDEZ, J.J., GUERRA, E., and DE LARA, J., 2014. Assessing the Quality of Meta-models. In MoDeVVa, F. BOULANGER, M. FAMELIS and D. RATIU Eds. CEUR Workshop Proceedings, Valencia, Spain, 3-22.

[36] HU, K., BAKKER, M.A., LI, S., KRASKA, T., and HIDALGO, C., 2019. VizML: A Machine Learning Approach to Visualization Recommendation. In Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems ACM, 128.

[37] THANG, P.Q., LAM, H.T., and THUY, N.T., 2018. Improving Simplification of Support Vector Machine for Classification. International Journal of Machine Learning and Computing 8, 4.

[38] VIDNEROVÁ, P. and NERUDA, R., 2018. Kernel Function Tuning for Single-Layer Neural Networks. International Journal of Machine Learning and Computing 8, 4, 354-360.

7.22 Appendix V. Beneficios de la aplicación del paradigma de líneas de productos software para generar dashboards en contextos educativos

I.S.S.N.: 1138-2783

Beneficios de la aplicación del paradigma de líneas de productos *software* para generar *dashboards* en contextos educativos

(Benefits of the software product line paradigm in generating dashboards for educational contexts)

Andrea Vázquez-Ingelmo
Roberto Therón
Universidad de Salamanca, USAL (España)

DOI: http://dx.doi.org/10.5944/ried.23.2.26389

Cómo referenciar este artículo:
Vázquez-Ingelmo, A., y Therón, R. (2020). Beneficios de la aplicación del paradigma de líneas de productos software para generar *dashboards* en contextos educativos. *RIED. Revista Iberoamericana de Educación a Distancia, 23*(2), pp. 169-185. doi: http://dx.doi.org/10.5944/ried.23.2.26389

Resumen

Los datos son cruciales para mejorar la toma de decisiones y obtener mayores beneficios en cualquier tipo de actividad. Sin embargo, la gran cantidad de información generada debido a las nuevas tecnologías ha convertido el análisis de los datos y la generación de conocimiento a partir de ellos en una tarea compleja. Numerosas herramientas han surgido para facilitar esta generación de conocimiento, como es el caso de los *dashboards* o paneles de información. Aunque los paneles de control sean herramientas muy potentes, su efectividad puede verse afectada por un mal diseño o por no tener en cuenta el contexto en el que se encuadran. Por ello, es necesario diseñar y crear paneles de control a medida en función de la audiencia y dominio de los datos. Crear paneles de control personalizados puede ser muy beneficioso, pero también un proceso costoso en lo que al tiempo y recursos se refiere. Este trabajo presenta una aplicación del paradigma de líneas de productos *software* para generar paneles de control adaptados a cualquier contexto de manera más sencilla, reutilizando tanto componentes *software* como conocimiento. Uno de los contextos que puede verse especialmente favorecido por este enfoque es el contexto educativo, donde la analítica del aprendizaje y el análisis de datos sobre el rendimiento de los estudiantes se está popularizando. Contar con paneles de control personalizables para cualquier rol (estudiante, profesor, administrador, etc.) puede mejorar los procesos de toma de decisiones, mostrando a cada usuario la información que más le interesa de la forma que mejor le permita comprenderla.

Palabras clave: tecnología educativa; investigación educativa; alfabetización en datos.

Abstract

Data are crucial to improve decision-making and to obtain greater benefits in any type of activity. However, the large amount of information generated by new technologies has made data analysis and knowledge generation a complex task. Numerous tools have emerged to facilitate this knowledge generation, such as dashboards. Although dashboards are very powerful tools, their effectiveness can be affected by a bad design or by not taking into account the context in which they are placed. Therefore, it is necessary to design and create tailored dashboards according to the audience and data domain. Creating tailored dashboards can be very beneficial, but also a costly process in terms of time and resources. This paper presents an application of the software product line paradigm to generate dashboards adapted to any context in a more straightforward way by reusing both software components and knowledge. One of the contexts that can be especially favored by this approach is the educational context, where Learning Analytics and the analysis of student performance to improve learning methodologies are becoming very popular. Having tailored dashboards for any role (student, teacher, administrator, etc.) can improve decision making processes by showing each user the information that interests them most in the way that best enables them to understand it.

Keywords: educational technology; educational research; data literacy

Los datos han comenzado a ser cruciales en cualquier actividad, ya sean actividades profesionales o cotidianas. Realizar el proceso de toma de decisiones con una base robusta de información es crucial para potenciar los resultados positivos de esta tarea (Albright, Winston y Zappe, 2010). Los frutos de esta toma de decisiones son una serie de acciones, acciones que buscan reportar algún tipo de beneficio en el contexto de aplicación.

Sin embargo, llevar a cabo una toma de decisiones dirigida por datos (Patil y Mason, 2015) no es una tarea trivial. Primero, porque es necesaria una cantidad significativa de datos para poder generar conocimiento. Y segundo, porque los procesos de análisis de dichos datos requieren que la persona al frente de la toma de decisiones o del análisis sea capaz de comprender e interpretar conjuntos de datos que en muchas ocasiones son complejos y extensos.

No obstante, la evolución de las tecnologías ha permitido que estas tareas de análisis estén al alcance de perfiles menos técnicos. Existen herramientas que facilitan el análisis y la generación de conocimiento a partir de conjuntos de datos. Una de las herramientas que más se han popularizado son los *dashboards* o paneles de control (Sarikaya, Correll, Bartram, Tory y Fisher, 2018).

Los paneles de control son elementos digitales compuestos por una serie de visualizaciones de datos. Estas visualizaciones tratan de codificar información a

DOI: http://dx.doi.org/10.5944/ried.23.2.26389 – ISSN: 1138-2783 – E-ISSN: 1390-3306

través de las propiedades de sus elementos visuales, como la posición, los colores, los tamaños, las formas, etc. (Munzner, 2014).

Pero contar con un panel de control no garantiza la generación de conocimiento. Es necesario tener en cuenta la audiencia y el perfil de los usuarios que utilizarán estas herramientas. Pueden existir usuarios que puedan comprender visualizaciones complejas, mientras que otros necesitarán otras metáforas visuales para entender correctamente sus conjuntos de datos (Aldrich y Sheppard, 2000).

Así pues, los paneles de control son herramientas potentes, pero necesitan un elaborado proceso de diseño y desarrollo para cumplir con las expectativas de sus usuarios y para ser eficaces a la hora de generar conocimiento. Además, en contextos en los que existen diversos roles y perfiles de usuario, este proceso de diseño es aún más complejo, dado que hay que tener en cuenta muchos más factores. Una solución sería realizar un panel de control hecho a medida para cada perfil implicado, pero sería prácticamente imposible de llevar a cabo debido a la cantidad de tiempo y recursos que serían necesarios.

En la literatura se han considerado diversos enfoques para permitir la automatización de este proceso de diseño e implementación de paneles de control, con el objetivo de disminuir el tiempo necesario para desarrollar paneles de control hechos a medida (Vázquez-Ingelmo, García-Peñalvo y Therón, 2019c).

Entre estos enfoques se encuentran desde asistentes gráficos de configuración que permiten a los usuarios elegir qué gráficos formarán parte de sus paneles de control, hasta desarrollo dirigido por modelos que consiguen generar paneles de control personalizados basándose en descripciones formales del dominio (Kintz, Kochanowski y Koetter, 2017; Logre, Mosser, Collet y Riveill, 2014; Palpanas, Chowdhary, Mihaila y Pinel, 2007; Pleuss, Wollny y Botterweck, 2013), entre otras variadas soluciones.

Entre estas soluciones se encuentra el paradigma de las líneas de productos *software* (Clements y Northrop, 2002; Gomaa, 2004). Este paradigma analiza el dominio de los productos que quieren desarrollarse para encontrar similitudes y diferencias entre ellos, de tal modo que puedan construirse productos a través de la composición de diversas características. Este enfoque se ajusta perfectamente al dominio de los paneles de control, dado que son herramientas que comparten varias propiedades.

En este trabajo se presentan y discuten los beneficios de la aplicación de este paradigma al dominio de los paneles de control, especialmente en contextos educativos, en los que diversos perfiles están involucrados. El objetivo principal es acercar la visualización de datos a cualquier usuario, así como fomentar la generación de conocimiento. En un contexto educativo, contar con paneles de control hechos a medida para cada actor involucrado podría significar una explotación más eficaz y efectiva de los datos de aprendizaje generados y obtenidos a través de diversas fuentes.

LOS PANELES DE CONTROL EN EL CONTEXTO EDUCATIVO

Como se ha mencionado en la introducción, los paneles de control son herramientas cada vez más populares, debido a su utilidad a la hora de soportar el análisis visual de conjuntos de datos complejos. El contexto educativo es uno de los contextos en los que estas herramientas pueden traer beneficios significativos, dado que el uso de datos para tomar decisiones relativas a los procesos educativos puede mejorar el aprendizaje (Cooper, 2007; Mandinach y Honey, 2008).

Los paneles de control educacionales (Yoo, Lee, Jo y Park, 2015) son instrumentos que permiten a sus usuarios identificar patrones, relaciones, datos relevantes, etc., entre un conjunto de variables de aprendizaje.

Sin embargo, en un contexto como el educativo, muchos roles pueden verse involucrados: desde los propios estudiantes, hasta profesores, jefes de estudio o directores, entre otros. Como es evidente, estos roles tendrán diversos objetivos a la hora de explorar sus datos, dependiendo de sus necesidades.

Esta diversidad de roles se analizó en una revisión de la literatura realizada por Schwendimann et al. (2017) en cuanto a los paneles de control educativos. En dicha revisión de la literatura se observó que la mayoría de los usuarios suelen ser profesores, pero también se encuentran estudiantes, administradores e investigadores entre los principales usuarios de estas herramientas (Schwendimann et al., 2017). Los paneles de control educativos también son diversos en lo que respecta a sus objetivos; monitorización propia, monitorización de otros estudiantes y monitorización administrativa (Schwendimann et al., 2017).

La mencionada revisión de la literatura también muestra los principales tipos de gráficos o visualizaciones utilizadas para mostrar información sobre el aprendizaje en función del rol del usuario. Así pues, los gráficos más utilizados en general por todos los roles son los diagramas de barras, los diagramas de líneas y las tablas, lo que puede deberse a su simplicidad.

Este tipo de investigaciones permiten observar que los paneles de control son muy diversos en el contexto educativo, tanto en sus funcionalidades como en su diseño, dado que estas características son las que definen el fin del instrumento (y su eficiencia).

Debido a estos factores, se han buscado métodos y realizado propuestas para diseñar paneles de control educativos y de analítica del aprendizaje para que puedan ser adaptados según sus propósitos y audiencia.

No existe un enfoque único que sirva para todos los usuarios (Teasley, 2017). En el contexto educativo, los paneles de control no solo buscan informar a tutores sobre el rendimiento de los alumnos, sino que también pueden convertirse en herramientas que motiven a estos últimos. Pueden incluso servir como instrumentos para que los estudiantes se auto-regulen y puedan comparar sus propios resultados. Sin embargo, puede que no todos los estudiantes respondan de la misma forma ante la información mostrada en un panel de control sobre su rendimiento (Teasley, 2017).

RIED. Revista Iberoamericana de Educación a Distancia (2020), 23(2), pp. 169-185.
DOI: http://dx.doi.org/10.5944/ried.23.2.26389 – ISSN: 1138-2783 – E-ISSN: 1390-3306

Los paneles de control deben personalizarse para ofrecer la información necesaria de la manera más efectiva. De hecho, en un estudio realizado por Roberts, Howell y Seaman (2017) se confirmó el deseo generalizado de los estudiantes por contar con paneles de control que puedan ser personalizados a su gusto, dándoles opción de configurarlos para mostrar la información que más les interesa o que ven más útil.

Así pues, no es solo la variedad de roles de usuarios en el contexto educativo, sino la variedad de objetivos y perfiles entre usuarios con un mismo rol, lo que convierte el desarrollo de paneles de control que presentan analíticas del aprendizaje en una actividad elaborada. Sumado a todo ello, la cantidad de datos generados y su compleja estructura pueden dificultar aún más el proceso de descubrimiento de conocimiento por parte de perfiles menos técnicos.

Por estas razones, se han propuesto modelos para intentar adaptar estas herramientas usando modelos conceptuales en los que se tienen en cuenta los indicadores, la descripción y necesidades de los usuarios, sus preferencias, su conocimiento del dominio, etc. De hecho, en Dabbebi, Iksal, Gilliot, May y Garlatti (2017) se presenta un generador de paneles de control de analítica del aprendizaje que tiene en cuenta la mencionada información. Esta información es estructurada en modelos que alimentan un generador que permite obtener paneles de control adaptados al perfil de usuario y tareas descritas en los modelos.

Como puede observarse, los paneles de control en el contexto educativo han incrementado su popularidad debido a los beneficios que puede suponer el uso de los datos en la toma de decisiones (Patil y Mason, 2015). Sin embargo, para sacarles partido, es necesario tener en cuenta a los usuarios y el contexto en el que serán empleados, y realizar pruebas de aceptación para comprobar que dichas herramientas pueden mejorar los procesos de aprendizaje.

METODOLOGÍA

Como se ha introducido, el desarrollo de paneles de control hechos a medida supone un reto, tanto a nivel de diseño como de implementación. Por ello, es necesario contar con paradigmas que potencien la productividad a la hora de desarrollar estas herramientas.

El paradigma de líneas de productos *software* permite identificar, a través de ingeniería de dominio y la abstracción de este, puntos comunes entre los sistemas que conforman el espacio de posibles productos. Esta identificación de similitudes no solo sirve para reutilizar código y disminuir los tiempos de implementación, sino también para identificar y estructurar factores importantes que afectan al diseño de los paneles de control.

Las líneas de producto *software*

Las líneas de productos *software* es uno de los paradigmas de reutilización sistemática de código más extendidos y aplicados en la práctica y en contextos reales. Es especialmente su viabilidad práctica lo que coloca esta metodología como una potente herramienta para abordar el desarrollo de software de manera masiva.

Este paradigma se compone de dos fases. La primera, denominada fase de ingeniería de dominio (Pohl, Böckle y Van der Linden, 2005), es clave para la creación de líneas de productos *software*. Durante esta fase se identifican las principales propiedades y características que tendrán los productos pertenecientes a la familia. Estas características representan las similitudes y diferencias entre los diversos productos del dominio. La identificación de propiedades abstractas del dominio permite formalizar los componentes o *core assets* necesarios para su implementación.

En esta fase también se definen los denominados puntos de variabilidad. Los puntos de variabilidad son la forma de introducir, modificar o adaptar ciertas funcionalidades según los requisitos del producto a desarrollar (Pohl, Böckle y Van der Linden, 2005). Esta tarea es esencial, ya que es la que permite materializar las características del dominio en componentes *software*. Existen diversos métodos para modelar los mencionados puntos de variabilidad (Metzger y Pohl, 2007; Van Gurp, Bosch y Svahnberg, 2001).

Uno de los mecanismos más extendidos para representar la variabilidad en las líneas de productos software es el método FODA (*Feature-Oriented Domain Analysis*) (Kang, Cohen, Hess, Novak y Peterson, 1990). Los diagramas de características (*feature models*) se introducen por primera vez en (Kang et al., 1990) como mecanismo formal de descripción de las propiedades del dominio. A través de estos diagramas jerárquicos pueden especificarse las características obligatorias, alternativas, opcionales, etc., de la familia de productos, o lo que es lo mismo, sus puntos de variabilidad.

Una vez estudiado el dominio e identificadas las características de la línea, la siguiente fase es la fase de ingeniería de aplicación. Haciendo uso de la información y recursos obtenidos en la fase anterior, se crean instancias concretas de la línea de productos, seleccionando las características que compondrán el producto específico a construir (Gomaa, 2004).

En función de la configuración seleccionada a través de los modelos instanciados, se configurarían y compondrían los componentes base (*core assets*) para implementar el producto final. Gracias a su implementación durante la fase de ingeniería del dominio, los componentes base estarían preparados para ser configurables, con lo cual solo sería necesario adaptarlos a los requisitos específicos, demostrando la potencia de este paradigma.

RIED. Revista Iberoamericana de Educación a Distancia (2020), 23(2), pp. 169-185.
DOI: http://dx.doi.org/10.5944/ried.23.2.26389 – ISSN: 1138-2783 – E-ISSN: 1390-3306

Generación de código

Existen diversas formas de inyectar variabilidad en los componentes base de la familia de productos. La generación o instanciación automática de productos de la familia podría verse como una combinación del paradigma de las líneas de productos y el desarrollo dirigido por modelos (Anquetil et al., 2008).

La división de los productos en una serie de componentes configurables con características definidas posibilita la creación de generadores de código que seleccionen y configuren acordemente los componentes base para obtener el código fuente del producto final.

Para conseguir esto, es necesario materializar los puntos de variabilidad obtenidos en la fase de ingeniería del dominio en el propio código fuente de los componentes base. El rol del generador de código consistiría en inyectar los parámetros de configuración específicos para adaptarlos a los requisitos del producto a generar.

Existen diversas técnicas para materializar estos puntos de variabilidad en el código fuente, por ejemplo: delegación de funcionalidades, herencia, parametrización, sobrecarga de métodos o funciones, bibliotecas de enlace dinámico (DLL) o compilación condicional (Gacek y Anastasopoules, 2001).

Las plantillas de código son otra técnica muy extendida para abordar este problema (Tajali, Corriveau y Shi, 2013). Éstas permiten definir partes estáticas y partes dinámicas en las que puede inyectarse código en función de una serie de reglas (Greifenberg et al., 2016; Magdalenić, Radošević y Kermek, 2011).

Para esta investigación se han utilizado plantillas como método de generación de código, debido a que es un método viable para las líneas de productos *software* (Greifenberg et al., 2016; Tajal, Corriveau y Shi, 2013), un método ampliamente utilizado en el desarrollo web (Kirda y Kerer, 2000; Tatsubori y Suzumura, 2009) y permiten la introducción de sentencias condicionales y bucles, haciendo sencilla la materialización de los puntos de variabilidad a partir de modelos. Además, las plantillas permiten lograr un grado de granularidad muy fino en lo que se refiere a la materialización de características, lo que es altamente beneficioso en un dominio tan complejo como el de los paneles de control (Vázquez-Ingelmo, García-Peñalvo y Therón, 2019a).

En cuanto a la generación de código, se ha utilizado la siguiente estructura: el generador de código toma la configuración del panel de control previamente especificada a través de ficheros de configuración XML (cuya sintaxis está basada en las características abstractas del dominio previamente identificadas), inyectando la información de configuración en las plantillas de código que, en este caso, conformarán los componentes base de la línea de productos de paneles de control.

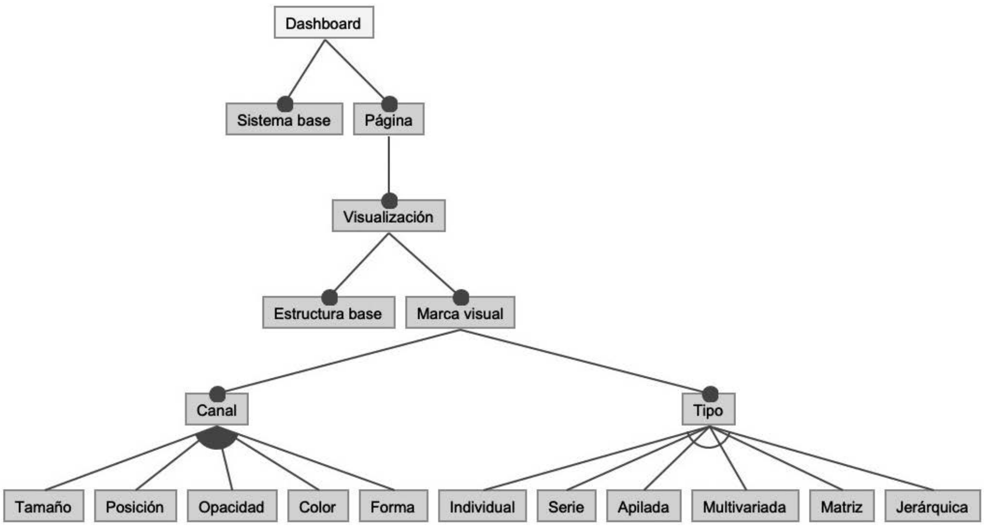
Las plantillas han sido realizadas con el lenguaje de plantillas Jinja2 (Ronacher, 2008). Este método es independiente al marco de desarrollo, y, aunque deba ser utilizado junto al lenguaje de programación Python, es posible inyectar información externa y generar cualquier tipo de archivos (Clark, 2018).

Cada componente gráfico de los paneles de control identificado mediante ingeniería de dominio ha sido implementado de manera individual, de tal forma que estos elementos primitivos puedan ser compuestos a través de plantillas Jinja2 para obtener visualizaciones concretas.

RESULTADOS PRELIMINARES

Gracias a la aplicación del enfoque de las líneas de productos *software* al dominio de los paneles de control, ha sido posible obtener una serie de componentes base (*core assets*) que pueden ser combinados para obtener paneles de control funcionales.

Figura 1. Diagrama de características simplificado de la línea de productos *software* de paneles de control



Como puede observarse en la figura 1, un panel de control estará compuesto por un sistema base y una o más páginas. Estas páginas contendrán una serie de visualizaciones, que contarán con una estructura base común y un conjunto de marcas visuales. Las marcas visuales pueden ser de distintos tipos según los datos que quieran representarse (marcas individuales, serie, apiladas, multivariadas, jerárquicas o matrices). Para codificar la información, se utilizarán uno o más canales, que se corresponden con propiedades de la marca visual, como su color, posición, tamaño, opacidad, forma, etc.

RIED. Revista Iberoamericana de Educación a Distancia (2020), 23(2), pp. 169-185.
DOI: http://dx.doi.org/10.5944/ried.23.2.26389 – ISSN: 1138-2783 – E-ISSN: 1390-3306

La figura 1 muestra una versión simplificada del diagrama con el objetivo de facilitar la comprensión del mismo, pero las visualizaciones contienen más elementos básicos, como sus ejes, leyendas, patrones de interacción, etc. (Vázquez-Ingelmo, García-Holgado, García-Peñalvo y Therón, 2019; Vázquez-Ingelmo, García-Peñalvo y Therón, 2019b; Vázquez-Ingelmo, García-Peñalvo, Therón y Conde González, 2019).

Estas pequeñas piezas abstractas mostradas en el diagrama de características tienen un grano fino de configuración, lo que permite personalizar el panel de control no solo a nivel global, sino también a nivel de visualización.

Para ilustrar la línea de productos obtenida mediante un ejemplo, se ha utilizado parte de los registros del conjunto de datos recogido en (Kuzilek, Hlosta y Zdrahal, 2017), el cual contiene datos demográficos, de las interacciones y del rendimiento de los estudiantes de Open University (OU).

Por ejemplo, a través de un fichero de configuración como el mostrado en la figura 2, es posible definir un panel de control concreto basado en las propiedades abstractas del dominio. Este fichero utiliza la tecnología XML para estructurar las características de los paneles de control a generar. Por ejemplo, el panel de control descrito en la figura 2 tendrá como título "*Learning Dashboard*", y contará con una página en la que se encuentra un componente de tipo visualización donde se mostrarán las puntuaciones de tareas por estudiante a través de marcas visuales de tipo "barra". Toda esta sintaxis está basada en el diagrama de características mostrado en la figura 1.

Figura 2. Fragmento de un fichero de configuración utilizado para generar un panel de control

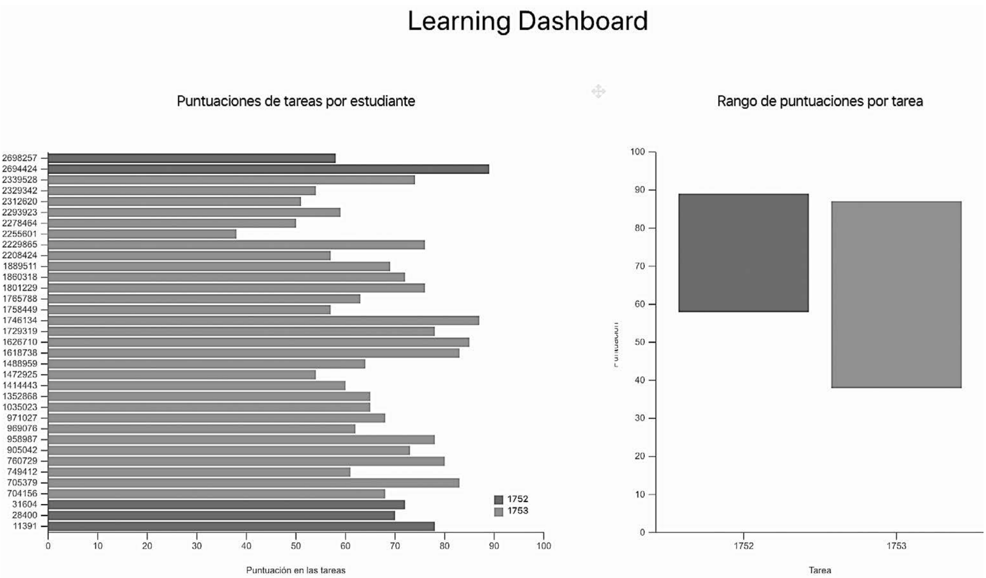
```
<Dashboard>
    <Title>Learning Dashboard</Title>
    <Dataset>
        <Path>studentAssessment.csv</Path>
        <Type>CSV</Type>
    </Dataset>
    <Page page_id="1">
            <Component type="visualization" component_id="1">
                <Position>
                    <x>0</x>
                    <y>0</y>
                    <width>5</width>
                    <height>8</height>
                </Position>
                <Title>Puntuaciones de tareas por estudiante</Title>
                <Primitives>
                    <Axis type="y" linear="true" multi="false">
                        <Scale>
                            <Accessor>id_student</Accessor>
                        </Scale>
                    </Axis>
                    <Axis type="x" linear="true" multi="false">
                        <Scale>
                            <Accessor>score</Accessor>
                            <Domain>percentage</Domain>
                        </Scale>
                        <Label>
                            Puntuación en las tareas
                        </Label>
                    </Axis>
                    <Legend>
                        <Scale>
                            <Accessor>id_assessment</Accessor>
                            <ScaleType>band</ScaleType>
                        </Scale>
                        <Position>bottom-right</Position>
                    </Legend>
                    <Mark type="individual" linear="true">
                        <Shape>bar</Shape>
                        <Channels>
                            <Position_Y>
                                <Accessor>id_student</Accessor>
                            </Position_Y>
                            <Position_X>
                                <Accessor>score</Accessor>
                                <Domain>percentage</Domain>
                            </Position_X>
                            <Color>
                                <Accessor>id_assessment</Accessor>
                            </Color>
                        </Channels>
```


DOI: http://dx.doi.org/10.5944/ried.23.2.26389 – ISSN: 1138-2783 – E-ISSN: 1390-3306

Una vez definidas las características del panel de control y de sus componentes (visualizaciones, canales de codificación, estilos), el fichero de configuración es utilizado por el generador de código para instanciar el panel de control a través de la inyección en las plantillas de código de los parámetros especificados. El resultado son los ficheros fuente del panel de control con las características establecidas (figura 3).

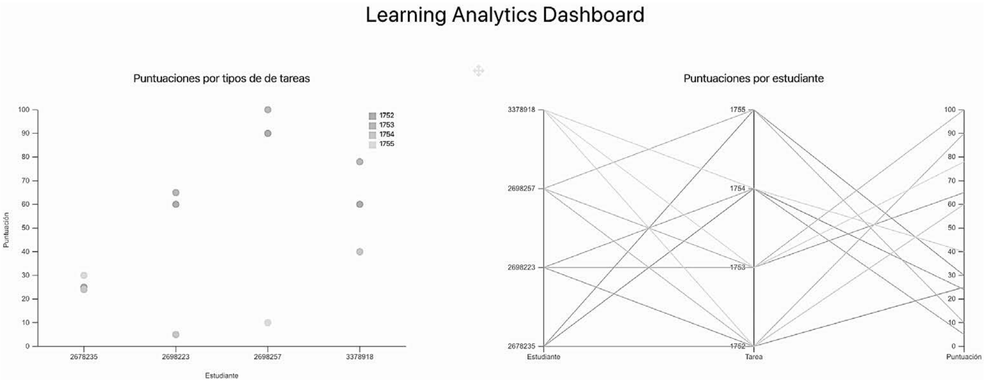
En este caso, el panel de control contiene dos visualizaciones que representan las puntuaciones de las tareas realizadas por cada estudiante y el rango de puntuaciones (mínimo y máximo) por tarea, respectivamente.

Figura 3. Ejemplo de un panel de control generado a través de su definición XML



Cambiar el tipo de gráfico o metáforas visuales para mostrar los datos en una visualización es tan sencillo como modificar sus primitivas en el fichero de configuración, lo que permite probar nuevas estructuras o composiciones para el panel de control sin preocuparse de su implementación. Por ejemplo, la figura 4 muestra otra composición para el panel de control utilizando un diagrama de dispersión y unas coordenadas paralelas para codificar la información relativa a las puntuaciones de las tareas de los estudiantes.

Figura 4. Ejemplo de un segundo panel de control generado a través de otra definición XML



DISCUSIÓN

La creación de paneles de control es una tarea compleja, sea cual sea el contexto en el que estén enmarcados, dado que la generación de conocimiento es crucial para mejorar los procesos de negocio. En lo que se refiere al ámbito educativo, este tipo de herramientas se vuelven especialmente útiles para generar conocimiento respecto al rendimiento de los estudiantes o la efectividad de los métodos de enseñanza utilizados.

Utilizar métodos para disminuir los tiempos de desarrollo de los paneles de control no solo reporta beneficios a la hora de crear estos instrumentos, sino que, además, permite generar paneles de control personalizados con menor esfuerzo, pudiéndose dedicar mayor tiempo y recursos a la fase de elicitación de requisitos.

La elicitación de requisitos es una fase esencial en cualquier proceso de desarrollo *software*, pero en el caso de los paneles de control toma especial relevancia, debido a que una mala interpretación de estos puede comprometer la efectividad de estas herramientas. Contar con un robusto sistema de generación de paneles de control facilita la realización de pruebas con usuarios (dado que pueden crearse prototipos rápidamente) y permite ejecutar cambios en la configuración de estos sin consumir muchos recursos.

Como se ha mencionado anteriormente, el contexto educativo puede ser un claro beneficiario de la aplicación de las líneas de productos *software* al dominio de los paneles de control. La cantidad de datos de aprendizaje que son generados debido a la popularización de nuevas tecnologías en la educación (Ferguson, 2012) hace necesario contar con nuevos métodos e instrumentos que permitan obtener beneficios de dicha información.

Si bien los paneles de control son herramientas muy útiles para estos procesos de análisis, es necesario tener en cuenta la audiencia que los utilizará, sobre todo

RIED. Revista Iberoamericana de Educación a Distancia (2020), 23(2), pp. 169-185.
DOI: http://dx.doi.org/10.5944/ried.23.2.26389 – ISSN: 1138-2783 – E-ISSN: 1390-3306

en ámbitos educativos donde los roles y los perfiles de usuario pueden ser muy heterogéneos en cuanto a objetivos, características y preferencias (Schwendimann et al., 2017).

La analítica del aprendizaje busca la mejora de los procesos de aprendizaje desde diversos puntos de vista (Long y Siemens, 2011); mientras que los educadores pueden beneficiarse de los datos para mejorar sus planificaciones y metodologías, los estudiantes pueden ver su progreso e incluso motivarse a través de los datos recolectados. Por otro lado, a más alto nivel, los administradores pueden utilizar la información para asignar recursos o planificar presupuestos de una forma más informada (Long y Siemens, 2011).

Poder generar paneles de control de manera sencilla, dedicando más tiempo al diseño y conceptualización del panel de control que a su implementación, permite tener productos mejor diseñados y adaptados a situaciones concretas en menor tiempo, así como responder de manera rápida y adecuada a la continua evolución que presentan los datos de aprendizaje (Liñán y Pérez, 2015; Picciano, 2012).

A su vez, contar con un generador de código permite realizar un análisis previo de los datos que permite inferir ciertas características de los paneles de control antes de ser generados, asegurándose de una correcta elección de las propiedades visuales antes de obtener el código fuente (por ejemplo, una variable de tipo nominal no puede codificarse usando una escala cuantitativa). De esta forma, es posible descargar a los usuarios de tareas más técnicas, como elegir las escalas o dominios a utilizar para representar una determinada variable.

Sin embargo, este enfoque sigue requiriendo de un riguroso análisis de los requisitos y perfiles de cada usuario, puesto que, de lo contrario, el panel de control generado puede resultar poco útil o efectivo para sus propósitos.

Uno de los siguientes pasos para conseguir un sistema más potente es implementar la capacidad de inferir automáticamente las características de los paneles de control a partir del perfil del usuario que lo empleará. De este modo, podrían presentarse gráficos o metáforas visuales más simples a un profesor con baja alfabetización en visualización de datos, mientras que un usuario más avanzado podría contar con visualizaciones más complejas en su pantalla, siempre de acuerdo con las tareas y objetivos que éstos tengan respecto a sus datos.

La toma de decisiones dirigida por datos puede llegar a ser muy útil, pero es necesario tener siempre en cuenta el contexto de aplicación y comprobar de manera continua que la implantación de estas herramientas y metodologías realmente funciona y trae beneficios a las personas involucradas.

CONCLUSIONES

El paradigma de las líneas de productos *software* ha sido aplicado al dominio de los paneles de control de información. El principal propósito de esta aplicación es disminuir la complejidad del proceso de diseño y creación de estas herramientas.

Mediante la abstracción del dominio, ha sido posible obtener una serie de componentes abstractos parametrizables que pueden ser combinados para generar paneles de control específicos con menor consumo de recursos.

Este enfoque puede aplicarse a cualquier contexto, en especial el educativo, donde el análisis de datos del aprendizaje se está popularizando de cara a mejorar los procesos y métodos de aprendizaje. Ofrecer paneles de control personalizados a los actores involucrados en estos procesos podría facilitar y acercar el análisis de datos a cualquier perfil de usuario.

Las líneas de investigación futuras incluyen la realización de pruebas con usuarios para obtener información sobre qué tipos de visualizaciones son más efectivas en función del contexto de los datos y las características de los usuarios, para incorporar estos datos al generador de código e inferir configuraciones efectivas automáticamente.

AGRADECIMIENTOS

Esta investigación ha sido parcialmente financiada por el Ministro de Economía y Competitividad del Gobierno de España a través del Proyecto DEFINES (TIN2016-80172-R). Este trabajo ha sido apoyado por el Ministerio de Educación, Cultura y Deporte bajo una beca FPU (FPU17/03276).

REFERENCIAS

Albright, S. C., Winston, W., y Zappe, C. (2010). *Data analysis and decision making*. Cengage Learning.

Aldrich, F., y Sheppard, L. (2000). Graphicacy-the fourth'R'? *Primary Science Review*, *64*, 8-11.

Anquetil, N., Grammel, B., Galvão, I., Noppen, J., Khan, S. S., Arboleda, H., . . . Garcia, A. (2008). Traceability for model driven, software product line engineering. Ponencia presentada en ECMDA Traceability Workshop Proceedings.

Clark, S. (2018). Render your first network configuration template using Python and Jinja2. *Cisco Blogs*. Recuperado de https://blogs.cisco.com/developer/network-configuration-template

Clements, P., y Northrop, L. (2002). *Software product lines*. Addison-Wesley.

Cooper, M. M. (2007). Data-driven education research. *Science, 317*(5842), 1168-1171. doi: 10.1126/science.317.5842.1171.

Dabbebi, I., Iksal, S., Gilliot, J.-M., May, M., y Garlatti, S. (2017). Towards Adaptive Dashboards for Learning Analytic: An Approach for Conceptual Design and implementation. Ponencia presentada en *9th International Conference on Computer Supported Education (CSEDU 2017)*, Porto, Portugal, (pp. 120-131). doi: 10.5220/0006325601200131.

Ferguson, R. (2012). Learning analytics: Drivers, developments and challenges. *International Journal of Technology Enhanced Learning,* 4(5/6), 304-317. doi: 10.1504/IJTEL.2012.051816.

Gacek, C., y Anastasopoules, M. (2001). Implementing product line variabilities. *Ponencia presentada en ACM SIGSOFT Software Engineering Notes,* New

York, NY, USA. 26, (pp. 109-117). doi: 10.1145/375212.375269.

Gomaa, H. (2004). *Designing Software Product Lines with UML: From Use Cases to Pattern-Based Software Architectures*. Addison Wesley Longman Publishing Co., Inc.

Greifenberg, T., Müller, K., Roth, A., Rumpe, B., Schulze, C., y Wortmann, A. (2016). Modeling variability in template-based code generators for product line engineering. *arXiv preprint arXiv:1606.02903*.

Kang, K. C., Cohen, S. G., Hess, J. A., Novak, W. E., y Peterson, A. S. (1990). *Feature-oriented domain analysis (FODA) feasibility study*. Tech. rep. CMU/SEI-90-TR-21. Software Engineering Institute, Carnegie Mellon University.

Kintz, M., Kochanowski, M., y Koetter, F. (2017). Creating User-specific Business Process Monitoring Dashboards with a Model-driven Approach. *Ponencia presentada en MODELSWARD. SCITEPRESS,* (pp. 353-361). doi: 10.5220/0006135203530361

Kirda, E., y Kerer, C. (2000). MyXML: An XML based template engine for the generation of flexible Web content. *Ponencia presentada en WebNet World Conference on the WWW and Internet*, (pp. 317-322). Association for the Advancement of Computing in Education (AACE).

Kuzilek, J., Hlosta, M., y Zdrahal, Z. (2017). Open university learning analytics dataset. *Scientific data, 4*(170171), 1-8. doi: 10.1038/sdata.2017.171.

Liñán, L. C., y Pérez, Á. A. J. (2015). Educational Data Mining and Learning Analytics: differences, similarities, and time evolution. *International Journal of Educational Technology in Higher Education, 12*(3), 98-112. doi: 10.7238/rusc.v12i3.2515.

Logre, I., Mosser, S., Collet, P., y Riveill, M. (2014). *Sensor data visualisation: a composition-based approach to support domain variability*. Ponencia presentada en European Conference on Modelling Foundations and Applications. 101-116. doi: 10.1007/978-3-319-09195-2_7.

Long, P. D., y Siemens, G. (2011). Penetrating the Fog: Analytics in Learning and Education. *Educause Review, 46*(5), 30-32. doi: 10.17471/2499-4324/195.

Magdalenić, I., Radošević, D., y Kermek, D. (2011). Implementation Model of Source Code Generator. *Journal of Communications and Software Systems*. 7(2). 71-79. Doi: 10.24138/jcomss.v7i2.180.

Mandinach, E. B., y Honey, M. (2008). *Data-driven school improvement: Linking data and learning*. Teachers College Press.

Metzger, A., y Pohl, K. (2007). Variability management in software product line engineering. *Ponencia presentada en the Companion to the proceedings of the 29th International Conference on Software Engineering*, (pp. 186-187). doi: 10.1109/ICSECOMPANION.2007.83.

Munzner, T. (2014). *Visualization analysis and design*. AK Peters/CRC Press.

Palpanas, T., Chowdhary, P., Mihaila, G., y Pinel, F. (2007). Integrated model-driven dashboard development. *Information Systems Frontiers, 9*(2-3), 195-208. doi: 10.1007/s10796-007-9032-9.

Patil, D., y Mason, H. (2015). *Data Driven*: " O'Reilly Media, Inc.". ISBN: 9781491925454.

Picciano, A. G. (2012). The evolution of big data and learning analytics in American higher education. *Journal of Asynchronous Learning Networks, 16*(3), 9-20. doi: 10.24059/olj.v16i3.267

Pleuss, A., Wollny, S., y Botterweck, G. (2013). Model-driven development and evolution of customized user interfaces. *Ponencia presentada en Proceedings of the 5th ACM SIGCHI symposium on Engineering interactive computing systems*, (pp. 13-22). doi: 10.1145/2494603.2480298.

Pohl, K., Böckle, G. y Van der Linden, F. J. (2005). *Software Product Line Engineering: Foundations, Principles and Techniques*. Springer-Verlag New York, Inc.

Roberts, L. D., Howell, J. A., y Seaman, K. (2017). Give me a customizable dashboard: Personalized learning analytics dashboards in higher education. *Technology, Knowledge and Learning, 22*(3), 317-333. doi: 10.1007/s10758-017-9316-1.

Ronacher, A. (2008). *Jinja2 Documentation*. Recuperado de http://jinja.pocoo.org/docs/2.10/:

Sarikaya, A., Correll, M., Bartram, L., Tory, M., y Fisher, D. (2018). What Do We Talk About When We Talk About Dashboards? *IEEE Transactions on Visualization Computer Graphics, 25*(1), 682 - 692. doi: 10.1109/TVCG.2018.2864903.

Schwendimann, B. A., Rodriguez-Triana, M. J., Vozniuk, A., Prieto, L. P., Boroujeni, M. S., Holzer, A., . . . Dillenbourg, P. (2017). Perceiving learning at a glance: A systematic literature review of learning dashboard research. *IEEE Transactions on Learning Technologies, 10*(1), 30-41. doi: 10.1109/TLT.2016.2599522

Tajali, S. B., Corriveau, J.-P., y Shi, W. (2013). A Template-Based Approach to Modeling Variability. *Paper presented at the Proceedings of the International Conference on Software Engineering Research and Practice* (SERP).

Tatsubori, M., y Suzumura, T. (2009). HTML templates that fly: a template engine approach to automated offloading from server to client. *Actas de 18th international conference on World wide web*, (pp. 951-960). doi: 10.1145/1526709.1526837.

Teasley, S. D. (2017). Student facing dashboards: One size fits all? *Technology, Knowledge and Learning, 22*(3), 377-384. doi: 10.1007/s10758-017-9314-3.

Van Gurp, J., Bosch, J., y Svahnberg, M. (2001). On the notion of variability in software product lines. Ponencia presentada en Software Architecture, 2001. *Proceedings. Working IEEE/IFIP Conference on Software Architecture*, Amsterdam, Netherlands, (pp. 45-54). doi: 10.1109/WICSA.2001.948406.

Vázquez-Ingelmo, A., García-Holgado, A., García-Peñalvo, F. J., y Therón, R. (2019). *Dashboard Meta-Model for Knowledge Management in Technological Ecosystem: A Case Study in Healthcare*. Ponencia presentada en the UCAmI 2019, Toledo, Castilla-La Mancha, Spain. *MDPI Proceedings 2019, 31*(1), 44, 1-13. doi: 10.3390/proceedings2019031044.

Vázquez-Ingelmo, A., García-Peñalvo, F. J., y Therón, R. (2019a). Addressing Fine-Grained Variability in User-Centered Software Product Lines: A Case Study on Dashboards. *Ponencia presentada en World Conference on Information Systems and Technologies. Advances in Intelligent Systems and Computing*, vol 930, (pp. 855-864). doi: 10.1007/978-3-030-16181-1_80.

Vázquez-Ingelmo, A., García-Peñalvo, F. J., y Therón, R. (2019b). Capturing high-level requirements of information dashboards' components through meta-modeling. *Ponencia presentada en 7th International Conference on Technological Ecosystems for Enhancing Multiculturality (TEEM 2019)*, León, Spain, 815-821, ACM. doi: 10.1145/3362789.3362837.

Vázquez-Ingelmo, A., García-Peñalvo, F. J., y Therón, R. (2019c). Information Dashboards and Tailoring Capabilities - A Systematic Literature Review. *IEEE Access, 7,* 109673-109688. doi: 10.1109/ACCESS.2019.2933472.

Vázquez-Ingelmo, A., García-Peñalvo, F. J., Therón, R., y Conde González, M. Á. (2019). Extending a dashboard meta-model to account for users' characteristics and goals for enhancing personalization. *Ponencia presentada en Learning Analytics Summer Institute (LASI) Spain*

2019, Vigo, Spain. CEUR Workshop Proceedings Series, no. 2415, (pp. 35-42). Recuperado de http://ceur-ws.org/Vol-2415/paper04.pdf

Yoo, Y., Lee, H., Jo, I.-H., y Park, Y. (2015). Educational dashboards for smart learning: Review of case studies. *Emerging issues in smart learning* (pp. 145-155): Springer.

PERFIL ACADÉMICO Y PROFESIONAL DE LOS AUTORES

Andrea Vázquez-Ingelmo. Investigadora en el grupo de investigación GRIAL. Graduada en Ingeniera Informática (2016, Universidad de Salamanca, España) y Máster en Ingeniería Informática (2018) por la misma universidad. Sus líneas de trabajo se centran en la visualización de datos y la ingeniería del software. ORCID: https://orcid.org/0000-0002-7284-5593
E-mail: andreavazquez@usal.es

Roberto Therón. Profesor Titular de Universidad del Departamento de Informática y Automática en la Universidad de Salamanca (USAL). Es encargado del grupo VisUsal (dentro del Grupo de Investigación GRIAL), el cual se centra en la combinación de enfoques procedentes de la Informática, Estadística, Diseño Gráfico y Visualización de Información, para obtener una adecuada comprensión de conjuntos de datos complejos. En los últimos años, se ha dedicado al desarrollo de herramientas de visualización avanzada para datos multidimensionales, como por ejemplo datos genéticos o paeloclimáticos. En el área de Analítica Visual desarrolla productivas colaboraciones con grupos e instituciones de reconocido prestigio internacional, como el Laboratorio de Ciencias del Clima y del Medio Ambiente (París) o el Centro de Analítica Visual Avanzada de la ONU (Suiza). ORCID: https://orcid.org/0000-0001-6739-8875
E-mail: theron@usal.es

Dirección:
Facultad de Ciencias
Universidad de Salamanca
Plaza de los caídos, s/n
Salamanca (España)

Fecha de recepción del artículo: 14/01/2020
Fecha de aceptación del artículo: 02/03/2020
Fecha de aprobación para maquetación: 15/03/2020

7.23 Appendix W. Towards a Technological Ecosystem to Provide Information Dashboards as a Service: A Dynamic Proposal for Supplying Dashboards Adapted to Specific Scenarios

applied sciences

MDPI

Article

Towards a Technological Ecosystem to Provide Information Dashboards as a Service: A Dynamic Proposal for Supplying Dashboards Adapted to Specific Scenarios

Andrea Vázquez-Ingelmo [1,*], Francisco José García-Peñalvo [1] and Roberto Therón [2]

[1] GRIAL Research Group, Computer Science Department, University of Salamanca, 37008 Salamanca, Spain; fgarcia@usal.es
[2] VisUSAL, Computer Science Department, University of Salamanca, 37008 Salamanca, Spain; theron@usal.es
* Correspondence: andreavazquez@usal.es; Tel.: +34-923294500 (ext. 3433)

Abstract: Data are crucial to improve decision-making and obtain greater benefits in any type of activity. However, the large amount of information generated by new technologies has made data analysis and knowledge generation a complex task. Numerous tools have emerged to facilitate this generation of knowledge, such as dashboards. Although dashboards are useful tools, their effectiveness can be affected by poor design or by not taking into account the context in which they are placed. Therefore, it is necessary to design and create custom dashboards according to the audience and data domain. This paper presents an application of the software product line paradigm and the integration of this approach into a web service to allow users to request source code for customized information dashboards. The main goal is to introduce the idea of creating a holistic ecosystem of different services to craft and integrate information visualizations in a variety of contexts. One of the contexts that can be especially favored by this approach is the educational context, where learning analytics, data analysis of student performance, and didactic tools are becoming very relevant. Three different use cases of this approach are presented to illustrate the benefits of the developed generative service.

Keywords: information dashboards; metamodeling; visualization goals; visualization tasks; data visualization; dashboard ecosystem; code generation

Citation: Vázquez-Ingelmo, A.; García-Peñalvo, F.J.; Therón, R. Towards a Technological Ecosystem to Provide Information Dashboards as a Service: A Dynamic Proposal for Supplying Dashboards Adapted to Specific Scenarios. *Appl. Sci.* **2021**, *11*, 3249. https://doi.org/10.3390/app11073249

Academic Editor: Arcangelo Castiglione

Received: 4 March 2021
Accepted: 3 April 2021
Published: 5 April 2021

1. Introduction

Information dashboards are very powerful tools. They not only support the understanding of complex datasets but also are applicable to a variety of contexts and data domains. In addition, information dashboards provide support to learn from data and can also be considered educational tools [1].

However, adapting these tools to different contexts is a compelling task because it requires the study of the data domain and the audience that will be using the dashboard to discover knowledge. It is this complexity that makes the development of dashboards a time-consuming process. That is why reducing the development time of these tools is a crucial factor in tackling the continuous and exponential generation of data.

Having a dashboard ready to use in any context is beneficial for exploiting data and learning from them, with the goal of supporting better-informed decision-making processes. For these reasons, this article presents an ecosystem proposal to manage the generation of information dashboards that can be tailored attending to fine-grained features.

Although the end goal is the straightforward generation of information dashboards, this objective can be broken down into different low-level tasks, such as cleaning data, selecting the right data encodings or dashboard configurations, generating source code, etc. That is why a technological ecosystem approach can be applicable to this situation. Technological ecosystems provide a context in which different services are connected through information flows, although they can be seen as independent components or tools [2,3]. The "ecosystem" metaphor transfers properties from the biological to the technology field, so the relationships among the organisms can be seen as information flows between the technological ecosystem's components. On the other hand, the physical environment can be seen as the mechanisms or methods to support these flows [2].

For these reasons, applying an ecosystem approach to this matter benefits users by providing access to a whole generative pipeline for information dashboards, with components that interact and collaborate among them to offer powerful features, but also to independent services for more specific tasks. In fact, this ecosystem would support knowledge management, because all the tacit knowledge associated with dashboard design processes, design decisions, data transformations, etc., would be managed by the different components.

Besides, the proposed ecosystem's features rely on a dashboard meta-model that accounts for other factors that determine the dashboard design process: the users' characteristics, the data domain, the potential data context, etc.

Several data domains could benefit from these kinds of services—specifically, data domains in which the variety of data sources and the heterogeneity of data is determinant. The educational context is one of these domains.

Educational dashboards [4] are instruments that allow their users to identify patterns, relationships, relevant data, etc., among a set of learning variables [5].

However, in a context such as education, many roles can be involved: from the students themselves to teachers, heads of studies, or principals; and these roles will have different objectives when exploring their data, depending on their needs.

This diversity of roles was analyzed in a literature review conducted by [6] regarding educational dashboards. While the majority of users are usually teachers, students, administrators and researchers are also among the primary users of these tools. Educational dashboards are also diverse in terms of their objectives; self-monitoring, monitoring of other students, and administrative monitoring [6].

The abovementioned literature review also shows the main types of charts or visualizations used to display learning information according to the user's role. Thus, the most commonly used graphics in general by all roles are bar charts, line charts, and tables.

This type of research allows us to observe that dashboards are very diverse in the educational context, both in their functionalities and in their design, since these characteristics are what define the purpose of the instrument (and its efficiency). Due to these factors, methods have been sought, and proposals made to design educational and learning analytics dashboards so that they can be adapted according to their purposes and audience, because there is no one-size-fits-all approach [7]. In the educational context, dashboards not only seek to inform tutors about student performance but can also become tools to motivate students. They can even serve as tools for students to self-regulate and compare their own results. However, not all students may respond in the same way to the information shown on a dashboard about their performance [7].

Thus, it is not only the variety of user roles in the educational context but the variety of objectives and profiles among users with the same role, which makes the development of dashboards that present learning analysis an elaborate activity. In addition, the amount of data generated and its complex structure can make the process of knowledge discovery even more difficult for less technical profiles.

As can be seen, dashboards in the educational context have increased in popularity due to the benefits that their use can bring. However, to take advantage of them, it is necessary to take into account the users and the context in which they will be used.

However, not only dashboards that show learning variables can be found in the educational context. As these kinds of tools provide an important means to understand data and extract information and knowledge from them, they are also employed as didactic tools for motivation and learning [1], adding more complexity to the domain.

For all these reasons, the present work describes a proposal to create a technological ecosystem for dynamically tailoring dashboards no matter the data context or the data domain. Specifically, this paper is focused on illustrating how dashboards can be generated through a web service, providing different use cases within the context of developing dashboards for educational purposes [1]. To sum up, we pose the following research question:

RQ1. Is a technological ecosystem approach applicable to provide dashboards in different contexts and data domains?

The rest of this paper is organized as follows. Section 2 contains background regarding the automatic generation of dashboards and visualizations. Section 3 describes the methodology followed throughout this work. Section 4 presents the dashboard generator service architecture, and Section 5 illustrates the application and integration of these services to generate dashboards with different purposes. Finally, Sections 6 and 7 discuss the results and outline the conclusions obtained through this work.

2. Background

The automatic generation and design of dashboards is a popular research topic, given its potential benefits for exploiting datasets. This generative process can be pursued through different methodologies and paradigms.

There are a variety of methods to tackle a generative approach when developing these tools [8]. One of the most common methods for customizing dashboards is using configuration wizards that support the users' decisions when developing dashboards without requiring programming skills. For example, [9–12] use graphical user interfaces that assist the selection of widgets to be included in the dashboard. Configuration wizards could be complemented with visual mapping methods to assist the users in the selection of visualization types taking into account the data types or structure [13–16].

On the other hand, another common method to generate dashboards is to configure them by using structured configuration files [17–19]. These files allow users to select the dashboard components and visualizations through higher levels of abstraction, maintaining a structured representation of the generated tool.

Some works also take advantage of software engineering methodologies such as the Software Product Line (SPL) paradigm [20,21] or Model-Driven Development (MDD) [22–24]. These methodologies are focused on the abstraction of features within a domain to reduce development times and increase flexibility and adaptability when generating final products.

Other methods also include agents [25,26], inclusive user modeling [27], semantic reasoners [28], and knowledge graphs and ontologies [29].

Pursuing generative approaches when developing information dashboards has several advantages, such as the decrease of development time. However, another benefit is that these approaches are mainly based on configuration files or models, providing structured data regarding the dashboards' features.

Materializing the dashboards' features (which are often expressed in unstructured requirement documentation) in a structured manner provides a high-level layer to specify dashboards and the possibility to create services that use these structured definitions programmatically.

3. Materials and Methods

3.1. Metamodeling

Metamodeling is the backbone methodology from the model-driven development (MDD) paradigm [30,31]. This paradigm enables the abstraction of the systems' development process's requirements, providing support for moving both data and operations specifications away from lower-level details.

By abstracting these details, it can be possible to obtain a generic "skeleton" of information systems, containing the main structures and relationships among its high-level components. Meta-models are very useful resources to understand the systems' domain because they ease the identification process of relevant features within the context, separating these features from technical details or specific technologies.

This methodology increases the reusability of components (thus, decreasing the development time) and the reusability of knowledge because the structures and relationships identified within the systems' domain can evolve to obtain better solutions when instantiating the meta-model.

The MDD approach can be implemented through the model-driven architecture (MDA), a guideline proposed by the Object Management Group (OMG). This guideline provides an architecture for software development driven by models describing and defining the target system [32]. The OMG proposal also determines a set of standards to develop the approach, such as meta-object facility (MOF), unified modeling language (UML), XML (Extensible Markup Language), metadata interchange (XMI), and query/view/transformation (QVT).

In this case, the proposed dashboard meta-model is part of this meta-model architecture proposal [33]. Although the first version of the dashboard meta-model [34,35] was an instance of MOF, it was finally transformed into an instance of Ecore [36] using Graphical Modelling for Ecore included in Eclipse Modeling Framework (EMF) (Figure 1).

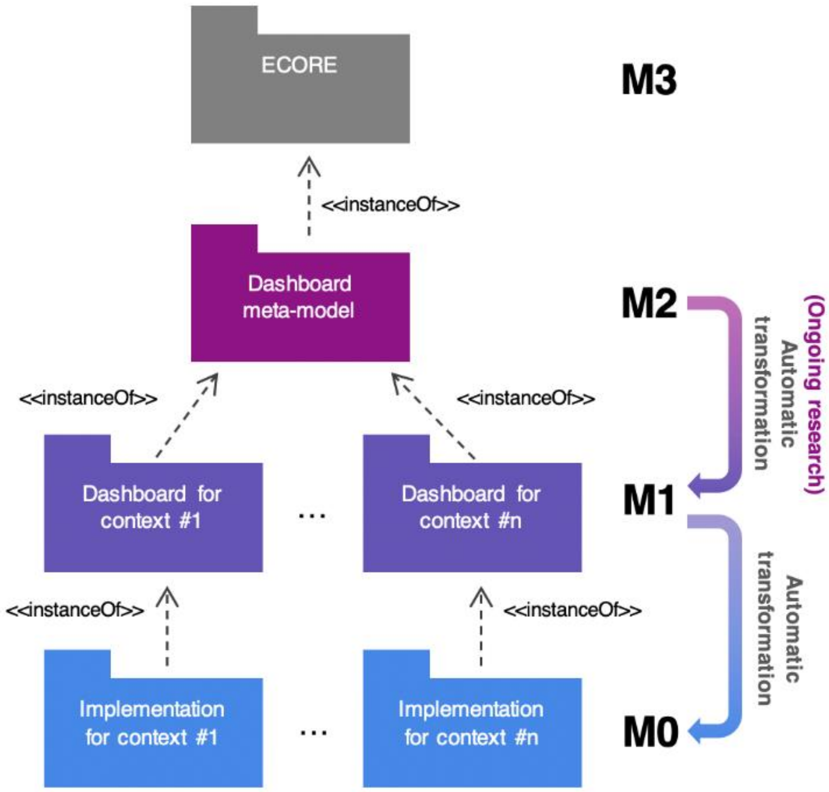


Figure 1. Location of the dashboard meta-model following the Object Management Group (OMG) architecture. The dashboard generator is in charge of transforming the meta-models' instances (M1 models) in specific dashboard implementations.

The transformations between the M2 and M1 levels are currently performed through manual processes: it is necessary to select and define the dashboard configuration manually; however, we are working on automating this process through artificial intelligence (AI) approaches. In [37], we explore this AI-based automation theoretically, and we plan to connect this approach with the M1-to-M0 transformations.

The M2 meta-model provides the basis to instantiate dashboards in different contexts and domains, and these models will be ultimately implemented as specific dashboards. This transition between this M1 model and the M0 model can be done automatically through the dashboard generator service, which complies with the M2 meta-model's structure and features.

As seen in the dashboard meta-model, these tools are composed of different sections or aspects, such as the layout, operations, visual marks and even the audience characterization. To adapt this model to an ecosystem proposal, we have divided the sections of the meta-model into the main tasks that can be found during a dashboard design process (user characterization, data transformations, data encoding and visualization design). Separating these phases into services could support the definition of a dashboard design pipeline.

3.2. Code Templates

Although the meta-model can be used as a conceptual resource for driving the development of dashboards, it can also have practical implications in this process. In the end, the meta-model is a structured set of elements and relationships that can be represented in different formats. One of these formats is XMI (XML-based Metadata Interchange), which can be quickly processed and converted into other formats, such as JSON objects.

By using a Python generator and an SPL [38–40] development, it has been possible to build a set of core software assets that can be combined into fully functional dashboards following the meta-model instance specification.

An example code template, configuration file, and rendered HTML code can be found at https://github.com/AndVazquez/generation-workflow-example (accessed on 4 April 2021) for further details.

3.3. Code as a Service

The possibility of automatically generating information dashboards by providing an external configuration enables providing this functionality as a service. The Python generator can be easily integrated into a web app (through the Django framework [41] and the Django REST Framework (DRF) module) to accept external requests containing the configuration of a visualization.

Specifically, this web service takes a JSON object as an input (containing the configuration of the visualization), and the user receives the HTML and JavaScript source code of the requested visualization.

4. Architecture Proposal

In this section, the architecture proposal for exploiting the previously explained framework will be detailed. As introduced before, one of the goals of applying this architecture is to obtain an ecosystem for generating and providing information visualizations as a service.

The ecosystem is planned to be a holistic set of well-defined components that provide unitary services, but that can also be combined to obtain a complete pipeline. Every service has well-defined interfaces that enable the connection of information flows among them. These services provide support for the generation of information dashboards that compile with the previously presented meta-model.

One of these services is the dashboard generator (Figure 2), based on plain JavaScript through the D3.js framework to allow better integration with external services avoiding other dependencies. The dashboard generator service accepts HTML requests containing information about the visualization component to craft. Specifically, this service is developed as an API in which the input is a JSON object with the configuration of an entire dashboard or a single visualization:

- Information about the dataset or datasets to be displayed. Data sources could be external APIs or files.
- The disposition or layout of the elements.
- The features of the visualization:
    - Number and type (X position, Y position, size, color, etc.) of visual channels;
    - Visual mark type (bar, circle, topographic, arc, etc.);
    - Dataset's variables to be represented;
    - Interaction events and effects [42].

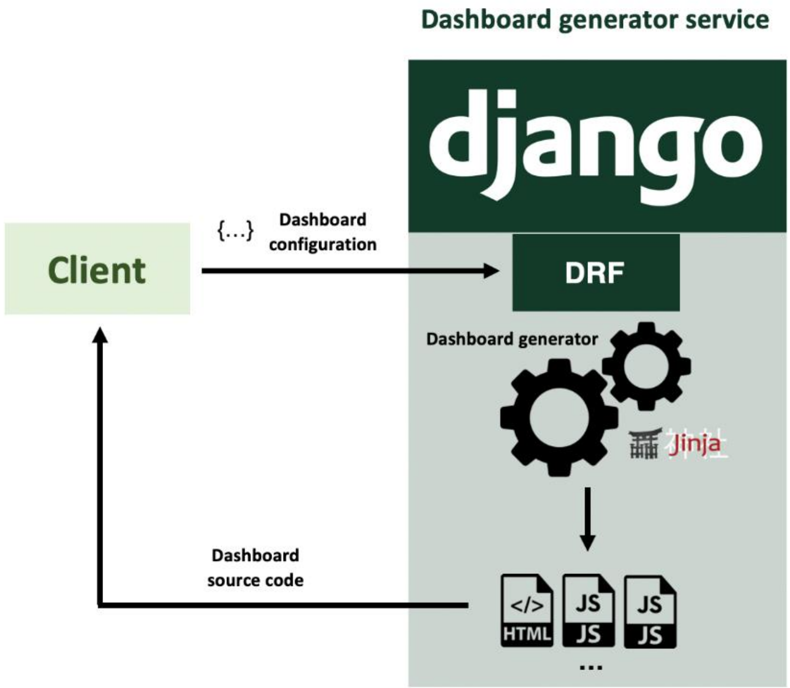


Figure 2. Schematic view of the dashboard generator service architecture.

The service processes this JSON object; then, the source code is generated using the previous section's code templates. These returned source code files are returned to the client, which could embed them in its own applications or use them standalone.

On the other hand, the ecosystem might support other information visualization-related tasks, such as data transformations. Formatting data is essential to support some encodings or layouts [43], so a service that carries out this task and unburdens the front-end with these computations can be connected to the dashboard generator component to offer a complete pipeline.

As the dashboard generator, this service is also developed as an API (Figure 3). In this case, the input data will provide information regarding the computations to perform. Target data must be sent along with the following configuration parameters to enable the service to perform the requested operations:

- The set of variables from the dataset that will take part in the computations;
- The operation or operations to be performed (summary statistics, regressions, ratios, etc.);
- Filters (optional);
- Groupings (optional);
- Output data layout: tabular (default), nested, linked, etc.

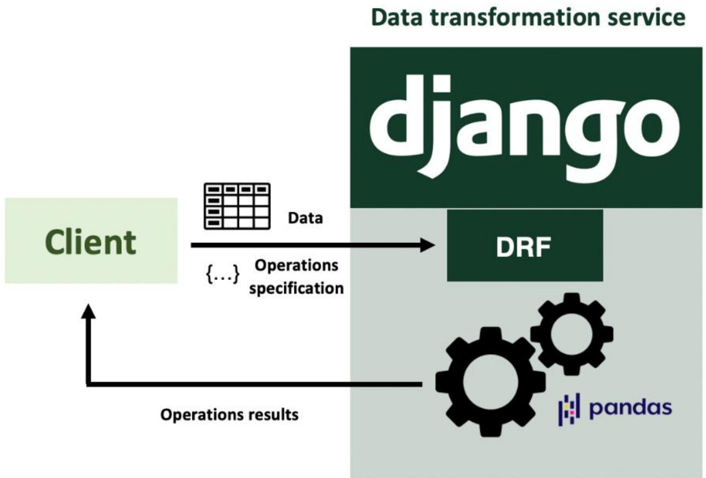


Figure 3. Schematic view of the data transformation service architecture.

5. Use Cases

This section aims to illustrate the generative process of the dashboard generator service. Through three use cases, we want to show not only the flexibility in terms of the data domain but also the flexibility in terms of necessities: the services can be used when data are stored in files locally (CSV, XLSX) or even make petitions to external endpoints (proprietary endpoints or even the ecosystem's computational service). Also, the generated source code could be stored as files or dynamically embedded or loaded in other applications.

5.1. Requesting Source Code to Obtain a Standalone Dashboard

The services' independence allows the integration of the dashboard generator with other technologies of the educative domain. In this example, part of the Open University (OU) dataset has been used to show a dashboard request [44].

Two visualizations will be requested, one to display the scores obtained in different assessments by the students and the other to display the range of scores by assessment. To obtain the source code, we need to build an HTTP POST request containing the dashboard layout and each visualization's configuration to generate both the HTML and the JavaScript code (Figure 4).

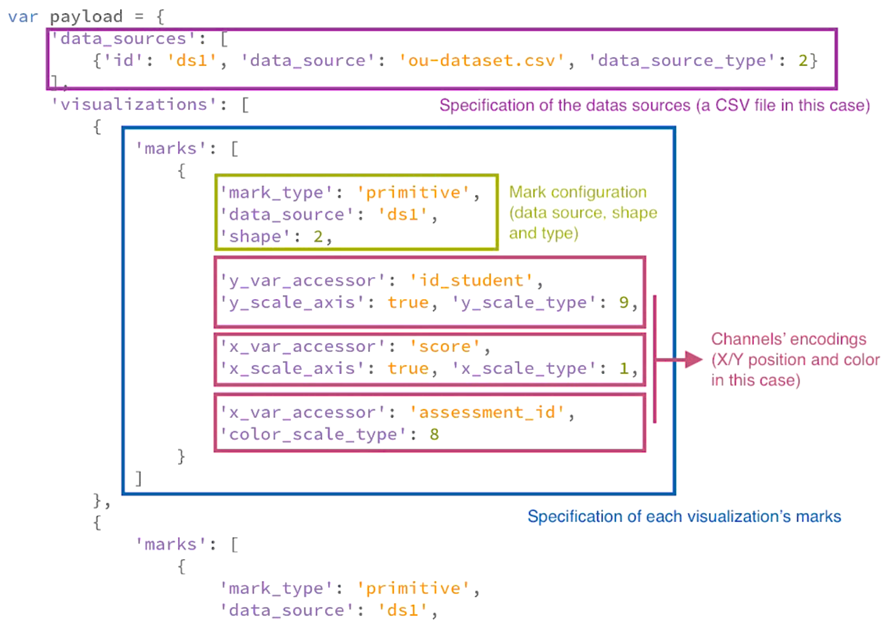

```
var payload = {
    'data_sources': [
        {'id': 'ds1', 'data_source': 'ou-dataset.csv', 'data_source_type': 2}
    ],
    'visualizations': [
        {
            'marks': [
                {
                    'mark_type': 'primitive',
                    'data_source': 'ds1',
                    'shape': 2,
                    'y_var_accessor': 'id_student',
                    'y_scale_axis': true, 'y_scale_type': 9,
                    'x_var_accessor': 'score',
                    'x_scale_axis': true, 'x_scale_type': 1,
                    'x_var_accessor': 'assessment_id',
                    'color_scale_type': 8
                }
            ]
        },
        {
            'marks': [
                {
                    'mark_type': 'primitive',
                    'data_source': 'ds1',
```


Figure 4. HTTP POST request's payload, which contains the dashboard configuration.

Once the HTTP request has been sent, the service gets the payload data (i.e., the dashboard configuration) and performs the application engineering process to yield the personalized source code. To do so, the input JSON object is processed by the dashboard generator process, which is in charge of filling the code templates with the specific information handed by the client.

This process' outcomes will be the source code of the dashboard, which is included in text format inside the API call response to the client. The source code could be used standalone and embedded within other applications by injecting the HTML and dynamically loading the JavaScript code. Figure 5 shows an excerpt of one of the generated JavaScript files.

```
function render_viz_0(ds1) {
    var margin = {
            top: 10,
            bottom: 40,
            left: 100,
            right: 50
        },
        width = 550,
        height = 450;

    var svg_0 = d3.select("#visualization-0")
        .append("svg")
        .attr("width", width + margin.left + margin.right)
        .attr("height", height + margin.top + margin.bottom)
        .append("g")
        .attr("transform", "translate(" + margin.left + "," + margin.top + ")");

    var x0 = d3.scaleLinear().domain([0, 100]).padding(0.2).range([0, 550])
```

A function is created for each declared visualization

SVG declaration

Scale declaration

Figure 5. Excerpt of a generated JavaScript source file.

Every generated JavaScript file follows the same structure:

1. Creation of the SVG container;
2. Declaration of the scales;
3. Creation of the visual marks;
4. Addition of each visual mark's channels.

The obtained source code can be deployed as a standalone web page: Figure 6 displays the rendered dashboard with the configured features at the beginning of the example.

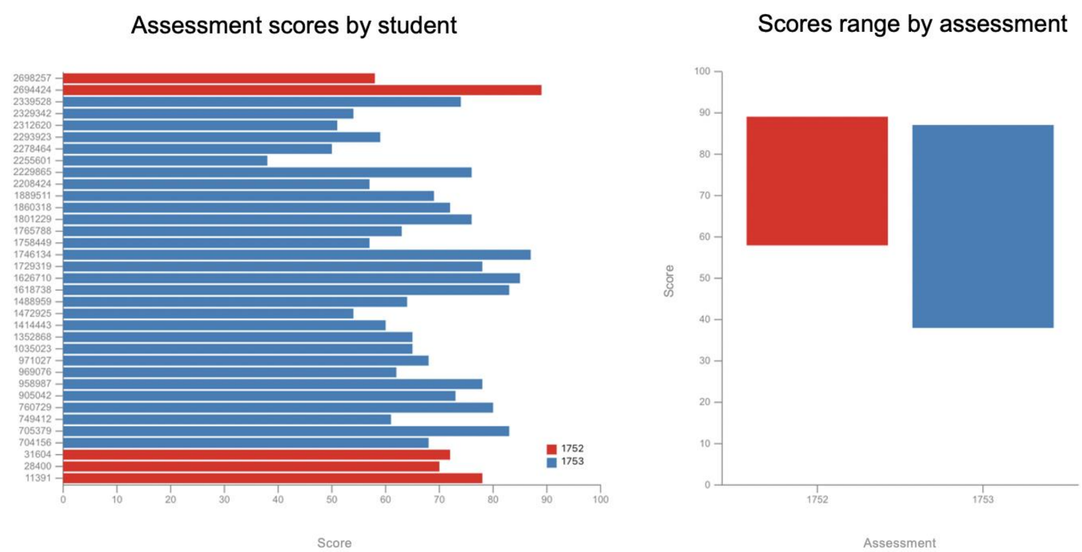


Figure 6. Rendered dashboard.

5.2. Integration with Other Components

In the previous use case, a dashboard was generated based on a configuration file and a dataset. This generation was straightforward because data were already in the right format for the chosen visualizations (in this case, simple visualizations such as bar charts), and no additional computations were needed.

However, data are not always in the right format for every visualization [43], and most times, it is necessary to transform the datasets before visualizing them. That is why a complementary data transformation service is included within the ecosystem. As will be discussed, adding this component benefits users in terms of delegating data transformations to an independent component and captures the implicit knowledge that is contained in the execution of data preprocessing tasks.

The data transformation component provides a solution for performing data computations and also to format data to different formats. As explained in Section 4, this component is also based on API calls. This example performs calculations on sociodemographic data to offer a Sankey diagram (which can be classified as a "flow layout" [43] or "parallel sets layout" [45]) in which its links represent the count of each category within each variable (Figure 7).

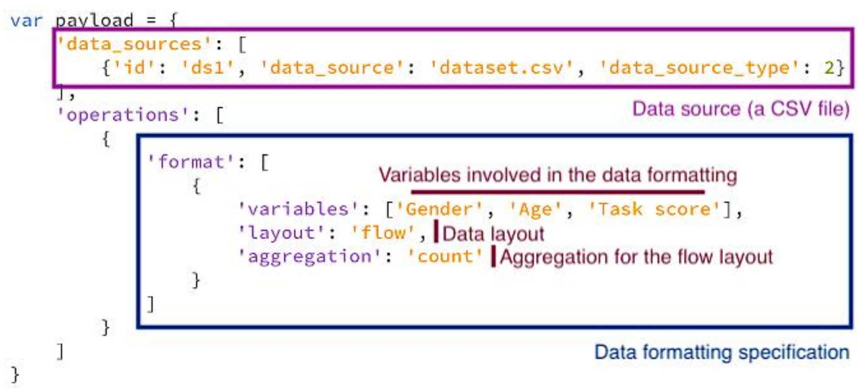


Figure 7. Formatting operation specification.

As with the dashboard generator service, the HTTP POST is processed by the data transformation service. In this case, the operation specification contains the information needed for the service to perform the data transformations requested by the client. Once the data are processed, they are returned within an HTTP response.

The API call results can be subsequently used with the dashboard generator service to create a visualization: in this case, a Sankey diagram (Figure 8).

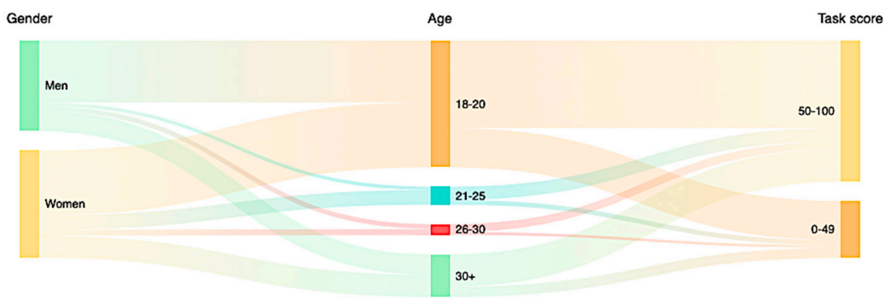


Figure 8. Rendered Sankey visualization that formats data as flows.

5.3. Dynamic Implementation of a Dashboard with Educative Purposes

The last use case is focused on the possibility of integrating source code dynamically within an existing web application. A user interface has been designed to explore and visualize a dataset within the medical domain in the following example.

The user interface allows users to explore the variables within the dataset and to craft personalized visualizations based on their variables' selection. Users can drag and drop the variables, select a visualization, and then, once confirmed, the front end computes the dashboard configuration and sends it to the dashboard generator service, which follows the same workflow as explained in the first example.

The final source code returned from the service can be dynamically loaded through JavaScript's DOM manipulation functions. In this sense, the existing web application is benefited by not carrying out all the data visualization logic, only focussing on offering a usable didactic tool for the medical domain (Figure 9).

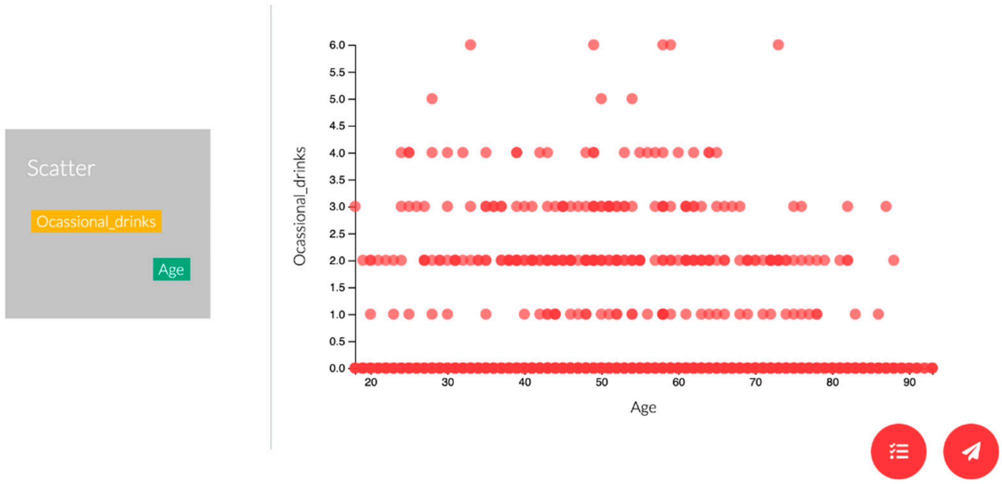


Figure 9. Fragment of the user interface integrated with the ecosystem's services. The right section of the figure shows the drag and drop space in which users are allowed to configure their visualizations graphically. The left section shows the generated visualization, which is dynamically embedded.

6. Discussion

This paper set the foundations for developing a technological ecosystem for designing and building information visualizations through holistic web services. Two of these ecosystem services are presented in this work: a service to generate information dashboards (transforming M1 models into M0 models) and a service to perform data transformations.

Adding a high-level layer to the design process of dashboards reduces their development time and their complexity in terms of programming. Another benefit is the possibility of structuring dashboard features in documents, which allow version control and further processing to identify interesting or useful features in different contexts.

A web service has been developed to serve this functionality through HTTP requests. This approach aims at the integration of different services programmatically. Returning the whole source code in plain JavaScript and HTML allows the users to retrieve a fully functional set of visualizations and rely on a template if they want to modify the generated code to match further requirements.

The generator service relies on a meta-modeling and software product line approach. The meta-model has been a useful resource for designing and developing the service; however, the domain engineering process has enabled a better dashboard domain understanding by identifying the primitive components generic to information dashboards and visualizations.

Relying on fine-grained features allows more variability points, meaning more elements can be customized when requesting a visualization source code. Fine-grained features also enable the analysis of the visualizations' primitives at a low level, thus providing a characterization of potentially useful visualizations. Suppose a data visualization works well in a specific context or use case. In that case, their features could be examined to identify which of them are beneficial and subsequently adapt them to other datasets or domains.

Another benefit that could yield a generative dashboard ecosystem is transparency and traceability, and knowledge management. By relying on services with well-defined interfaces, it is possible to follow the users' design decisions by analyzing their requests to the different services.

Sometimes it is difficult to materialize the implicit knowledge within design processes, as several times developers rely on heuristics, guidelines, or even default configurations. The generative dashboard ecosystem captures this implicit knowledge and structures it through the API calls' schema (which relies on the dashboard meta-model). This sets the foundations for reusing previously generated knowledge; if specific dashboard configurations worked well in a particular environment, they could be reused for similar contexts.

As mentioned before, the educational context can be a clear beneficiary of applying the software product lines to the dashboard domain. The amount of learning data generated due to the popularization of new technologies in education [46] makes it necessary to have new methods and instruments that allow obtaining benefits from such information. Nevertheless, as seen in the third use case, this service can be used for other educational purposes, such as creating didactic tools that let users exploring data without the necessity of having programming skills.

Although dashboards are handy tools for these analysis processes, it is necessary to consider the audience that will use them, especially in educational environments where user roles and profiles can be very heterogeneous in terms of objectives, characteristics, and preferences [6].

Being able to generate dashboards quickly, and dedicating more time to the design and conceptualization of the dashboard than its implementation, allows having products better designed and adapted to concrete situations in less time [47,48].

However, it is necessary to deeply evaluate this proposal. We plan to carry out meta-model validations through the automatic generation of already developed dashboards, to test the usability of the generated products against "manually" developed ones.

7. Limitations

Relying on web services implies the transference of data between systems, which could be critical if data are sensitive. This proposal addresses this problem by not storing the data after performing the operations or transformations. However, it is necessary to define a policy and even anonymization mechanisms if these services are exploited in production. In fact, not storing data could result in performance issues for large datasets and repetitive operations, so this challenge needs to be tackled both in terms of security and efficiency.

8. Conclusions

This work provides the foundation for designing an ecosystem for developing information dashboards based on different services with different well-defined functionalities. Specifically, this paper presents a web service to request the source code of customized dashboards and another web service to perform data operations. The dashboard generator service is implemented as an API that takes as an input the requested dashboard or visualization configuration and returns a set of HTML and JavaScript files containing the source code. On the other hand, the data transformation service takes as an input the transformation parameters and the dataset and returns the modified dataset.

Future steps will involve the addition of more services to the ecosystem to complement the dashboard generator and obtain services that could be connected to provide a whole dashboard developing pipeline: for example, a recommendation service of potentially features, a detector of potentially misleading information visualizations, data cleaning services, etc. In this proposal, we tested the approach's viability and flexibility; however, we also plan to test its acceptance, performance, and usability with users.

Author Contributions: Conceptualization, A.V.-I., F.J.G.-P. and R.T.; methodology, A.V.-I., F.J.G.-P. and R.T.; software, A.V.-I.; validation, A.V.-I., F.J.G.-P. and R.T; formal analysis, A.V.-I., F.J.G.-P. and R.T.; investigation, A.V.-I., F.J.G.-P. and R.T.; resources, A.V.-I., F.J.G.-P. and R.T.; writing—original draft preparation, A.V.-I.; writing—review and editing, F.J.G.-P. and R.T.; visualization, A.V.-I.; supervision, F.J.G.-P. and R.T.; project administration, F.J.G.-P. and R.T.; funding acquisition, F.J.G.-P. and R.T. All authors have read and agreed to the published version of the manuscript.

Funding: This research was partially funded by the Spanish Government Ministry of Economy and Competitiveness throughout the DEFINES project grant number [TIN2016-80172-R] and by the Spanish Government Ministry of Economy and Competitiveness and EU CHIST-ERA agreement throughout the PROVIDEDH project grant number [PCIN-2017-064]. This research was supported by the Spanish *Ministry of Education*, *Culture and Sport* under a FPU fellowship (FPU17/03276).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: No new data were created or analyzed in this study. Data sharing is not applicable to this article.

Acknowledgments: The authors would like to thank the InterAction and eLearning Research Group (GRIAL) for its support to conduct the present research https://grial.usal.es (accessed on 4 April 2021).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Sarikaya, A.; Correll, M.; Bartram, L.; Tory, M.; Fisher, D. What Do We Talk About When We Talk About Dashboards? *IEEE Trans. Vis. Comput. Graph.* **2018**, *25*, 682–692. [CrossRef] [PubMed]
2. García-Holgado, A.; García-Peñalvo, F.J. The evolution of the technological ecosystems: An architectural proposal to enhancing learning processes. In *Proceedings of the First International Conference on Technological Ecosystems for Enhancing Multiculturality (TEEM'13), Salamanca, Spain, 14–15 November 2013*; ACM International Conference Proceeding Series (ICPS); García-Peñalvo, F.J., Ed.; ACM: New York, NY, USA, 2013; pp. 565–571.
3. García-Holgado, A.; García-Peñalvo, F.J. A metamodel proposal for developing learning ecosystems. In *Learning and Collaboration Technologies. Novel Learning Ecosystems. 4th International Conference, LCT 2017. Held as Part of HCI International 2017, Vancouver, BC, Canada, 9–14 July 2017. Proceedings, Part I*; Lecture Notes in Computer Science, No. 10295; Zaphiris, P., Ioannou, A., Eds.; Springer International Publishing: Cham, Switzerland, 2017; pp. 100–109.
4. Yoo, Y.; Lee, H.; Jo, I.-H.; Park, Y. Educational dashboards for smart learning: Review of case studies. In *Emerging Issues in Smart Learning*; Springer: Berlin/Heidelberg, Germany, 2015; pp. 145–155.
5. Álvarez-Arana, A.; Villamañe-Gironés, M.; Larrañaga-Olagaray, M. Improving Assessment Using Visual Learning Analytics. *Educ. Knowl. Soc.* **2020**, *21*. [CrossRef]
6. Schwendimann, B.A.; Rodriguez-Triana, M.J.; Schwendimann, B.A.; Vozniuk, A.; Prieto, L.P.; Boroujeni, M.S.; Holzer, A.; Gillet, D.; Dillenbourg, P. Perceiving learning at a glance: A systematic literature review of learning dashboard research. *IEEE Trans. Learn. Technol.* **2017**, *10*, 30–41. [CrossRef]
7. Teasley, S.D. Student facing dashboards: One size fits all? *Technol. Knowl. Learn.* **2017**, *22*, 377–384. [CrossRef]
8. Vázquez-Ingelmo, A.; García-Peñalvo, F.J.; Therón, R. Information Dashboards and Tailoring Capabilities A Systematic Literature Review. *IEEE Access* **2019**, *7*, 109673–109688. [CrossRef]
9. Filonik, D.; Medland, R.; Foth, M.; Rittenbruch, M. A Customisable Dashboard Display for Environmental Performance Visualisations. In *Persuasive Technology. PERSUASIVE 2013*; Springer: Berlin/Heidelberg, Germany, 2013; pp. 51–62.
10. Mayer, B.; Weinreich, R. A dashboard for microservice monitoring and management. In Proceedings of the 2017 IEEE International Conference on Software Architecture Workshops (ICSAW), Gothenburg, Sweden, 5–7 April 2017; pp. 66–69.
11. Michel, C.; Lavoué, E.; George, S.; Ji, M. Supporting awareness and self-regulation in project-based learning through personalized dashboards. *Int. J. Technol. Enhanc. Learn.* **2017**, *9*, 204–226. [CrossRef]
12. Miotto, G.L.; Magnoni, L.; Sloper, J.E. The TDAQ Analytics Dashboard: A real-time web application for the ATLAS TDAQ control infrastructure. *J. Phys. Conf. Ser.* **2011**, *331*, 022019. [CrossRef]
13. Nascimento, B.S.; Vivacqua, A.S.; Borges, M.R. A flexible architecture for selection and visualization of information in emergency situations. In Proceedings of the 2016 IEEE International Conference on Systems, Man, and Cybernetics (SMC), Budapest, Hungary, 9–12 October 2016; pp. 003317–003322.
14. Elias, M.; Bezerianos, A. Exploration views: Understanding dashboard creation and customization for visualization novices. In *IFIP Conference on Human-Computer Interaction*; Springer: Berlin/Heidelberg, Germany, 2011; pp. 274–291.
15. Yalçın, M.A.; Elmqvist, N.; Bederson, B.B. Keshif: Rapid and expressive tabular data exploration for novices. *IEEE Trans. Vis. Comput. Graph.* **2018**, *24*, 2339–2352. [CrossRef] [PubMed]

16. Petasis, G.; Triantafillou, A.; Karstens, E. YourDataStories: Transparency and Corruption Fighting Through Data Interlinking and Visual Exploration. In *International Conference on Internet Science*; Springer: Berlin/Heidelberg, Germany, 2017; pp. 95–108.
17. Kumar, K.; Bose, J.; Soni, S.K. A Generic Visualization Framework based on a Data Driven Approach for the Analytics data. In Proceedings of the 2017 14th IEEE India Council International Conference (INDICON), Roorkee, India, 15–17 December 2017; pp. 1–6.
18. Cardoso, A.; Teixeira, C.J.V.; Pinto, J.S. Architecture for Highly Configurable Dashboards for Operations Monitoring and Support. *Stud. Inform. Control* **2018**, *27*, 319–330. [CrossRef]
19. Pastushenko, O.; Hynek, J.; Hruška, T. Generation of test samples for construction of dashboard design guidelines: Impact of color on layout balance. In *World Conference on Information Systems and Technologies*; Springer: Berlin/Heidelberg, Germany, 2018; pp. 980–990.
20. Vázquez-Ingelmo, A.; García-Peñalvo, F.J.; Therón, R. Domain engineering for generating dashboards to analyze employment and employability in the academic context. Presented at the 6th International Conference on Technological Ecosystems for Enhancing Multiculturality, Salamanca, Spain, 24–26 October 2018.
21. Logre, I.; Mosser, S.; Collet, P.; Riveill, M. Sensor data visualisation: A composition-based approach to support domain variability. In *European Conference on Modelling Foundations and Applications*; Springer: Berlin/Heidelberg, Germany, 2014; pp. 101–116.
22. Kintz, M. A semantic dashboard description language for a process-oriented dashboard design methodology. In Proceedings of the 2nd International Workshop on Model-based Interactive Ubiquitous Systems (MODIQUITOUS 2012), Copenhagen, Denmark, 25–28 June 2012; Volume 947, pp. 31–36.
23. Kintz, M.; Kochanowski, M.; Koetter, F. Creating User-specific Business Process Monitoring Dashboards with a Model-driven Approach. In Proceedings of the MODELSWARD, Porto, Portugal, 19–21 February 2017; pp. 353–361.
24. Palpanas, T.; Chowdhary, P.; Mihaila, G.; Pinel, F. Integrated model-driven dashboard development. *Inf. Syst. Front.* **2007**, *9*, 195–208. [CrossRef]
25. Radovanović, S.; Majstorović, B.; Kukolj, S.; Bjelica, M.Z. Device Cloud platform with customizable Remote User Interfaces. In Proceedings of the 2014 IEEE Fourth International Conference on Consumer Electronics Berlin (ICCE-Berlin), Berlin, Germany, 7–10 September 2014; pp. 202–204.
26. Belo, O.; Rodrigues, P.; Barros, R.; Correia, H. Restructuring Dynamically Analytical Dashboards Based on Usage Profiles. In *International Symposium on Methodologies for Intelligent Systems*; Springer: Berlin/Heidelberg, Germany, 2014; pp. 445–455.
27. Arjun, S. Personalizing data visualization and interaction. In Proceedings of the Adjunct Publication of the 26th Conference on User Modeling, Adaptation and Personalization, Nanyang Technological University, Singapore, 8–11 July 2018; ACM: New York, NY, USA, 2018; pp. 199–202.
28. van Hoecke, S.; Huys, C.; Janssens, O.; Verborgh, R.; van de Walle, R. Dynamic Monitoring Dashboards Through Composition of Web and Visualization Services. In *International Internet of Things Summit*; Springer: Berlin/Heidelberg, Germany, 2015; pp. 465–474.
29. Santos, H.; Dantas, V.; Furtado, V.; Pinheiro, P.; McGuinness, D.L. From data to city indicators: A knowledge graph for supporting automatic generation of dashboards. In *European Semantic Web Conference*; Springer: Berlin/Heidelberg, Germany, 2017; pp. 94–108.
30. Pleuss, A.; Wollny, S.; Botterweck, G. Model-driven development and evolution of customized user interfaces. In Proceedings of the 5th ACM SIGCHI Symposium on Engineering Interactive Computing Systems, London, UK, 24–27 June 2013; pp. 13–22.
31. Kleppe, A.G.; Warmer, J.; Bast, W. *MDA Explained. The Model Driven Architecture: Practice and Promise*; Addison-Wesley Longman Publishing Co., Inc.: Boston, MA, USA, 2003.
32. Mellor, S.J.; Scott, K.; Uhl, A.; Weise, D. Model-Driven Architecture. In *Proceedings of the Advances in Object-Oriented Information Systems: OOIS 2002 Workshops, Montpellier, France, 2 September 2002*; Bruel, J.-M., Bellahsene, Z., Eds.; Springer: Berlin/Heidelberg, Germany, 2002; pp. 290–297.
33. Álvarez, J.M.; Evans, A.; Sammut, P. Mapping between Levels in the Metamodel Architecture. In *Proceedings of the UML 2001—The Unified Modeling Language. Modeling Languages, Concepts, and Tools. UML 2001*; Lecture Notes in Computer Science; Gogolla, M., Kobryn, C., Eds.; Springer: Berlin/Heidelberg, Germany, 2001; Volume 2185, pp. 34–46.
34. Vázquez-Ingelmo, A.; García-Peñalvo, F.J.; Therón, R.; González, M.Á.C. Extending a dashboard meta-model to account for users' characteristics and goals for enhancing personalization. Presented at the Learning Analytics Summer Institute (LASI), Vigo, Spain, 27–28 June 2019.
35. Vázquez-Ingelmo, A.; García-Peñalvo, F.J.; Therón, R. Capturing high-level requirements of information dashboards' components through meta-modeling. Presented at the 7th International Conference on Technological Ecosystems for Enhancing Multiculturality (TEEM 2019), León, Spain, 16–28 October 2019.
36. García-Holgado, A.; García-Peñalvo, F.J. Validation of the learning ecosystem metamodel using transformation rules. *Future Gener. Comput. Syst.* **2019**, *91*, 300–310. [CrossRef]
37. Vázquez-Ingelmo, A.; García-Peñalvo, F.J.; Therón, R.; Filvà, D.A.; Escudero, D.F. Connecting domain-specific features to source code: Towards the automatization of dashboard generation. *Clust. Comput.* **2020**, *23*, 1803–1816. [CrossRef]
38. Clements, P.; Northrop, L. *Software Product Lines*; Addison-Wesley: Boston, MA, USA, 2002.
39. Gomaa, H. *Designing Software Product Lines with UML: From Use Cases to Pattern-Based Software Architectures*; Addison Wesley Longman Publishing Co., Inc.: Boston, MA, USA, 2004.

40. Vázquez-Ingelmo, A.; García-Peñalvo, F.J.; Therón, R. Taking advantage of the software product line paradigm to generate customized user interfaces for decision-making processes: A case study on university employability. *PeerJ Comput. Sci.* **2019**, *5*, e203. [CrossRef]
41. Django Software Foundation. Django Web Framework. Available online: https://www.djangoproject.com/ (accessed on 15 March 2015).
42. Vázquez-Ingelmo, A.; García-Peñalvo, F.; Therón, R.; García-Holgado, A. Specifying information dashboards' interactive features through meta-model instantiation. In *Proceedings of the LASI-SPAIN 2020. Learning Analytics Summer Institute Spain 2020: Learning Analytics. Time for Adoption? Valladolid, Spain, 15–16 June 2020*; CEUR-WS.org: Aachen, Germany, 2020.
43. Meeks, E. *D3. js in Action: Data Visualization with JavaScript*; Manning Publications: Shelter Island, NY, USA, 2018.
44. Kuzilek, J.; Hlosta, M.; Zdrahal, Z. Open university learning analytics dataset. *Sci. Data* **2017**, *4*, 170171. [CrossRef] [PubMed]
45. Wilke, C.O. *Fundamentals of Data Visualization: A Primer on Making Informative and Compelling Figures*; O'Reilly Media: Newton, MA, USA, 2019.
46. Ferguson, R. Learning analytics: Drivers, developments and challenges. *Int. J. Technol. Enhanc. Learn.* **2012**, *4*, 304–317. [CrossRef]
47. Liñán, L.C.; Pérez, Á.A.J. Educational Data Mining and Learning Analytics: Differences, similarities, and time evolution. *Int. J. Educ. Technol. High. Educ.* **2015**, *12*, 98–112.
48. Picciano, A.G. The evolution of big data and learning analytics in American higher education. *J. Asynchronous Learn. Netw.* **2012**, *16*, 9–20. [CrossRef]

7.24 Appendix X. Following up the progress of doctoral students and advisors' workload through data visualizations: A case study in a PhD programme

Following up the progress of doctoral students and advisors' workload through data visualizations: a case study in a PhD program

Andrea Vázquez-Ingelmo[1][0000-0002-7284-5593], Alicia García-Holgado[1][0000-0001-9663-1103], Helena Hernández Payo[1], Francisco José García-Peñalvo[1][0000-0001-9987-5584] and Roberto Therón[1,2][0000-0001-6739-8875]

[1] GRIAL Research Group, Computer Sciences Department, Research Institute for Educational Sciences, University of Salamanca, Salamanca, Spain

[2]VisUSAL Research Group. University of Salamanca. Salamanca, Spain
[andreavazquez, aliciagh, helenahdez, fgarcia, theron]@usal.es

Abstract. One of the most important aspects to consider during the development of a PhD is the students' progress, both for their advisors and the students themselves. However, several achievements of different natures are involved during a PhD (research stays, publications, seminars, research plans, etc.). For these reasons, we propose a set of data visualizations to support decision-making processes in a PhD program. A preliminary requirement elicitation process was carried out to obtain a design basis for the implementation and integration of these tools in the PhD portal. Once the visualizations were implemented, a usability study was performed to measure the perceived usability of the newly added PhD portal functionalities. This paper presents the design process and usability study outcomes of applying data visualizations to the learning outcomes of the PhD Programme in Education in the Knowledge Society at the University of Salamanca.

Keywords: Data visualization, PhD program, SUS, Usability study, Learning outcomes, PhD milestones.

1 Introduction

Data visualizations are crucial tools to understand and exploit Learning Analytics and educational outputs. They allow the transformation of raw data into valuable knowledge that could lead stakeholders into better decision-making processes.

It is very important to lead decisions using data, especially in the educational domain, because data and evidence-based policymaking provide the means to improve learning, quality, and the relationships between every involved actor within this context [1-3].

However, data is not always presented using the best visualizations (or not even presented using visualizations at all), which could result in weaker informed decision-making processes because data is not fully exploited. In fact, several roles and actors must be considered within the educational context, which also adds more complexity to the data exploitation process, because each actor have his or her own information goals and requirements.

For all these reasons, the design of data visualizations for Learning Analytics and educational outputs is not a trivial task, and it needs to be tackled using a user-centered approach [4].

This work describes the integration of data visualizations in a PhD portal to follow the progress of students and advisors during the development of their PhD thesis. The purpose of this integration is to provide more context to the PhD Program managers and academic committee regarding the workload and achievements of the PhD advisors and students, as well as to improve the engagement of the platform by presenting the milestones and current situation within the PhD to each student.

These data were already being collected through the PhD portal, but the statistics about them were not accessible through straightforward methods. The addition of visualizations is set to exploit these data and offer an accessible tool to fully understand the milestones, achievements and related metrics derived from the PhD program.

The rest of this paper is organized as follows. Section 2 outlines the Doctoral Programme in Education in the Knowledge Society, which is the PhD Program in which we integrated the information visualizations. Section 3 describes the methodology followed to collect the information requirements, as well as the user study that we carried out to validate the modified portal. Section 4 details the visualizations included, while section 5 presents the results of the user study. Finally, section 6 discusses the results and section 7 concludes the results with the conclusions derived from this work.

2 Context

The PhD Programme, "Education in the Knowledge Society", was established and launched in the academic year 2013-2014 at the University of Salamanca (Spain), following the Spanish Royal Decree 99/2011 [5, 6]. The Programme is based on four cornerstones.

Firstly, multidisciplinarity and interdisciplinarity. In this sense, the Programme is based on the foundations of the knowledge society, i.e., on technology and learning. The problems and challenges of this society are so complex that they cannot be tackled from a single perspective, hence the need for multidisciplinarity. However, many of them require the application of approaches from various disciplines; therefore, interdisciplinary interventions are sought and recommended, although the longer-term objective is to achieve true transdisciplinarity.

Secondly, this Programme is aligned with the University's R&D&I strategy, the regional and national R&D&I strategy and the objectives of the European H2020 Programme and its continuation Horizon Europe 2021-2027. The interdisciplinary fusion of engineering, medicine, communication, information and education is a global objective that has been reflected in other doctoral proposals at top international institutions such as Harvard.

Finally, the commitment to the quality of research and its dissemination and scientific outreach. Besides, the Education in the Knowledge Society PhD Programme has been aligned with the Open Access / Open Knowledge / Open Science movement [7-14], promoting that all the educational and research resources will be available in open access on the PhD portal [15, 16] or on the institutional repositories [17-19].

The web portal enables knowledge management inside and outside the PhD Programme (https://knowledgesociety.usal.es). This tool provides an environment in which students can manage all the knowledge they generate throughout their doctoral studies. Likewise, the PhD portal gives visibility and disseminates this knowledge, so the work carried out by junior researchers has a greater impact at a national and international level.

In addition, the PhD portal allows online monitoring of the doctoral students' progress, enabling the Academic Committee, the Quality Committee and PhD advisors to carry out periodic monitoring tasks. The doctoral students share their evidence, such as the research plan, annual reports, pre-doctoral visits, grants, publications, conferences, and another kind of activities related to their doctoral studies.

3 Methodology

3.1 Requirements elicitation

Before defining the design of the data visualizations to be included in the PhD portal, it is necessary to understand the requirements of the involved roles. As described in section 2, three main roles arise within the PhD portal users: doctoral student, PhD advisor and Manager/Academic Committee member.

A requirement elicitation process was carried out to capture important information requirements that must be considered during the design of the PhD portal's information visualizations.

The detailed results and the prototypical design of the PhD portal's visualizations can be consulted in [20]. To summarize, the main information requirements were the following:

- **PhD advisors:** information regarding their doctoral students, including publications, conference attendance/participation, and research profiles.
- **Academic committee members:** information related to the progress of all the doctoral students and their deadlines, as well as the number of doctoral students associated with each research group/advisor.
- **Doctoral students:** remaining activities, distribution of activities/milestones (publications, seminars, etc.) by type, status of each milestone, deadlines, enrollment dates, comparisons with other doctoral students, etc.

3.2 User study

The selected tool for this preliminary usability study of the integration of data visualizations into the PhD portal was the System Usability Scale (SUS) questionnaire [21].

Due to the fact that the majority of people involved in the PhD in Education in the Knowledge Society are Spanish-speaking, we employed the Spanish version of the scale [22].

The SUS questionnaire consists of 10 items rated on a 1 to 5 Likert scale (from “strongly disagree” to “strongly agree, respectively). The items are positive and negative alternated statements (to avoid response biases).

This questionnaire provides an effective, valid and reliable [23, 24] manner to rate a system’s usability. It is also an efficient test, due to the short quantity of items required to score the usability (10 items), and it can be applied over a wide range of systems [25]. In addition to the 10 items of the SUS questionnaire, we also collected a set of demographic variables, including:

- Age range
- Birthplace
- Gender
- PhD Programme role
- Enrollment year
- Frequency of use of the PhD portal
- PhD advisors’ situation (current students being advised)

Besides these demographic variables, two open fields were provided when at the end of the survey to allow users to remark any positive and negative aspects of the visualizations. These open fields enabled us to collect qualitative feedback in addition to the quantitative measures of the SUS.

We implemented the SUS questionnaire using a customized version of LimeSurvey (https://www.limesurvey.org), an Open Source online statistical survey web application.

3.3 Participants

We sent the implemented SUS questionnaire to the PhD Programme participants (including students, PhD advisors, and managers). A total of 35 persons answered the questionnaire, which, according to the literature, provide fairly reliable results (the SUS is reliable with a minimum sample size of 12 participants [24]).

Table 1 presents an overview of the participants that took part in the usability test.

Table 1. Participants in the usability test

Role	Female	Male
Students	18	9
Advisors / Quality Committee	3	5
Total	21	14

On the other hand, regarding the frequency of use of the PhD portal, most participants remarked that they use the portal weekly or monthly. These data are detailed in Table 2.

Table 2. Frequency of use of the PhD Portal

Role	Daily	Weekly	Monthly	3 months basis
Students	2	10	7	8
Advisors / Quality Committee	0	1	7	0
Total	2	11	14	8

3.4 Data analysis

The outcomes of the instrument were analyzed using the Python Pandas [26] library. The individual SUS score of each participant was computed, to finally obtain the mean of all scores (i.e., the usability score of the PhD portal visualizations).

The interpretation of the results is based on previous System Usability Scale studies and benchmarks [27, 28], which allow meaningful SUS score comparisons to provide significant insights into the study outcomes.

All the source code developed for this analysis is available at https://github.com/AndVazquez/phd-visualizations-sus.

4 Visualizations

The PhD portal helps to monitor the doctoral student's activities, but even so, when the student has a lot of activity, it is very difficult to know if he/she is meeting all the necessary milestones. Based on the requirements elicitation, the portal needs to improve this monitoring support. In particular, the PhD advisors and the Academic Committee need information regarding the doctoral students' progress. Moreover, doctoral students requested information about the distribution of milestones and remaining activities.

For this reason, a set of tools has been implemented in the portal to facilitate the monitoring of doctoral students.

4.1 PhD students' visualizations

Firstly, a set of tools that help doctoral students, their advisors, and the Academic Committee to know the status of the main milestones to be reached by a doctoral student. This is accomplished through a timeline displayed in the doctoral student's profile, and it is only visible if the student is logged in the portal (Fig. 1). This timeline is complemented by a detailed view that allows students to find reports, research plans, and other milestones (Fig. 2) quickly and easily.

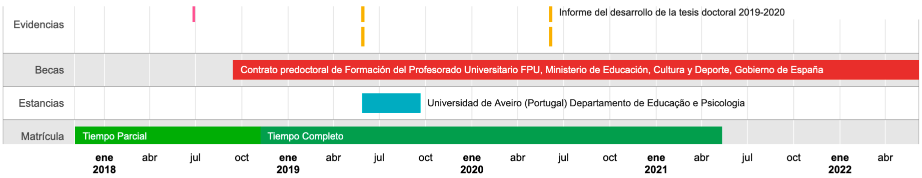


Fig. 1. Doctoral student's progress timeline.

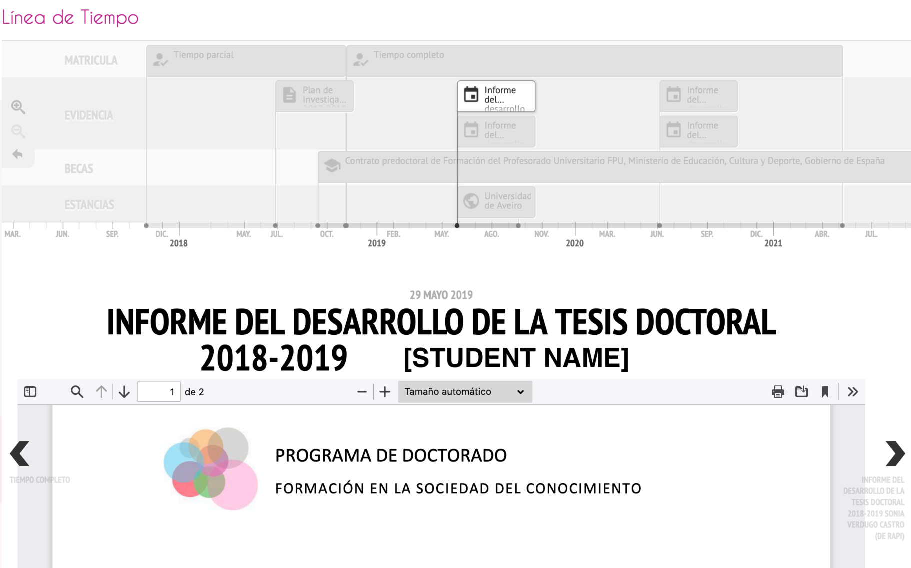


Fig. 2. Timeline with detailed information about the main evidence uploaded by a doctoral student.

4.2 PhD advisors' visualizations

Furthermore, the requirement elicitation process also identified the need for information about the workload of PhD advisors, including the number of doctoral students associated with each advisor.

Therefore, a set of static visualizations has been implemented to provide statistics on the advisors' workload inside the PhD Programme. This tool is not only for the advisors but also for the Academic and Quality Committees. Although the information was already available at the PhD portal, it was not easy to get and analyze. In particular, a new tab was included in the advisors' profile, which is only visible for themselves and the Academic and Quality Committees.

The charts show the number of theses supervised and the status of the theses (in progress, completed, failed, and doctoral students who have dropped out). Besides, there is a chart that summarizes the results in the completed theses. In addition, there are two charts related to quality indicators. Firstly, the number of doctoral grants held by the advisor's doctoral students. Secondly, the number of seminars or workshops delivered as part of the activities organized by the PhD Programme. Fig. 3 and 4 show a real example of an advisor with the heaviest workload in the PhD Programme.

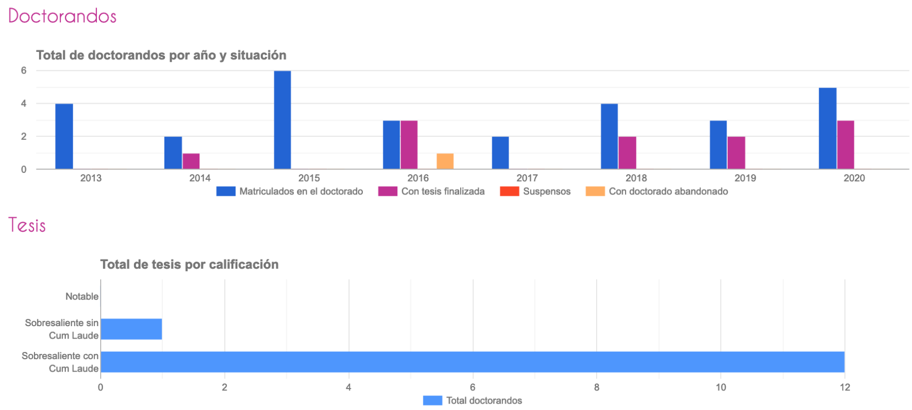


Fig. 3. Advisor's statistics about doctoral thesis supervised across the years.

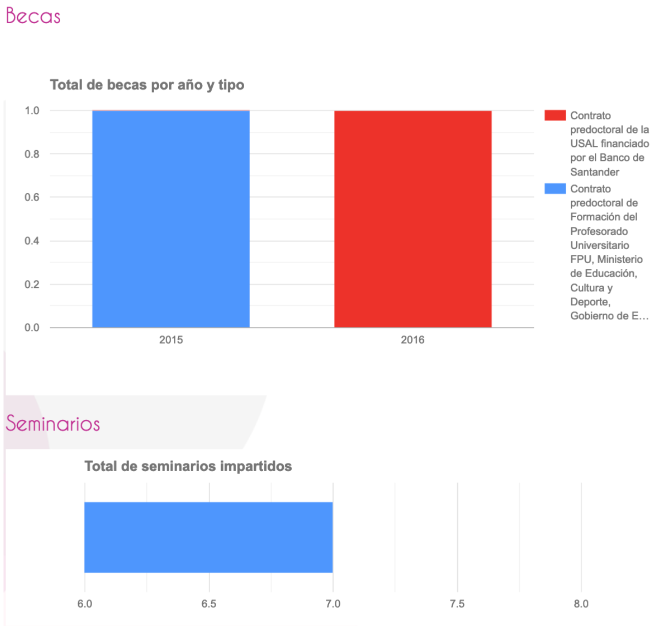


Fig. 4. Advisor's statistics about the number of doctoral students with grants and seminars delivered.

5 Results

Although the SUS is set to provide a single usability score [23], subsequent research found a two-dimensional nature of this scale [29], which allows the calculation of two complementary measures: the system's learnability score, and the system's usability score. From the items that are part of the SUS questionnaire, items 4 and 10 can be employed to remark the learnability of the system, while the remaining items are used to obtain its perceived usability [29].

Considering this, we analyzed the results to obtain every result, including the learnability. First, the overall SUS score (considering every item), and the learnability and usability scores (taking into account items 4 and 10, and the remaining items, respectively).

These scores were transformed into a scale from 0 to 100, as in the original SUS scoring method, allowing researchers to perform comparisons.

The calculations yielded the following outcomes:

- The average perceived usability of the PhD portal visualizations is **75.36**, which can be considered as a good SUS score, as it is above the average SUS score (68.00) and falls around the 75th percentile (interpretation based on the studies done in [27, 28]).
- On the other hand, the perceived learnability is **74.29**, a slightly lower score than the usability (**75.63**), both being acceptable and good scores following the adjective scale of the SUS [27].

Fig. 5. summarizes these results, also including the individual scores for every participant (represented by overlapping circles) across the three dimensions considered: total SUS score, Usability score and Learnability score.

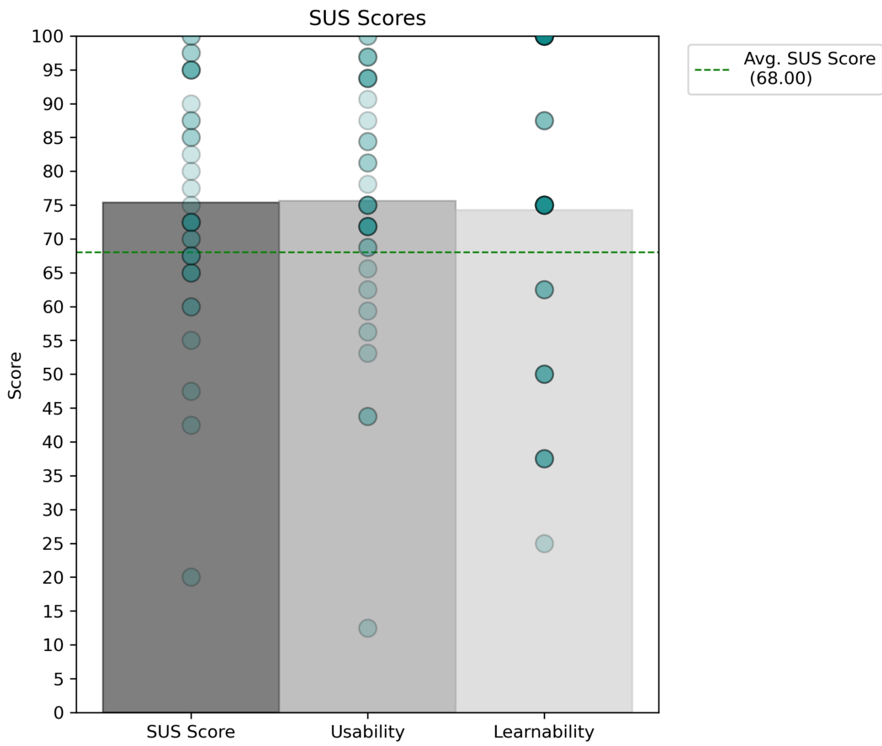


Fig. 5. Visual representation of the SUS questionnaire results regarding the PhD portal data visualization tools' usability and learnability scores.

6 Discussion

The outcomes of the usability study showed that the information visualizations included in the PhD portal were appreciated by the students as well as by the PhD advisors and members of the Quality Committee.

Specifically, from the 35 answers collected using the SUS questionnaire, we obtained an average perceived usability of 75.36, which is a "Good" score according to the SUS adjective ratings [27]. Learnability and Usability scored are interpreted in the same way.

The positive and negative feedback collected through open text fields shed more light on these results and complemented them to provide a more detailed picture of the included data visualizations' implications. Most of the positive comments remarked on the ease-of-use of the data visualizations as well as their design, versatility, and engagement that these visual tools provide.

One specific comment pointed out that "although it can be obvious that each person knows their situation [in the PhD program], providing a timeline where each [PhD] milestone is visually displayed allows the organization and planning of the remaining time within the PhD program". This was one of the main motivations to include data visualizations in the PhD portal; visualizations are powerful tools to understand data at first sight, which fit perfectly as tools to tackle the challenge of providing a big picture of the achievements and learning outcomes during a PhD.

In fact, as introduced before, these visualizations were not only designed for students, but also for managers. These roles also rated high the new functionality of the PhD portal. However, there are some limitations to this approach. One member of the Quality Committee commented that "statistics and graphs are based on data that students upload themselves, and this upload task is not always carried out properly".

The effectiveness of the data visualizations is tightly coupled with the quality of the data they display. If a doctoral student does not update his or her progress, then the visualizations will not be useful at all, because there will not be meaningful information to show. However, another reason to include this functionality in the PhD portal was to improve the engagement of doctoral students with the portal, and to motivate them to keep their progress up to date. We also plan to research the impact of the data visualizations on the students' initiative to upload their achievements into the portal.

To sum up, the application of the SUS test provided very useful insights about the usability of data visualizations to follow the doctoral students' progress and PhD advisors' workload. The SUS placed the system in a good range of usability, with room for improvement based on qualitative feedback.

7 Conclusions

A set of data visualizations have been included in the PhD portal of the Doctoral Programme in Education in the Knowledge Society. These data visualizations have several goals: to provide a big picture of the different reached milestones and achievements of doctoral students and to visualize the workload of PhD advisors in a straightforward way for better decision-making. Moreover, these visualizations also aim at improving the engagement of the involved actors with the PhD portal.

The System Usability Score was carried out to obtain insights about the usability of the data visualizations. It is important to remark that this test is not diagnostic; it gives an overview of the usability of a system. The outcomes of the study provided an average score of 75.36, which is a score above the average (68) and considered a good result.

Our implementation of the SUS also included open fields in the questionnaire. These comments provided valuable qualitative feedback from the participants, which allows us to mark our path to further improve the visualizations. Also, it provided feedback about the general thoughts regarding the newly added functionality, being most of them positive.

Future works will be focused on solving some technical problems detected by some participants during the usability test, as well as carrying out more studies to keep improving the visualization components. On the other hand, we also plan to add more functionalities (filters, advanced interactivity, etc.) with the goal of improving decision-making processes related with the PhD Program management.

Acknowledgements. This research work has been carried out within the University of Salamanca PhD Programme on Education in the Knowledge Society scope (http://knowledgesociety.usal.es) and was supported by the Spanish *Ministry of Education*, *Culture and Sport* under a FPU fellowship (FPU014/04783).

References

1. Cooper, M.M.: Data-driven education research. Science 317, 1171-1171 (2007)
2. Custer, S., King, E.M., Atinc, T.M., Read, L., Sethi, T.: Toward Data-Driven Education Systems: Insights into Using Information to Measure Results and Manage Change. Center for Universal Education at The Brookings Institution (2018)
3. Mandinach, E.B., Honey, M.: Data-driven school improvement: Linking data and learning. Teachers College Press (2008)
4. Abras, C., Maloney-Krichmar, D., Preece, J.: User-centered design. Bainbridge, W. Encyclopedia of Human-Computer Interaction. Thousand Oaks: Sage Publications 37, 445-456 (2004)
5. García-Peñalvo, F.J.: Education in knowledge society: A new PhD programme approach. Proceedings of the First International Conference on Technological Ecosystem for Enhancing Multiculturality (TEEM'13) (Salamanca, Spain, November 14-15, 2013), pp. 575-577. ACM, New York, NY, USA (2013)
6. García-Peñalvo, F.J.: Formación en la sociedad del conocimiento, un programa de doctorado con una perspectiva interdisciplinar. Revista Teoría de la Educación: Educación y Cultura en la Sociedad de la Información 15, 4-9 (2014)
7. García-Peñalvo, F.J., García de Figuerola, C., Merlo-Vega, J.A.: Open knowledge management in higher education. Online Information Review 34, 517-519 (2010)
8. García-Peñalvo, F.J., García de Figuerola, C., Merlo-Vega, J.A.: Open knowledge: Challenges and facts. Online Information Review 34, 520-539 (2010)
9. García-Peñalvo, F.J.: Open Access Myths and Realities. Education in the Knowledge Society 18, 7-20 (2017)
10. García-Peñalvo, F.J.: Publishing in open access. Journal of Information Technology Research 10, vi-viii (2017)
11. Ramírez-Montoya, M.S., García-Peñalvo, F.J.: Co-creation and open innovation: Systematic literature review. Comunicar 26, 9-18 (2018)
12. Ramírez-Montoya, M.S., García-Peñalvo, F.J., McGreal, R.: Shared Science and Knowledge. Open Access, Technology and Education. Comunicar 26, 1-5 (2018)
13. Merlo-Vega, J.A. (ed.): Ecosistemas del acceso abierto. Ediciones Universidad de Salamanca, Salamanca, España (2018)
14. Fecher, B., Friesike, S.: Open Science: One Term, Five Schools of Thought. In: Bartling, S., S., F. (eds.) Opening Science. The Evolving Guide on How

the Web is Changing Research, Collaboration and Scholarly pp. 17-47. Springer, Cham (2014)

15. García-Holgado, A., García-Peñalvo, F.J., Rodríguez-Conde, M.J.: Definition of a technological ecosystem for scientific knowledge management in a PhD Programme. In: Alves, G.R., Felgueiras, M.C. (eds.) Proceedings of the Third International Conference on Technological Ecosystems for Enhancing Multiculturality (TEEM'15) (Porto, Portugal, October 7-9, 2015), pp. 695-700. ACM, New York, NY, USA (2015)
16. García-Peñalvo, F.J., Rodríguez-Conde, M.J., Verdugo-Castro, S., García-Holgado, A.: Portal del Programa de Doctorado Formación en la Sociedad del Conocimiento. Reconocida con el I Premio de Buena Práctica en Calidad en la modalidad de Gestión. In: Durán Ayago, A., Franco Pardo, N., Frade Martínez, C. (eds.) Buenas Prácticas en Calidad de la Universidad de Salamanca: Recopilación de las I Jornadas. REPOSITORIO DE BUENAS PRÁCTICAS (Recibidas desde marzo a septiembre de 2019), pp. 39-40. Ediciones Universidad de Salamanca, Salamanca, España (2019)
17. Ferreras-Fernández, T., García-Peñalvo, F.J., Merlo-Vega, J.A.: Open access repositories as channel of publication scientific grey literature. In: Alves, G.R., Felgueiras, M.C. (eds.) Proceedings of the Third International Conference on Technological Ecosystems for Enhancing Multiculturality (TEEM'15) (Porto, Portugal, October 7-9, 2015), pp. 419-426. ACM, New York, USA (2015)
18. García-Peñalvo, F.J.: The Future of Institutional Repositories. Education in the Knowledge Society 18, 7-19 (2017)
19. González-Pérez, L.I., Ramírez-Montoya, M.S., García-Peñalvo, F.J.: User experience in institutional repositories: A systematic literature review. International Journal of Human Capital and Information Technology Professionals (IJHCITP) 9, 70-86 (2018)
20. Vázquez-Ingelmo, A., García-Peñalvo, F.J., Therón, R.: Generating Dashboards Using Fine-Grained Components: A Case Study for a PhD Programme. pp. 303-314. Springer International Publishing, (2020)
21. Brooke, J.: SUS-A quick and dirty usability scale. Usability evaluation in industry 189, 4-7 (1996)
22. Sevilla-Gonzalez, M.D.R., Moreno Loaeza, L., Lazaro-Carrera, L.S., Bourguet Ramirez, B., Vázquez Rodríguez, A., Peralta-Pedrero, M.L., Almeda-Valdes, P.: Spanish Version of the System Usability Scale for the Assessment of Electronic Tools: Development and Validation. JMIR Hum Factors 7, e21161 (2020)
23. Brooke, J.: SUS: a retrospective. Journal of usability studies 8, 29-40 (2013)
24. Tullis, T.S., Stetson, J.N.: A comparison of questionnaires for assessing website usability. In: Usability professional association conference, pp. 1-12. (2004)
25. Bangor, A., Kortum, P.T., Miller, J.T.: An empirical evaluation of the system usability scale. Intl. Journal of Human–Computer Interaction 24, 574-594 (2008)

26. McKinney, W.: pandas: a foundational Python library for data analysis and statistics. Python for High Performance and Scientific Computing 1-9 (2011)
27. Bangor, A., Kortum, P., Miller, J.: Determining what individual SUS scores mean: Adding an adjective rating scale. Journal of usability studies 4, 114-123 (2009)
28. Sauro, J.: A practical guide to the system usability scale: Background, benchmarks & best practices. Measuring Usability LLC (2011)
29. Lewis, J.R., Sauro, J.: The Factor Structure of the System Usability Scale. pp. 94-103. Springer Berlin Heidelberg, (2009)

7.25 Appendix Y. Proof-of-concept of an information visualization classification approach based on their fine-grained features

Received: 13 December 2020 | Revised: 3 April 2021 | Accepted: 13 October 2021

DOI: 10.1111/exsy.12872

ORIGINAL ARTICLE

Expert Systems WILEY

Proof-of-concept of an information visualization classification approach based on their fine-grained features

Andrea Vázquez-Ingelmo[1] | Alicia García-Holgado[1] | Francisco José García-Peñalvo[1] | Roberto Therón[2]

[1]GRIAL Research Group, Computer Science Department, University of Salamanca, Salamanca, Spain

[2]VisUSAL, GRIAL Research Group, Computer Science Department, University of Salamanca, Spain

Correspondence
Andrea Vázquez-Ingelmo, GRIAL Research Group, Computer Science Department, University of Salamanca, Spain.
Email: andreavazquez@usal.es

Funding information
Spanish Ministry of Education and Vocational Training under an FPU fellowship (FPU17/03276)

Abstract

The misinformation problem affects the development of the society. Misleading content and unreliable information overwhelm social networks and media. In this context, the use of data visualizations to support news and stories is increasing. The use of misleading visualizations both intentionally or accidentally influence in the audience perceptions, which usually are not visualization and domain experts. Several factors influence o accurately tag a visualization as confusing or misleading. In this paper, we present a machine learning approach to detect if an information visualization can be potentially confusing and misunderstood based on the analytic task it tries to support. This approach is supported by fine-grained features identified through domain engineering and meta modelling on the information visualization and dashboards domain. We automatically generated visualizations from a tri-variate dataset through the software product line paradigm and manually labelled them to obtain a training dataset. The results support the viability of the proposal as a tool to support journalists, audience and society in general, not only to detect confusing visualizations, but also to select the visualization that supports a previous defined task according to the data domain.

KEYWORDS

data visualization, fake news, machine learning, misinformation, misleading visualization

1 | INTRODUCTION

Information visualizations are everywhere. They are employed to illustrate data to back-up news, reports, ideas, or are even used as standalone elements, where users can explore through interaction patterns the underlying data. Information visualizations allow users to understand complex datasets, and to generate knowledge from raw data through visual analysis (Andrews, 2019; Keim et al., 2008; Tufte & Graves-Morris, 2014; Ware, 2012).

However, these tools can present dangerous complexities if they are not properly designed. A poorly designed information visualization could not only be inefficient, but also misleading, which could even guide its audience to reach wrong insights (Cairo, 2019; Pandey et al., 2015). Misleading visualizations contribute to the growing abundance of misinformation, intentionally or accidentally, and this has an impact in the well-being of the society (Lewandowsky et al., 2017; Shu et al., 2019).

In a time in which fake news are proliferating, detecting if an information visualization is potentially confusing is crucial, because visual elements could be highly persuasive (Pandey et al., 2014).

But it is necessary to account for several factors in order to accurately tag a visualization as confusing or misleading. For example, the data domain, the target audience, the underlying relationships among the variables, the supported tasks (Maletic et al., 2002), are very important factors that could determine if a visualization has the best configuration for its purpose. Moreover, studying all these factors is time-consuming and needs support from visualization and domain experts.

For these reasons, we present a viability test for automating the identification of potentially misleading or confusing features in the data visualization domain, as it could serve as a tool for practitioners for checking if the characteristics of the information visualizations they are designing fit their goal and do not distort the underlying data they are trying to convey.

This proposal is based on fine-grained features that we have arranged into a meta-model through domain engineering. While domain engineering provides a framework to identify common features and variability points of systems within a domain (in this case, the dashboards/information visualization domain), meta-modelling provides a methodology to organize these features into abstract models that can be instantiated to obtain specific products.

By identifying these fine-grained characteristics, we have trained a random forest (RF) classifier whose outcome is an indication regarding if a visualization could be confusing given its data context. The training set has been automatically generated using a software product line (SPL) approach, and manually tagged by the authors.

The main contributions of this work are:

1. A theoretical definition of the problem's domain: information visualizations and dashboards need to be decomposed and structured into fine-grained features in order to apply more complex algorithms to them.
2. The definition of an automatic information visualization generator based on a meta-model. Through this approach visualizations can be massively generated by configuring a set of parameters and constraints.
3. A proof of concept of the application of classification algorithms to detect if a visualization can be potentially confusing based on features identified through meta-modelling

The rest of this paper is organized as follows. Section 2 outlines the background in which this work is framed. Section 3 describes the methodology followed, including the development of the meta-model, the generation of the training set through a tagging process and the employed ML algorithms. Section 4 presents the proof of concept to validate the viability of the presented approach, while Section 5 discusses the obtained results. Finally, Section 6 outlines the limitations we have encountered by using the proposed approach, and Section 7 presents the conclusions derived from this work.

2 | BACKGROUND

Information visualizations are powerful tools to convey information in a straightforward way. But visualizing data is not a universal, deterministic process. There are a lot of factors that can influence and determine the effectivity of information visualizations, and these could even vary depending on the context, the specific audience or the data domain.

Studies found in the literature point out the influence of different design choices or configurations on the audiences' reached insights or their understanding of the displayed data. For example, in (Correll et al., 2020), the effect of truncating the Y-axis scale was measure to show how it influences the perceived severity of the data effect size. In (Pandey et al., 2015), the influence of different visualization distortion techniques (such as message reversal or message exaggeration) was tested, proving that they could lead to the misunderstanding of the data on the reader's side.

On the other hand, other studies have tested the influence of the visualization task and data characteristics on the effectiveness of different information visualizations. In (Saket et al., 2018), the authors found that the users' effectiveness by using certain visualization were significantly different from one task to another. The visualization task along with the data distribution were also tested in (Y. Kim & Heer, 2018), concluding that there are significant differences in effectiveness depending on these factors and the selected encoding channels of some visualizations types.

Being aware of the factors that make an information visualization more effective and efficient could be used to create new design guidelines and even recommend the best configuration to improve the effectiveness of a visualization (Gotz & Wen, 2009; Kaur & Owonibi, 2017; Vartak et al., 2017; Voigt et al., 2012).

Some methods use visual mapping and rules to recommend a certain visualization based on the target data to be displayed (Kaur & Owonibi, 2017) by analysing the dataset to be displayed and applying rules to select a proper visual configuration like Tableau's Show Me (Mackinlay et al., 2007), Manyeyes (Viegas et al., 2007) or Voyager (Wongsuphasawat et al., 2015).

On the other hand, the growing popularity of artificial intelligence (AI) to tackle different problems (Arthur, 2020; Halim et al., 2016; Halim & Rehan, 2020; Liu et al., 2019) has also led to some works taking advantage of AI approaches to infer the best information visualization configurations. VizML (Hu et al., 2019) employed the Plotly API to retrieve information about different Plotly-based visualizations to train a set of models. The outputs of these models are a set of visualization design choices both at visualization and encoding-level, including mark types or the axes' properties regarding axes. A similar approach was taken in Data2Vis (Dibia & Demiralp, 2019), where the characteristics of the datasets to be displayed were used as an input for a deep learning sequence to sequence model to obtain a recommended visualization specification in Vega-Lite. Another AI-based approach is applied in (Muhammad & Halim, 2016), where the authors predicted the best visualizations techniques given on meta-data and the task that the user wants to perform using an artificial neural network.

In this work, we aim to test the viability of applying machine learning (ML) algorithms to automatically identify those determining features that have been previously pointed out to be influential to the audiences' reached insights.

3 | METHODOLOGY

3.1 | Features

The main issue when training a ML is the necessity of relying in datasets with proper and relevant features. As introduced in the background section, information visualization and dashboards are composed by different primitive elements (axes, scales, marks, etc.) which users perceive and decode to understand the displayed data and to generate knowledge (Keim et al., 2008; Ware, 2012).

However, it is important to decompose and refactor these features to make them useful as an input for a ML algorithm.

The approach we have taken for identifying and structuring information visualizations based on their most fine-grained features is based on domain engineering (Arango, 1988; Kang et al., 1990; Metzger & Pohl, 2007) and meta-modelling (Álvarez et al., 2001; Kleppe et al., 2003).

These approaches provide frameworks to arrange abstract and common features among products of the same domain into high-level models that can be subsequently instantiated to obtain specific systems. The outcome of this process has been a dashboard meta-model, which is composed by three main section: the user, the layout and the primitive components (Figures 1 and 2). The explanation of the meta-model's elements is out of the scope of this paper, but it can be consulted in (Vázquez-Ingelmo, García-Peñalvo, & Therón, 2019c; Vázquez-Ingelmo, García-Peñalvo, Therón, & Conde González, 2019; Vázquez-Ingelmo, García-Peñalvo, Therón, & Conde, 2020).

The meta-model was the preliminary step before building the training dataset. Using the abstract features and relationships identified during the mentioned domain-engineering phase, we managed to design the data structures that would feed the ML algorithm.

3.2 | Visualization generation

We looked for different ways of obtaining visualization datasets that we could use as a training input for a ML algorithm. The first approach we tried was to research different news portal in pursue of misleading or confusing information visualization, to subsequently structure their features taking as a reference the previously presented meta-model. However, this approach was time-consuming, and the number of tagged visualizations we obtained through this method was not very significant given the time devoted.

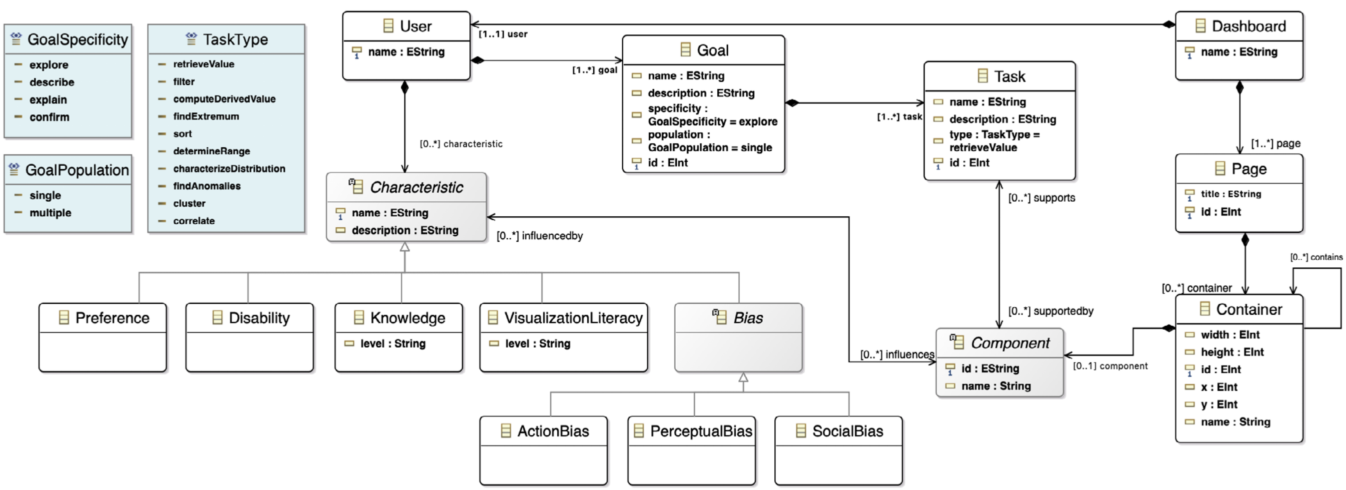


FIGURE 1 User and layout section of the dashboard meta-model, including the visualization goals, supported tasks and user characteristics. *Source*: (Vázquez-Ingelmo, García-Holgado, García-Peñalvo, & Therón, 2020)

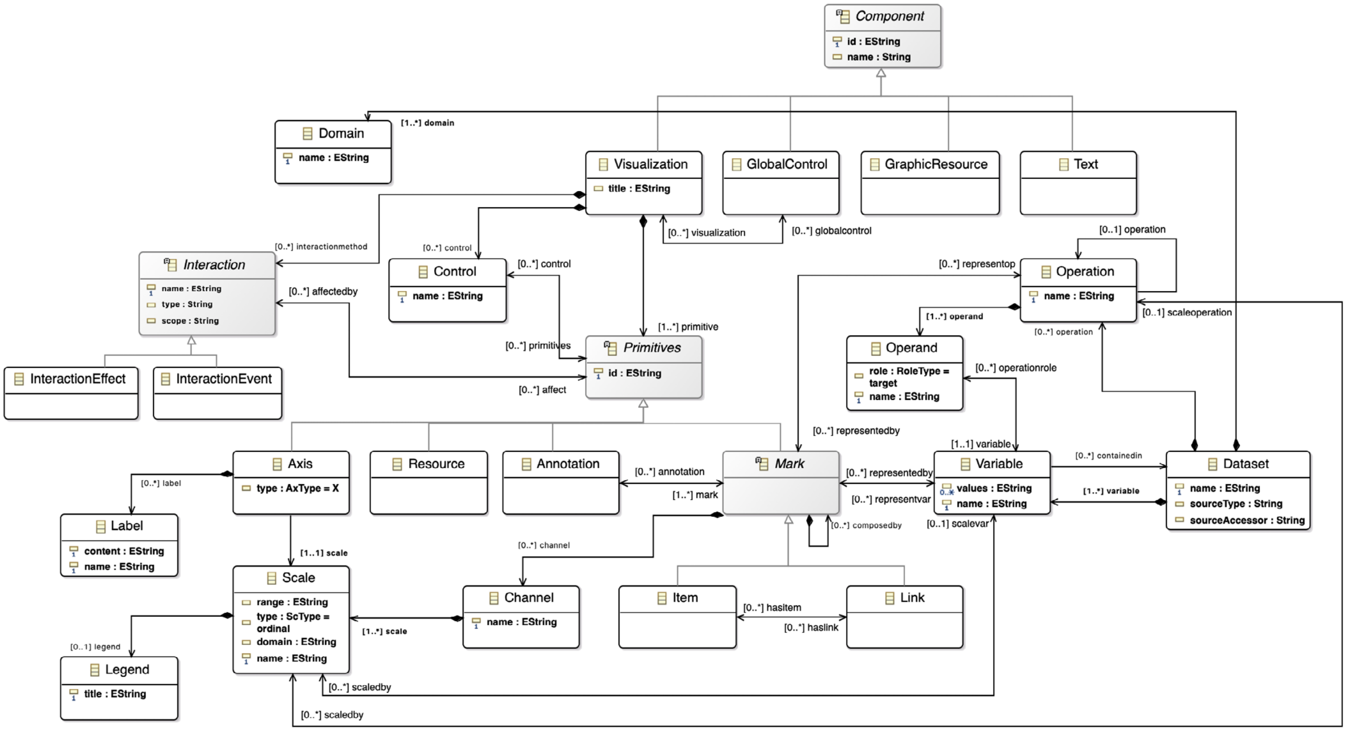


FIGURE 2 Components' section of the dashboard meta-model, in which information visualizations' fine-grained features and relationships are represented. *Source*: (Vázquez-Ingelmo, García-Holgado, et al., 2020)

That is why we decided to generate our own dataset of different information visualizations by using a SPL engineering approach (Clements & Northrop, 2002; Pohl et al., 2005). We have previously tested this approach to generate information dashboards in different domains (Vázquez-Ingelmo, García-Holgado, García-Peñalvo, & Therón, 2019; Vázquez-Ingelmo, García-Holgado, et al., 2020; Vázquez-Ingelmo et al., 2018, 2019d), which fits perfectly as a method for tackling the issue of generating a significant amount of information visualizations with different configurations. In addition, by using this approach, the configurations of the generated visualizations are already structured and prepared for their processing and use as an input, which also saved us time and allowed us to focus more thoroughly on the tagging process.

The generation process relies on the use of code templates and a python script in which the configuration parameters can be tuned. The output of the generation process are HTML and JavaScript files containing D3 (Bostock et al., 2011; Meeks, 2018) code to render each individual visualization.

However, one of the issues derived from the generation process was the explosion of configuration combinations, because the Python script obtains a combinatory of all possible values that a feature can hold. For these reasons, to make a proof of concept, we constrained the generation of configurations to generate only visualizations with two (X and Y position) and three (X and Y position and size) encoding channels. Other features were also fixed and binned to constrain the number of possibilities for example:

- The task supported by the visualization. In this work, we only focused on the task of identifying correlation among certain variables, thus in order to tag a visualization as "not helpful", it has to obscure the process of perceiving correlation.
- The visual mark or shape. Although several marks could be used to represent data (bars, circles, lines, areas, arcs, etc.) we constrained the generation process to only generate scatter plots (i.e., by fixing the visual mark to "circles").
- The scales' domain range. We binned the potential values of the visualization scales by relying on the dataset characteristics. The minimum value of a scale domain can take one of these values: the scale variable mean minus two-times the SD of that column in the dataset, the column's minimum value, the column's minimum value multiplied by 0.5 and zero. On the other hand, the maximum value could take one of these values: the scale variable mean plus two-times the SD of that column in the dataset, the column's maximum value, and the column's maximum value multiplied by 1.5. In the case of nominal variables, the domain will hold all the existing nominal values within that column of the dataset.

We used these reference values as the scales' domain ranges because it allowed us to generalize the generation process and adapt it to any dataset. The reason of including the mean plus/minus two-times the SD formula is because it allowed us to also account for the effect of the column's distribution on the visualization channels' scales.

The scales' range values are also set to a fixed width to prevent the number of generated visualizations to be unfeasible, but it is another important feature to test subsequently, as the physical space provided for a visualization can be also very influential for the perception of the displayed data.

In addition, we tested this approach with a tri-variate dataset with two numeric columns and one nominal column, so the combinations of displayed variables are also constrained; we will discuss in the last sections this decision and its implications in the development of the ML model.

Finally, we included a basic tagging infrastructure into the generated HTML file to ease the tagging process through two buttons that automatically save the configuration and the given tag and a text field to store notes regarding the tagging process. Figure 3 shows a screenshot of the generated tool.

Although the generation process is constrained to the aforementioned values and parameters, it is necessary to clarify that this proposal is applicable to any other kind of configuration. We wanted to simplify the visualization tagging process to test the approach, but subsequent work will focus on exploring the influence of other combination of features and analytic tasks.

3.3 | Training dataset generation: Visualization tagging process

We involved all the authors in the labelling process as a measure to mitigate subjectivity regarding the perception of the visualizations as helpful or not helpful. We initially proposed a Delphi method, an individual peer review approach to arrive at a group decision by individual tagging of the

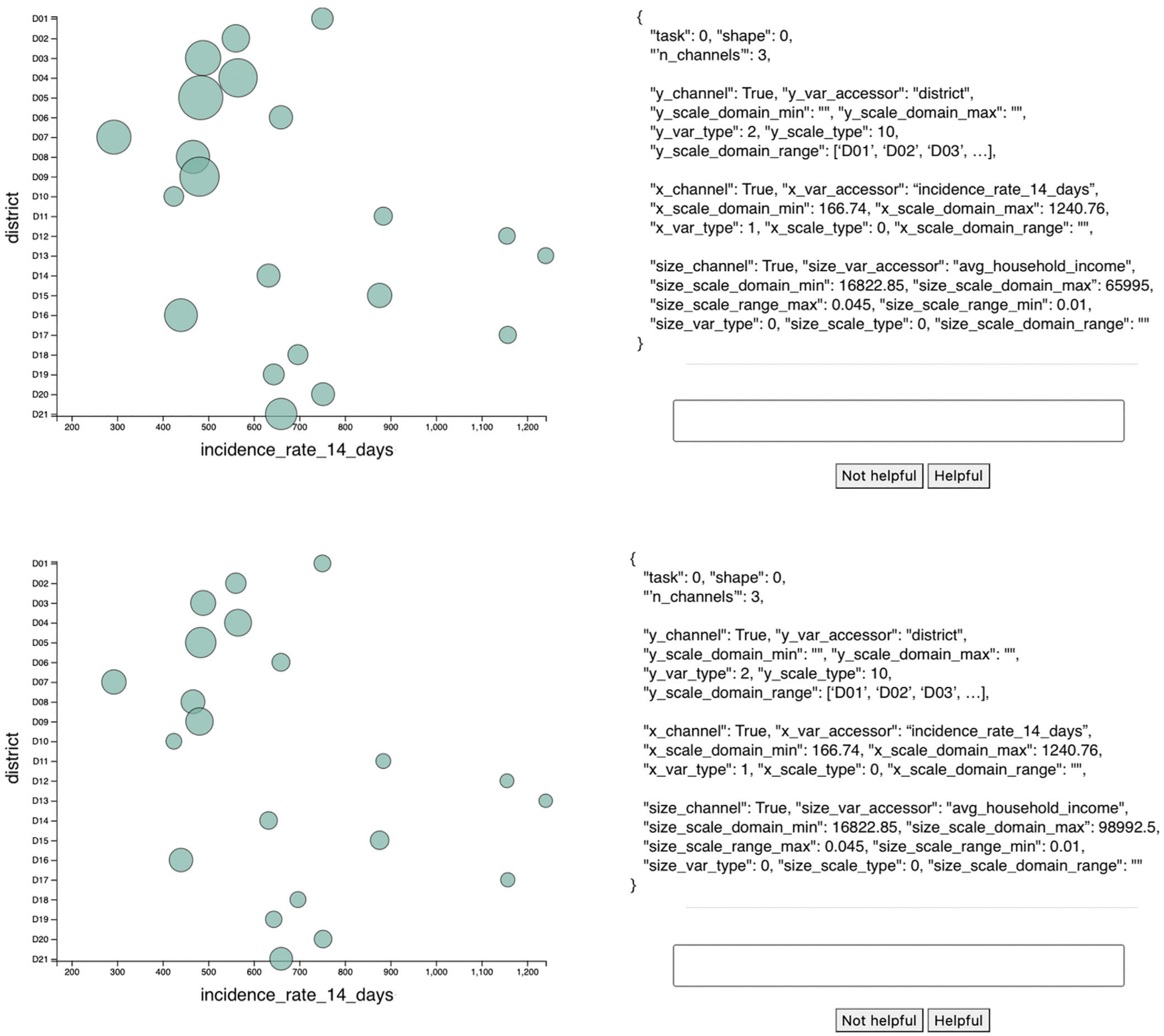


FIGURE 3 A demonstration of the generation process output, with the different visualizations (left side of the figure) followed by their configuration and the labelling functionalities (right side of the figure)

full set of generated visualizations and discuss over those visualizations where there was no agreement. However, the problem with this approach is that it eventually required re-examining almost all the visualizations to ensure that all the authors applied the same criteria. For this reason, we decided to apply the labeling process by involving all the experts at once.

The full process is divided into two phases (Figure 4), a first phase focused on generating the visualization dataset described above, and a second phase for the visualization tagging process. Both phases involve all the authors as experts in areas related to data management and visualization:

- A Ph.D. student whose doctoral dissertation deals with customizable dashboards to analyse and visualize any kind of data.
- A web developer and researcher with 12 years of experience in developing technological ecosystems and tools to support knowledge management and decision-making processes.
- A professor and researcher with 20 years of experience in human-computer interaction, and data visualization in different fields such as cinema and digital humanities.
- A professor and research with more than 20 years of experience in software engineering and human-computer interaction.

The first phase was an iterative process to constrain the generation of configurations in order to avoid the explosion of configuration combinations and ensure that the visualization dataset is suitable for manual labelling.

Regarding the tagging process, it was divided into two activities. First, experts defined the criteria to indicate when a visualization of the dataset is helpful or not helpful to achieve the defined task (identifying correlation). Experts took into account the different features that introduce misleading in a visualization for a particular analytic task and domain. They followed three steps: (1) Each expert formulated his criteria; (2) each expert reviewed the criteria of the other experts; (3) those criteria without agreement between the experts were reviewed and discussed to get an agreement.

The final criteria, based on the fine-grained features, the selected domain (COVID-19 incidence rate in Madrid) and the defined task (identifying correlation), was:

- Omitting data: the scale does not include all the data available in the dataset.
- Nominal scale in X or Y position for visualizing two variables: the nominal variable does not allow identifying correlation if we encode a numerical value on the X/Y position channel.
- Higher maximum: although a higher maximum allows identifying the correlation, it exaggerates the underlying correlation.
- Lower minimum: start the scale at 0 when it is not a possible value for the variable represented. This criterion depends on the domain. In particular, the average income per household for each district in Madrid cannot be 0, so it does not make sense to start the axis at that value.
- Improper scaling: the minimum and maximum values of the size scale domain difficult identifying correlation among the variables in the X and Y position.

Finally, all experts together, at the same time, reviewed each visualization and labelled it as 'helpful' or 'not helpful'. The group approach ensured the correct application of the criteria previously agreed.

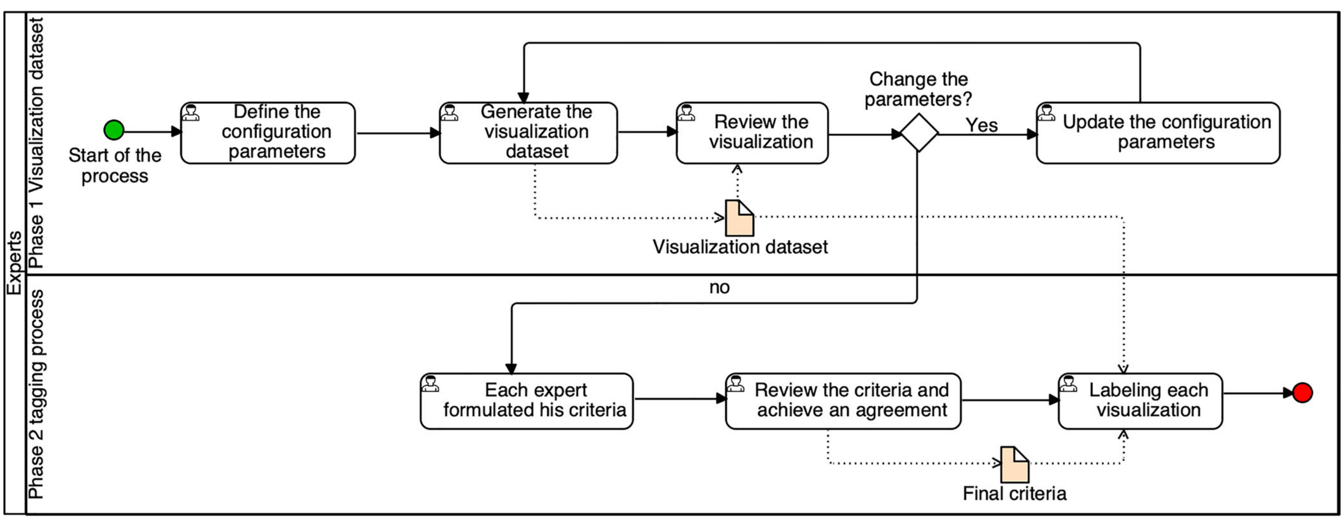


FIGURE 4 Method for training dataset generation

4 | ML APPROACH

4.1 | Input dataset

The outcome of the tagging process was an array of JSON objects containing the visualizations' generated features (through the process described in the Visualization generation section) with an additional Boolean attribute named "helpful", which indicates if the visualization with those features was considered helpful or not helpful during the tagging process. Table 1 shows the correspondence among the training dataset features and the meta-model, as well as the possible values that these attributes can have.

To sum up, the input/training dataset for the ML algorithm is a tabular dataset in which each row represents a visualization and columns represent the attributes of each visualization (i.e., shape, n_channels, x_channel, x_var_type, etc.). Figure 5 provides a detailed view of the dataset schema.

Scales' domains and ranges can be represented through an interval that goes from a minimum to a maximum value (as it happens with linear scales, for example), but also through a range of finite values (as it happens with ordinal scales). This is the reason why there could be empty values within these variables. The next section will explain the approach taken to tackle missing values found in the input dataset.

4.2 | Classification algorithm

The problem we want to solve can be framed as a binary classification task: given our set of visualization features and its supported task, we want to know if the visualization could be considered as helpful or not helpful.

As seen in the last subsection, visualizations are represented through a finite set of features based on the meta-model, meaning that several types of visualizations can be represented through this approach by changing the shape and/or channels' values, as well as other kind of tasks by modifying the "task" attribute. The main drawback is that this data schema leads to several missing values, because not every visualization will have all attributes for all channels, for example, a scatter plot might encode data through its X and Y channels but not through the size channel; however other scatter plot could have all three channels present.

This led to an unbalanced dataset in terms of number of features present for each record. To mitigate this issue, missing values were imputed and a binary flag was added to inform the model whether a value was originally missing.

In addition, another of the limitations of the generated dataset was the imbalance of the classification categories. As it will be explained in the next section, the majority of the configurations were classified as "not helpful" for the correlation identification task. To tackle this imbalance,

TABLE 1 Training dataset schema and its relationship with the meta-model

Variable	Possible values	Meta-model representation
Task Possible values based on (Amar et al., 2005)	Correlate, Cluster, Find Anomalies, Characterize Distribution, Determine Range, Sort, Find Extremum, Compute Derived Value, Filter, Retrieve Value	Task
Shape	Circle, Bar, Line, Area, Pie arc, Donut arc, Geographic, Text	Mark
N_channels	Derived from the number of channels present on the visualization	Derived from the number of instantiated channels
For every channel C ∈ (Y, X, size, colour, opacity, …)		Channel name attribute
C_channel	True if the channel is used to encode data in the visualization, False if not	Channel
C_var_type	Integer, Float, String	Variable type attribute
C_scale_type	Linear, Pow, Logarithmic, Squared, Time, Quantize, Quantile, Threshold, Ordinal	Scale type attribute
C_scale_domain_min	Number or NaN	Scale domain attribute
C_scale_domain_max	Number or NaN	Scale domain attribute
C_scale_domain_range	Range of values or NaN	Scale domain attribute
C_scale_range_min	Number or NaN	Scale range attribute
C_scale_range_max	Number or NaN	Scale range attribute
C_scale_range_range	Range of values or NaN	Scale range attribute

Training dataset

Shape	X_channel	X_var_type	X_scale_type	...	Size_channel	...	Size_scale_domain_range
Circle	True	String	Ordinal	...	True	...	NaN
...	...	...	...	...	...	...	...
Circle	True	Float	Linear	...	False	...	NaN

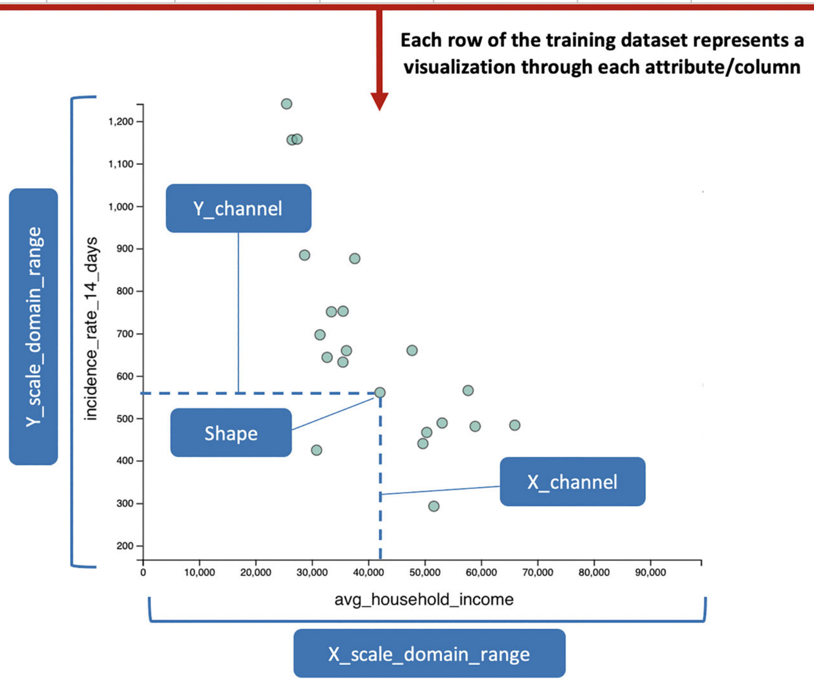


FIGURE 5 Representation of information visualizations through the training dataset

we performed a data up-sampling through the resampling module from Scikit-Learn (Pedregosa et al., 2011; Scikit-learn, 2017), using a replacement approach to match the majority class.

Different classification algorithms were considered to tackle this classification problem. Specifically, RF (Breiman, 2001), AdaBoost (Freund & Schapire, 1997), Support Vector Machines (Cortes & Vapnik, 1995) and Naïve Bayes (Friedman et al., 1997) classifiers.

According to our previous experiences, we believe that the possibility of explaining results and the accuracy desired for the classification provided by the RF classifier algorithm is the best fit for the posed classification issue. However, we compared the performance of RF against the aforementioned algorithms to rely on more evidence and be sure that the selected model is suitable for this classification problem.

4.3 | Proof of concept

To carry out the proof of concept for the presented approach, we chose a trending dataset, which the authors were familiar to. This last factor is important because knowing and being familiar to the data domain is crucial to correctly classify a visualization as helpful or not helpful (Cairo, 2019).

The chosen dataset is a tri-variate dataset regarding the COVID-19 incidence rate as of September 15, 2020 in the 21 Madrid (Spain) districts.[1] This dataset also includes data regarding the 2017 average income per household for each district[2]; these data regarding average income per household is the most recent data available through official sources.

It has been noted that Madrid districts with less average income have presented greater COVID-19 incidence than wealthier districts (Castellanos & Laudette, 2020). Thus, this tri-variate dataset presents a clear correlation example, which we used to classify automatically generated visualizations based on their capacity of conveying this statistical property in a straightforward and honest manner.

By following the methodology proposed in the previous section, we classified the generated visualizations involving the three variables of the dataset.

For example, the visualization presented in Figure 6 uses the Y position channel to encode the district of Madrid (nominal variable), the X position channel to encode the COVID-19 incidence rate as of September 15, 2020, and the size channel to encode the average income.

Following the criteria that we described in the methodology section, we labelled the Figure 6 visualization as "helpful," because it eases the recognition of a pattern for the fixed task (identifying correlation). Districts with lower incidence rates (at the leftmost section of the X axis) have greater average income per household (which is why their visual marks are bigger). In this case it is important to understand that the correlation is spotted through the size channel and X position channel (the bigger the circles, the less the incidence rate) and not between the X and Y position channels. In fact, is not possible to spot correlation between X and Y values because the Y position encodes a nominal value.

However, the generated visualization shown in Figure 7 was labelled as "not helpful", because, although the size channel stills encoding the same data, the minimum and maximum values of the size scale domain (in this case, the column's mean minus two times the column's SD and the column's maximum multiplied by 1.5, respectively) do not ease the recognition of the correlation pattern present in the dataset, because the differences among the districts' income is not as clearer as in Figure 7.

The labelling process was repeated for a total of 960 generated visualizations by all authors, in order to ensure that we were following the same criteria and to avoid the bias that could arise if only one person was in charge of tagging all visualizations. Although we already believed that the labelling process could be vulnerable to subjectivity, we verified that assumption while carrying this experiment out, as we will describe in the limitations section.

Once all visualizations were labelled, we processed the data through a Jupyter Notebook (Kluyver et al., 2016). From the 960 labelled visualizations, 828 were labelled as "not helpful", while 132 were labelled as "helpful", but as explained in the methodology section, we resampled the dataset to balance the two categories, obtaining a dataset of 1656 labelled visualizations.

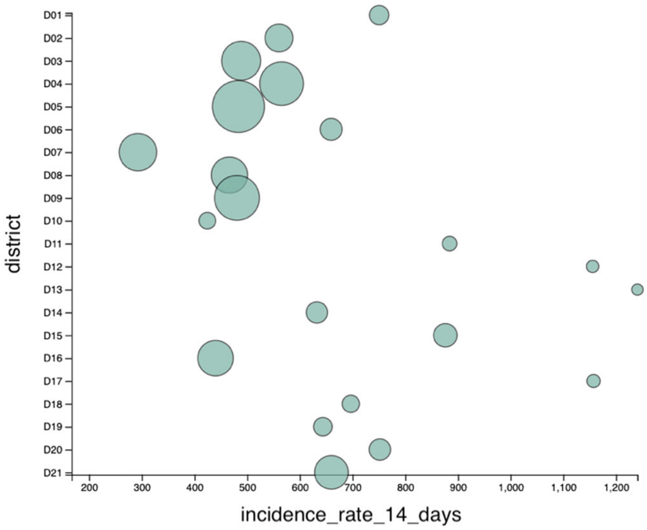


FIGURE 6 Example of a visualization that eases the recognition of correlation among two variables through X position and size encodings

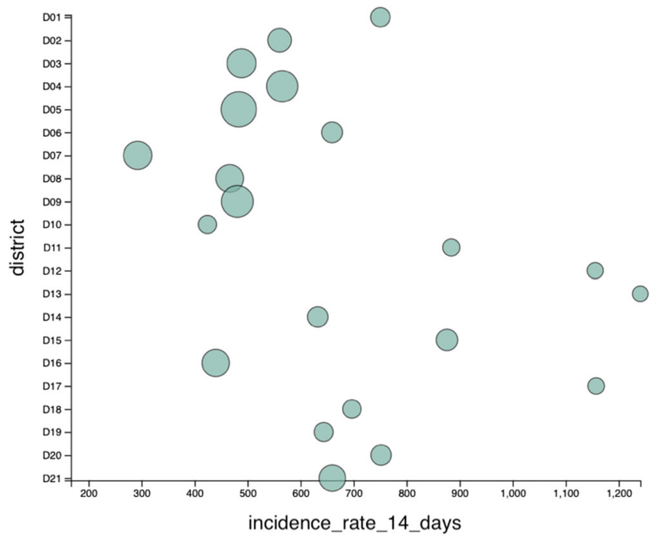


FIGURE 7 Example of a visualization that is not very helpful for the recognition of correlation among two variables through X position and size encodings

The training set size was a 70% of the whole dataset, leaving a 30% for testing.

Regarding the performance of the different algorithms, the following results were obtained: Tables 2, 3, 4 and 5 show the results derived from the Naïve Bayes, SVM, AdaBoost and Random Forest classifiers.

TABLE 2 Classification report of the Naïve Bayes classifier

Class	Precision	Recall	F1-score
0	0.66	0.79	0.72
1	0.73	0.59	0.65
Accuracy: 0.688			

TABLE 3 Classification report of the SVM classifier

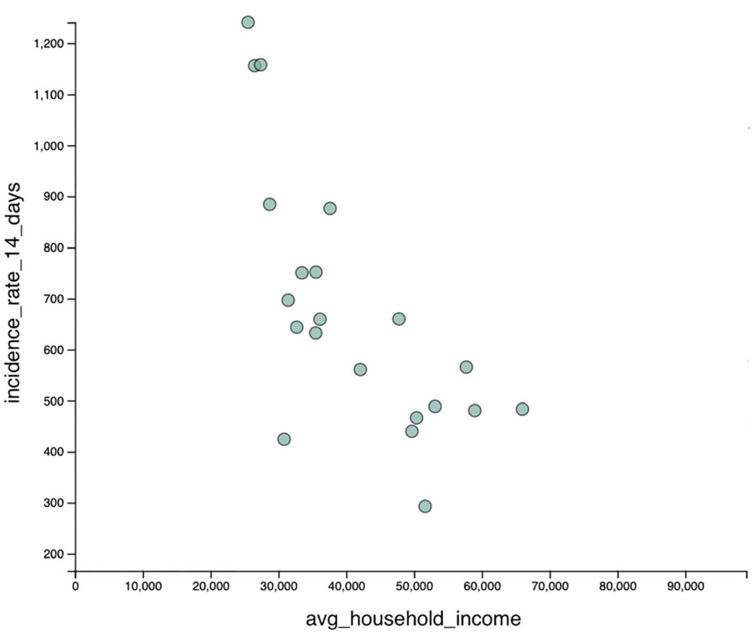
Class	Precision	Recall	F1-score
0	0.71	0.98	0.82
1	0.97	0.59	0.73
Accuracy: 0.786			

TABLE 4 Classification report of the AdaBoost classifier

Class	Precision	Recall	F1-score
0	0.94	1.00	0.97
1	1.00	0.94	0.97
Accuracy: 0.969			

TABLE 5 Classification report of the random forest classifier

Class	Precision	Recall	F1-score
0	0.98	1.00	0.99
1	1.00	0.98	0.99
Accuracy: 0.989			



{

"task": 0, "shape": 0,
"'n_channels'": 3,

"y_channel": True,
"y_var_accessor": "incidence_rate_14_days",
"y_scale_domain_min": 166.74, "y_scale_domain_max": 1240.76,
"y_var_type": 1, "y_scale_type": 0,
"y_scale_domain_range": "",

"x_channel": True,
"x_var_accessor": "avg_household_income",
"x_scale_domain_min": 0, "x_scale_domain_max": 98992.5,
"x_var_type": 0, "x_scale_type": 0,
"x_scale_domain_range": "",

"size_channel": False, "size_var_accessor": "",
"size_scale_domain_min": "", "size_scale_domain_max": "",
"size_scale_range_max": "", "size_scale_range_min": "",
"size_var_type": "", "size_scale_type": "",
"size_scale_domain_range": ""

}

FIGURE 8 Example of a visualization (and its configuration) that exaggerates the underlying correlation

```
prediction = clf.predict([[0.,              # task
                  0.,                       # shape
                  2.,                       # n_channels
                  1., 100., 1300.,          # y_channel, min, max
                  1., 0., 100000.,          # x_channel, min, max
                  0., 0., 0.,               # size_channel, min, max
                  1., 10.,                  # y_var_type, scale_type
                  0., 0.,                   # x_var_type, scale_type
                  -1., -1.,                 # size_var_type, scale_type
                  False, False,             # NA indicators X, Y, size domains
                  False, False,
                  True, True]])

if prediction[0] == 1:
    print("Not helpful.")
else:
    print("Helpful.")
```

```
Not helpful.
```

FIGURE 9 Prediction output for a visualization with the characteristics from Figure 8

```
prediction = clf.predict([[0.,              # task
                  0.,                       # shape
                  2.,                       # n_channels
                  1., 100., 1300.,          # y_channel, min, max
                  1., 20000., 66000.,       # x_channel, min, max
                  0., 0., 0.,               # size_channel, min, max
                  1., 10.,                  # y_var_type, scale_type
                  0., 0.,                   # x_var_type, scale_type
                  -1., -1.,                 # size_var_type, scale_type
                  False, False,             # NA indicators X, Y, size domains
                  False, False,
                  True, True]])

if prediction[0] == 1:
    print("Not helpful.")
else:
    print("Helpful.")
```

```
Helpful.
```

FIGURE 10 Prediction output for a visualization with the characteristics from Figure 8, but with a variation on the X-axis scale domain range

The RF algorithm outperformed the rest, followed by the AdaBoost classifier. Given these results, the RF classifier was chosen for test the outcomes of individual predictions by introducing other visualizations' values manually. The predictions yielded the expected result.

For example, if we use the configuration of the Figure 8 visualization as a prediction input, the model will yield that the visualization is not helpful (Figure 9). In fact, thanks to the interpretability provided by decision trees built through the RF classifier, we know that the visualization was rejected because the correlation is exaggerated by configuring extreme values at the X-axis.

If the X scale of the visualization is modified to represent values closer to the displayed variable's ranges (Figure 10), then the result would be that the visualization is helpful, because the correlation can be spotted and it is not exaggerated as it was in Figure 8

5 | DISCUSSION

Through this work, we wanted to test the viability of the application of ML algorithms to information visualizations' fine-grained features.

We compared different classification approaches, but the algorithm that performed better was the RF algorithm. The decision of choosing the RF approach was also supported by the problem set up: the variety of the features that might be present for each record can lead to several quantities of missing values, and RF provides support for handling them (Tang & Ishwaran, 2017).

We also considered another approach: to train a model for each type of visualization in terms of number of channels, i.e., a model for visualizations that encode data through the X and Y channels, a model for visualizations that encode data through the X, Y and size channels, and so on. However, given the fact that we want to test in the future several combinations of channels and visualization types, this approach would be impractical. For these reasons, we decided to train a single model with a single dataset that holds a column for every potential channel and imputed the missing values adding a missing indicator.

Another characteristic of RF that make the algorithm suitable for the approach is the possibility of explaining results through their measures of variable importance. The importance of features is useful not only to predict if a visualization is or not helpful, but also to raise attention on which characteristics influence not properly designed visualizations the most.

Although the accuracy of the RF model is very high, it needs to be thoroughly discussed. One of the reasons of the high accuracy is due to the fact that the model is mimicking the criteria we previously defined to classify each visualization. However, it is clear that the model has learned the most important features from the classification process. The top five important features were the following:

- x_scale_domain_max: 0.332726
- y_scale_domain_max: 0.230253
- size_scale_domain_max: 0.137168
- size_scale_domain_min: 0.078635
- y_scale_domain_min: 0.066706 (tightly followed by x_scale_domain_min)

This result aligns with previous research found in the literature, in which the definition of a visualization scales' ranges is determining for its proper understanding (Pandey et al., 2014).

It is important to note that the model accuracy is restricted to the training set of visualizations. As it will be explained in the limitations section, several visualizations have been generated as a part of the training set; however, the generated visualizations are still limited by the constraints we applied to avoid the generation of a non-viable quantity of visualizations to label. This means that the model would not be able to generalize its predictions if a visualization with characteristics that are not present in the training set is handed as an input, which is an important drawback to take into account.

Also, in this proof of concept we only tested the influence of the X position, Y position and size channel scales' domain on the usefulness of scatter charts to identify correlation. Other combination of channels (colour, opacity, etc.), visual mark types (bars, lines, areas, etc.) and analytic tasks need to be explored and tagged to train a more complete model.

On the other hand, using the meta-modelling approach as a base methodology for this work provided two main benefits. First, following the model-driven development and the SPL engineering, we managed to automatically generate the visualizations that we later labelled to build our training set. And second, the meta-model provided the fine-grained features and relationships to structure the training dataset schema, which was a crucial step before applying any ML algorithm.

The precision and accuracy metrics show that the resulting ML model has indeed learn from the implicit expertise and heuristics that we used to manually label the training set. The criteria followed might be seen as obvious or very basic (such as not exaggerating the scale values or using certain encodings), but our future work is focused on developing an automatic detector of data visualization potential design flaws, so that novices or not skilled users are aware of latent misleading configurations they are unconsciously introducing in their designs.

Although we encountered limitations during the execution of this experiment, we think this approach is an interesting starting point for applying ML models to the elements that compose information visualizations to automate classification tasks such as the one presented in this work.

We want to highlight that our main goal for this experiment was not to obtain a perfect model with an outstanding prediction accuracy, because more training data need to be generated and labelled to obtain a more reliable and robust model. The goal was to test the viability of the proposed approach and to identify the limitations and drawbacks that could arise, in order to set the foundations of future research lines focused on the refinement of the proposal.

6 | LIMITATIONS

Some limitations were identified during the design and execution of this proposal. First, to train a ML model it is necessary to rely on dataset with significant records. We decided to generate the training set and label each visualization manually because it gave us the freedom to define the data structure by means of a previously validated meta-model and because this method allowed us to generate great quantities of visualizations automatically.

However, we identified the first limitation, which is the subjectivity and bias related to the labelling process. We were conscious about this limitation because, during the labelling, several discussions regarding the classification of a visualization as “helpful” or “not helpful” arose. We tried to mitigate this bias by involving all authors in the labelling process, but, this approach is inevitably vulnerable to subjectivity because the interpretation of visualization is vulnerable to subjectivity and personal perceptions itself (Dimara et al., 2018; Hullman et al., 2011; Y.-S. Kim et al., 2018; Valdez et al., 2017; Wall et al., 2017).

On the other hand, we only use one tri-variate dataset from one specific domain related to the COVID-19 incidence. This means that the model is only useful for datasets with similar characteristics to the one used for training the model. The data domain is extremely crucial for identifying if a visualization is well designed (Berinato, 2016; Cairo, 2019; Tufte & Graves-Morris, 2014), so testing this approach in a variety of data domains is necessary to obtain a practical model.

Finally, we only tested the "identify correlation" task, but this approach needs to be tested with other visualization tasks such as identifying individual values, comparing categories, identifying outliers, etc. (Amar et al., 2005), because the "helpful/not helpful" tag could vary depending on the analytic task.

7 | CONCLUSIONS

This work presents a proof of concept of a novel ML application on the data visualization domain. Data visualizations encode data through different visual features, which have been captured and structured through a meta-modelling approach. The identified features were employed to automatically generate a set of parameterized visualizations that were subsequently discussed through a tagging process to obtain a training dataset of "helpful/not helpful" information visualizations. Finally, the resulting dataset was employed to train ML algorithms that classify information visualizations as helpful or not helpful given their features and supported analytic task.

The experiment shows promising results as the viability of the approach has been tested through a proof-of-concept in the domain of visualizations that display tri-variate datasets with the goal of identifying correlation among their variables. Although some limitations were identified, this experiment can set the foundations for subsequent research on this domain.

It is important to clarify that this proposal does not aim to replace the work of data journalists and other experts but helping them and other people to use the available know-how regarding confusing visualizations to facilitate the selection of good visualizations depending on the domain and context. This proposal is also focused on supporting other existing tools to detect misinformation and fake news across social networks and digital media.

Future research lines will try to address the limitations encountered during the execution of the experiment and to refine the proposal with the goal of obtaining a tool for classifying information visualizations based on its most primitive components. Despite the proposal could be used as a standalone tool, it could be also used as Software as a Service (SaaS) or included as a component in other existed tools to improve the detection of fake news to non-expert users.

ACKNOWLEDGEMENTS

This research work has been supported by the Spanish Ministry of Education and Vocational Training under an FPU fellowship (FPU17/03276).

CONFLICT OF INTEREST

The author declares there is no potential conflicts of interest.

ENDNOTES

[1] Official data source from the Madrid's government open data portal: https://datos.comunidad.madrid/catalogo/dataset/covid19_tia_muni_y_distritos/resource/f22c3f43-c5d0-41a4-96dc-719214d56968

[2] Official data source from the Spanish National Institute of Statistics: https://www.ine.es/jaxiT3/Tabla.htm?t=31097

DATA AVAILABILITY STATEMENT

The data that support the findings of this study are available from the corresponding author upon reasonable request.

ORCID

Andrea Vázquez-Ingelmo https://orcid.org/0000-0002-7284-5593
Alicia García-Holgado https://orcid.org/0000-0001-9663-1103
Francisco José García-Peñalvo https://orcid.org/0000-0001-9987-5584
Roberto Therón https://orcid.org/0000-0001-6739-8875

REFERENCES

Álvarez, J. M., Evans, A., & Sammut, P. (2001). Mapping between levels in the metamodel architecture. *Paper presented at the International Conference on the Unified Modeling Language*, *2185*, 34–46.

Amar, R., Eagan, J., & Stasko, J. (2005). Low-level components of analytic activity in information visualization. *Paper presented at the IEEE symposium on information visualization, 2005, INFOVIS*, *1*, 111–117.

Andrews, R. (2019). *Info we trust: How to inspire the world with data*. John Wiley & Sons.

Arango, G. F. (1988). *Domain engineering for software reuse*. UC Irvine: Donald Bren School of Information and Computer Sciences.

Arthur, M. P. (2020). Automatic source code documentation using code summarization technique of NLP. *Procedia Computer Science*, *171*, 2522–2531. https://doi.org/10.1016/j.procs.2020.04.273

Berinato, S. (2016). *Good charts: The HBR guide to making smarter, more persuasive data visualizations*. Harvard Business Review Press.

Bostock, M., Ogievetsky, V., & Heer, J. (2011). D^3 data-driven documents. *IEEE Transactions on Visualization and Computer Graphics*, *17*(12), 2301–2309.

Breiman, L. (2001). Random forests. *Machine Learning*, *45*(1), 5–32. https://doi.org/10.1023/a:1010933404324

Cairo, A. (2019). *How charts lie: Getting smarter about visual information*. WW Norton & Company.

Castellanos, S., & Laudette, C.-L. (2020). Madrid region orders partial lockdown in poorer areas hit by COVID-19. *Reuters*. https://www.reuters.com/article/us-health-coronavirus-spain-madrid/madrid-region-orders-partial-lockdown-in-poorer-areas-hit-by-covid-19-idUSKBN26927C

Clements, P., & Northrop, L. (2002). *Software product lines*. Addison-Wesley.

Correll, M., Bertini, E., & Franconeri, S. (2020). Truncating the Y-Axis: Threat or menace? *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. Association for Computing Machinery, 1, 1–12.

Cortes, C., & Vapnik, V. (1995). Support-vector networks. *Machine Learning*, *20*(3), 273–297.

Dibia, V., & Demiralp, Ç. (2019). Data2Vis: Automatic generation of data visualizations using sequence to sequence recurrent neural networks. *IEEE Computer Graphics and Applications*, *39*(5), 33–46.

Dimara, E., Franconeri, S., Plaisant, C., Bezerianos, A., & Dragicevic, P. (2020). A Task-Based Taxonomy of Cognitive Biases for Information Visualization. *IEEE transactions on visualization and computer graphics*, *26*(2), 1413–1432. https://doi.org/10.1109/TVCG.2018.2872577

Freund, Y., & Schapire, R. E. (1997). A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, *55*(1), 119–139.

Friedman, N., Geiger, D., & Goldszmidt, M. (1997). Bayesian network classifiers. *Machine Learning*, *29*(2–3), 131–163.

Gotz, D., & Wen, Z. (2009). Behavior-driven visualization recommendation. *Paper Presented at the Proceedings of the 14th International Conference on Intelligent User Interfaces*, *1*, 315–324.

Halim, Z., Kalsoom, R., Bashir, S., & Abbas, G. (2016). Artificial intelligence techniques for driving safety and vehicle crash prediction. *Artificial Intelligence Review*, *46*(3), 351–387. https://doi.org/10.1007/s10462-016-9467-9

Halim, Z., & Rehan, M. (2020). On identification of driving-induced stress using electroencephalogram signals: A framework based on wearable safety-critical scheme and machine learning. *Information Fusion*, *53*, 66–79. https://doi.org/10.1016/j.inffus.2019.06.006

Hu, K., Bakker, M. A., Li, S., Kraska, T., & Hidalgo, C. (2019). VizML: A machine learning approach to visualization recommendation. *Paper presented at the proceedings of the 2019 CHI conference on human factors in computing systems*, *1*, 1–12.

Hullman, J., Adar, E., & Shah, P. (2011). The impact of social information on visual judgments. In *Paper presented at the proceedings of the SIGCHI conference on human factors in computing systems*. Association for Computer Machinery, 1, 1461–1470.

Kang, K. C., Cohen, S. G., Hess, J. A., Novak, W. E., & Peterson, A. S. (1990). *Feature-oriented domain analysis (FODA) feasibility study*. Retrieved from Pittsburgh.

Kaur, P., & Owonibi, M. (2017). A review on visualization recommendation strategies. *Paper Presented at the 12th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (Visigrapp 2017)*, *1*, 266–273.

Keim, D., Andrienko, G., Fekete, J.-D., Görg, C., Kohlhammer, J., & Melançon, G. (2008). Visual analytics: Definition, process, and challenges. In *Information visualization* (pp. 154–175). Springer.

Kim, Y., & Heer, J. (2018). Assessing effects of task and data distribution on the effectiveness of visual encodings. *Paper presented at the Computer Graphics Forum*, *37*, 157–167.

Kim, Y.-S., Reinecke, K., & Hullman, J. (2018). Data through others' eyes: The impact of visualizing others' expectations on visualization interpretation. *IEEE Transactions on Visualization and Computer Graphics*, *24*(1), 760–769.

Kleppe, A. G., Warmer, J., & Bast, W. (2003). MDA explained. The model driven architecture: Practice and promise. In: Addison-Wesley Longman Publishing Co., Inc.

Kluyver, T., Ragan-Kelley, B., Pérez, F., Granger, B., Bussonnier, M., Frederic, J., Kelley, K., Hamrick,, J., Grout, J., & Corlay, S. (2016). *Jupyter notebooks—A publishing format for reproducible computational workflows* (p. 87). Positioning and Power in Academic Publishing.

Lewandowsky, S., Ecker, U. K. H., & Cook, J. (2017). Beyond misinformation: Understanding and coping with the "post-truth" era. *Journal of Applied Research in Memory and Cognition*, *6*(4), 353–369. https://doi.org/10.1016/j.jarmac.2017.07.008

Liu, X., Faes, L., Kale, A. U., Wagner, S. K., Fu, D. J., Bruynseels, A., Mahendiran, T., Moraes, G., Shamdas, M., Kern, C., Ledsam, J. R., Schmid, M. K., Balaskas, K., Topl, E. J., Bachmann, L. M., Keane, P. A., & Denniston, A. K. (2019). A comparison of deep learning performance against health-care professionals in detecting diseases from medical imaging: A systematic review and meta-analysis. *The lancet digital health*, *1*(6), e271–e297.

Mackinlay, J., Hanrahan, P., & Stolte, C. (2007). Show me: Automatic presentation for visual analysis. *IEEE Transactions on Visualization Computer Graphics*, *13*(6), 1137–1144.

Maletic, J. I., Marcus, A., & Collard, M. L. (2002). A task oriented view of software visualization. *Paper presented at the Proceedings First International Workshop on Visualizing Software for Understanding and Analysis*, *1*, 32–40.

Meeks, E. (2018). D3. Js in Action: Data Visualization with JavaScript: Manning.

Metzger, A., & Pohl, K. (2007). Variability management in software product line engineering. *Paper Presented at the Companion to the Proceedings of the 29th International Conference on Software Engineering*, *1*, 186–187.

Muhammad, T., & Halim, Z. (2016). Employing artificial neural networks for constructing metadata-based model to automatically select an appropriate data visualization technique. *Applied Soft Computing*, *49*, 365–384. https://doi.org/10.1016/j.asoc.2016.08.039

Pandey, A. V., Manivannan, A., Nov, O., Satterthwaite, M., & Bertini, E. (2014). The persuasive power of data visualization. *IEEE Transactions on Visualization and Computer Graphics*, *20*(12), 2211–2220.

Pandey, A. V., Rall, K., Satterthwaite, M. L., Nov, O., & Bertini, E. (2015). How deceptive are deceptive visualizations? An empirical analysis of common distortion techniques. *Paper presented at the Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, *1*, 1469–1478.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., & Dubourg, V. (2011). Scikit-learn: Machine learning in python. *Journal of Machine Learning Research*, *12*(Oct), 2825–2830.

Pohl, K., Böckle, G., & Van der Linden, F. J. (2005). *Software product line engineering: Foundations, principles and techniques*. Springer-Verlag.

Saket B., Endert A., & Demiralp C. (2019). Task-Based Effectiveness of Basic Visualizations. *IEEE Transactions on Visualization and Computer Graphics*, *25*(7), 2505–2512. https://doi.org/10.1109/tvcg.2018.2829750

Scikit-learn. (2017). API Reference - scikit-learn documentation: Metrics.

Shu, K., Mahudeswaran, D., & Liu, H. (2019). FakeNewsTracker: A tool for fake news collection, detection, and visualization. *Computational and Mathematical Organization Theory*, *25*(1), 60–71. https://doi.org/10.1007/s10588-018-09280-3

Tang, F., & Ishwaran, H. (2017). Random forest missing data algorithms. *Statistical Analysis and Data Mining: The ASA Data Science Journal*, *10*(6), 363–377.

Tufte, E., & Graves-Morris, P. (2014). *The visual display of quantitative information.*; *1983*. Graphics Press.

Valdez, A. C., Ziefle, M., & Sedlmair, M. (2017). *A framework for studying biases in visualization research*. DECISIVe 2017.

Vartak, M., Huang, S., Siddiqui, T., Madden, S., & Parameswaran, A. (2017). Towards visualization recommendation systems. *ACM SIGMOD Record*, *45*(4), 34–39.

Vázquez-Ingelmo, A., García-Holgado, A., García-Peñalvo, F. J., & Therón, R. (2019). Dashboard meta-model for knowledge management in technological ecosystem: A case study in healthcare. *Paper Presented at the UCAmI 2019*, *Toledo, Castilla-La Mancha*, *31*(1), 44–57.

Vázquez-Ingelmo, A., García-Holgado, A., García-Peñalvo, F. J., & Therón, R. (2020). A meta-model integration for supporting knowledge discovery in specific domains: A case study in healthcare. *Sensors*, *20*(15), 4072.

Vázquez-Ingelmo, A., García-Peñalvo, F. J., & Therón, R. (2018). Application of domain engineering to generate customized information dashboards. *Paper presented at the HCI International 2018. 20th Conference on Human-Computer Interaction (15–20 July 2018)*. LCT 2018: Learning and Collaboration Technologies. Learning and Teaching, 10925, 518–529.

Vázquez-Ingelmo, A., García-Peñalvo, F. J., & Therón, R. (2019c). Capturing high-level requirements of information dashboards' components through meta-modeling. *Paper presented at the 7th International Conference on Technological Ecosystems for Enhancing Multiculturality (TEEM 2019)*, *1*, 815–821.

Vázquez-Ingelmo, A., García-Peñalvo, F. J., & Therón, R. (2019d). Taking advantage of the software product line paradigm to generate customized user interfaces for decision-making processes: A case study on university employability. *PeerJ Computer Science*, *5*, e203. https://doi.org/10.7717/peerj-cs.203

Vázquez-Ingelmo, A., García-Peñalvo, F. J., Therón, R., & Conde González, M. Á. (2019). Extending a dashboard meta-model to account for users' characteristics and goals for enhancing personalization. *Paper presented at the Learning Analytics Summer Institute (LASI)*. CEUR, 2415, 35–42.

Vázquez-Ingelmo, A., García-Peñalvo, F. J., Therón, R., & Conde, M. Á. (2020). Representing data visualization goals and tasks through meta-modeling to tailor information dashboards. *Applied Sciences*, *10*(7), 2306.

Viegas, F. B., Wattenberg, M., Van Ham, F., Kriss, J., & McKeon, M. (2007). Manyeyes: A site for visualization at internet scale. *IEEE Transactions on Visualization and Computer Graphics*, *13*(6), 1121–1128.

Voigt, M., Pietschmann, S., Grammel, L., & Meißner, K. (2012). Context-aware recommendation of visualization components. *Paper presented at the The Fourth International Conference on Information, Process, and Knowledge Management (eKNOW)*, *1*, 101–109.

Wall, E., Blaha, L. M., Franklin, L., & Endert, A. (2017). Warning, bias may occur: A proposed approach to detecting cognitive bias in interactive visual analytics. *Paper presented at the 2017 IEEE Conference on Visual Analytics Science and Technology (VAST)*, *1*, 104–115.

Ware, C. (2012). *Information visualization: Perception for design*. Elsevier.

Wongsuphasawat, K., Moritz, D., Anand, A., Mackinlay, J., Howe, B., & Heer, J. (2015). Voyager: Exploratory analysis via faceted browsing of visualization recommendations. *IEEE Transactions on Visualization and Computer Graphics*, *22*(1), 649–658.

AUTHOR BIOGRAPHIES

Andrea Vázquez-Ingelmo received the bachelor's degree in computer engineering from the University of Salamanca, Salamanca, in 2016 and the master's degree in computer engineering from the same university in 2018. She is a member of the Research Group of Interaction and eLearning (GRIAL), where she is pursuing her PhD degree in computer sciences. Her area of research is related to human-computer interaction, software engineering, information visualization and machine learning applications.

Alicia García-Holgado She received the degree in Computer Sciences (2011), a M.Sc. in Intelligent Systems (2013) and a Ph.D. (2018) from the University of Salamanca, Spain. She is member of the GRIAL Research Group of the University of Salamanca since 2009. Her main lines of research are related to the development of technological ecosystems for knowledge and learning processes management in heterogeneous contexts, and the gender gap in the technological field. She has participated in many national and international R&D projects. She is a member of IEEE (Women in Engineering, Education Society and Computer Society), ACM (and ACM-W) and AMIT (Spanish Association for Women in Science and Technology).

Francisco José García-Peñalvo He received the degrees in computing from the University of Salamanca and the University of Valladolid, and a Ph.D. from the University of Salamanca (USAL). He is Full Professor of the Computer Science Department at the University of Salamanca. In addition, he is a Distinguished Professor of the School of Humanities and Education of the Tecnológico de Monterrey, Mexico. Since 2006 he is the head of the GRIAL Research Group GRIAL. He is head of the Consolidated Research Unit of the Junta de Castilla y León (UIC 81). He was Vice-dean of Innovation and New Technologies of the Faculty of Sciences of the USAL between 2004 and 2007 and Vice-Chancellor of Technological Innovation of this University between 2007 and 2009. He is currently the Coordinator of the PhD Programme in Education in the Knowledge Society at USAL. He is a member of IEEE (Education Society and Computer Society) and ACM.

Roberto Therón He holds a PhD in Computer Science (2002, University of Salamanca - USAL), a B.A in Audiovisual Communication (2007) and a B.A in Humanities (2008). He works as an Full Professor at the Computer Science Department of the USAL, and he is a researcher of the GRIAL Group. He is the leader of the visual analytics research group, which has carried out several research projects related to various domains related to big data problems, including bioinformatics, palaeoclimatology and digital humanities (collaborations with the Real Academia Española and Austrian Academy of Science). He has proposed and developed several highly interactive visual analytics tools that tackle the problem of understanding complex and vast datasets by integrating the cognitive capabilities of the domain experts with specific machine learning and data mining techniques suitable for the issue at hand. Since 2012 he is the coordinator of the Master on Intelligent Systems (both at the University of Salamanca).

How to cite this article: Vázquez-Ingelmo, A., García-Holgado, A., García-Peñalvo, F. J., & Therón, R. (2021). Proof-of-concept of an information visualization classification approach based on their fine-grained features. *Expert Systems*, e12872. https://doi.org/10.1111/exsy.12872

7.26 Appendix Z. A Meta-Modeling Approach To Take Into Account Data Domain Characteristics and Relationships In Information Visualizations

Vázquez-Ingelmo, A., García-Holgado, A., García-Peñalvo, F. J., & Therón, R. (2021). A Meta-modeling Approach to Take into Account Data Domain Characteristics and Relationships in Information Visualizations. In Á. Rocha, H. Adeli, G. Dzemyda, F. Moreira, & A. M. Ramalho Correia (Eds.), Trends and Innovations in Information Systems and Technologies, WorldCIST 2021 (Vol. 2, pp. 570-580). Springer Nature. https://doi.org/10.1007/978-3-030-72651-5_54

A Meta-modeling Approach to Take into Account Data Domain Characteristics and Relationships in Information Visualizations

Andrea Vázquez-Ingelmo[1](✉), Alicia García-Holgado[1], Francisco J. García-Peñalvo[1], and Roberto Therón[1,2]

[1] GRIAL Research Group, Computer Sciences Department, Research Institute for Educational Sciences, University of Salamanca, Salamanca, Spain
{andreavazquez,aliciagh,fgarcia,theron}@usal.es
[2] VisUSAL Research Group, University of Salamanca, Salamanca, Spain

Abstract. Visual explanations are powerful means to convey information to large audiences. However, the design of information visualizations is a complex task, because a lot of factors are involved (the audience profile, the data domain, etc.). The complexity of this task can lead to poor designs that could make users reach wrong conclusions from the visualized data. This work illustrates the process of identifying features that could make an information visualization confusing or even misleading with the goal of arranging them into a meta-model. The meta-model provides a powerful resource to automatically generate information visualizations and dashboards that take into account not only the input data, but also the audience's characteristics, the available data domain knowledge and even the data context.

Keywords: Data visualization · Information visualization · Misleading visualizations · Feature identification · Meta-modeling

1 Introduction

Visual explanations are everywhere: they convey complex information, raise attention over target topics, improves the understandability of certain domains, etc. They can take the form of infographics, simple graphs, or even elaborated information visualizations.

Visual explanations are very powerful, because they let users visually perceive information in order to generate knowledge. However, information visualizations might turn out to be a double-edged sword.

The persuasive power of visualizations [1] has its benefits (better data understanding, more attention and focus on the information, etc.) but they also can lead users to wrong conclusions. Wrong conclusions are not always predictable, because they can have its origin on a unproper data visualization design, but also be influenced by the end users' prior beliefs, biases or polarization regarding certain topics.

It is important to take all these factors into account in order to provide properly designed and honest methods of visualization, because the main goal must be focused on how the user perceives and processes the displayed data.

These factors can be taken into account through domain expertise, i.e., information visualization experts that also have knowledge regarding the visualization's data domain and can provide a well-designed product through its expertise.

However, it is very difficult that every professional that makes use of information visualizations to convey information has these levels of expertise or domain knowledge, because it might be time-consuming and visual explanations usually need to be delivered quickly (for example, for covering news stories).

For these reasons, in this paper we propose an approach based on meta-modeling in which the data domain characteristics and relationships among variables are accounted for. The goal of this work is to characterize domain expertise to include it as a part of a generative pipeline to automatically develop information visualizations and dashboards. This approach can assist novices or practitioners without a significant level of the data domain knowledge to select the best parameters for their information visualizations. To sum up, the main contribution of this paper is a new version of a meta-model for instantiating information visualizations taking into account data domain and expertise.

The rest of this paper is organized as follows. Section 2 describes the methodology employed to carry out the meta-model and the automatic generation of dashboards. Section 3 outlines the modification of a previously developed dashboard meta-model to hold information about the data domain and the data context. Section 4 presents a proof-of-concept of the visualization generation using domain knowledge. Finally, Sect. 5 discusses the results and Sect. 6 offers the conclusions derived from this work.

2 Materials and Methods

2.1 Identification of Features

Usually, the definition of an information visualization involves human-computer interaction experts specialized in visualization and also experts from the data domain. Their know-how avoids the development of misleading visualizations. The automatic development of information visualizations has to take into account the experts' know-how.

In particular, it requires the identification of the features that has an impact in the correct visualization of the data depending on the dataset and the goals related to the analysis of that data. The process to identify these features has provided the base to define the meta-modeling proposal described in this work.

The study covers a set of phases based on an experimental approach as a way to discover the features that has an impact in the automatic development of no misleading information visualizations. Figure 1 presents the three phases: testing, analysis and solution.

The first phase, testing, is an experimental phase. Two experiments were set up to see what happens when some features are automatic generated and their impact in achieving the goal of the developed visualization. The main difference between both

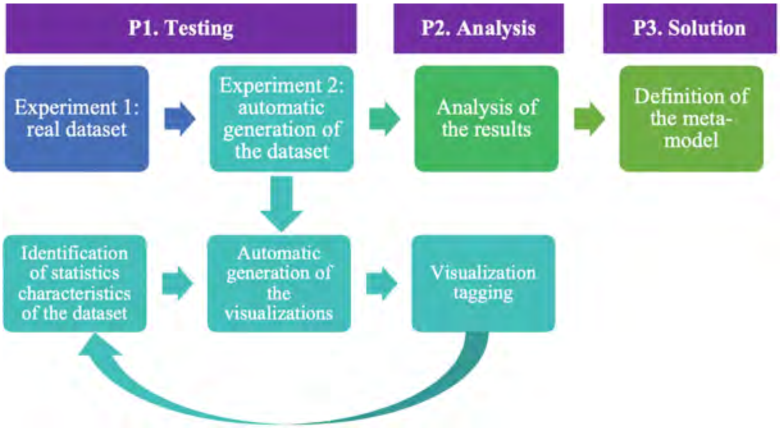


Fig. 1. Method used to identify the features that influence in the automatic generation of not misleading visualizations

experiments is the abstraction level. The first experiment is based on a real dataset from a particular domain. On the other hand, the second experiment has a high abstraction level because the dataset is random generated based on a set of statistic characteristics generated automatically.

Both experiments are focused on the automatic visualization generation. The generator use code templates based on a meta-model to define dashboards [2–4] and a Python script in which the different parameters are tuned to get a set of visualizations. The script processes the dataset changing a set of characteristics and provide a HTML and JavaScript file with the visualizations. The first experiment has the following characteristics:

- Dataset: Tri-variate dataset regarding the COVID-19 incidence rate in the 21 districts of Madrid (Spain)[1] and average income per household for each district[2].
- Goal: Identifying correlation among certain variables.
- Encoding channels: Two (X and Y position) and three (X and Y position and size).
- Type of visualization: Scatter plot with "circles" as visual mark.
- Scales' domain range: We changed the minimum and maximum of the scale domain. We changed these values using these measures: the scale variable mean minus/plus two-times the standard deviation of that column in the dataset, the column's minimum/maximum value, the column's minimum value multiplied by 0.5 and zero and the column's maximum value multiplied by 1.5. In the case of nominal variables, the domain holds all the existing nominal values within that column of the dataset.

[1] Official data source from the Madrid's government open data portal: https://datos.comunidad.madrid/catalogo/dataset/covid19_tia_muni_y_distritos/resource/f22c3f43-c5d0-41a4-96dc-719214d56968.

[2] Official data source from the Spanish National Institute of Statistics: https://www.ine.es/jaxiT3/Tabla.htm?t=31097.

Regarding the second experiment, the main characteristics are:

- Dataset: A bivariate dataset randomly generated using a set of statistical characteristics that are changed to generate the set of visualizations. The dataset is generated to cover different types of probability distribution with different statistical dispersion. Specifically, the standard deviation and median are the modified values.
- Goal: Comparison between certain variables.
- Encoding channel: Color scale and section of the map.
- Type of visualization: Map.
- Scales' domain range: Same variations as in the first experiment.

The last part of the experiments was the labeling process. All authors worked together to tag each generated visualization as misleading or no misleading. In the first experiment, this process enabled the identification of the features that introduce misleading in a visualization for a particular goal and domain. On the other hand, we applied the same process in the second experiment but following an iterative approach (Fig. 1). The labeling process enabled the redefinition of the statistic characteristics of the dataset in order to generate the visualization. This process enabled the identification of the features that have an objective impact in the automatic visualization generation; the features of the domain itself.

Finally, we analysed the results of the experimental phase and define a solution to consider the features of the domain as an input to automatic development of information visualizations.

2.2 Meta-modeling

The model-driven development (MDD) paradigm [5, 6] enables the abstraction of the characteristics and functionalities involved in the development of information systems. The main strength of this paradigm is that it moves data and operations specifications away from technologically specific details.

Following this approach, a dashboard meta-model was developed in previous studies, obtaining a set of abstract elements and relationships to define specific products [2–4]. A fragment of the dashboard meta-model is shown in Fig. 2.

2.3 Automatic Generation

We have developed an automatic dashboard generator based on the meta-model. The code generator takes as an input a set of parameters that account for the elements and attributes of the meta-model, and the result is the source code of a dashboard according to the provided configuration.

The approach taken to automatically generate the source code is based on the software product line (SPL) paradigm [7, 8] and we developed different HTML and JavaScript code templates [9] to materialize the variability points of the product line [10].

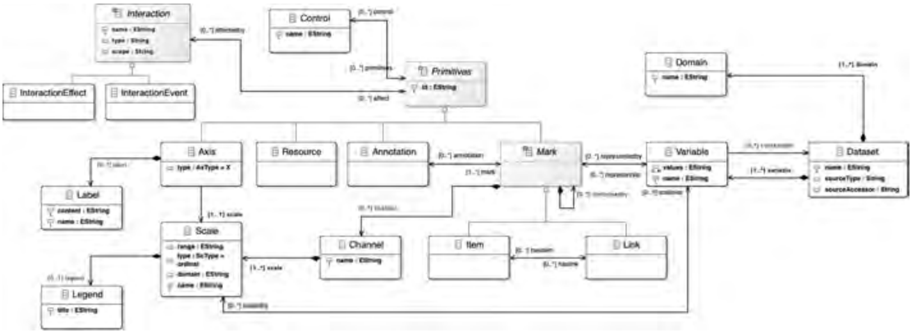


Fig. 2. Fragment of the previous version of the dashboard meta-model.

3 Meta-model Modification

3.1 Domain Characterization

It is necessary to identify relevant features to characterize the domain and to materialize those characteristics in useful visualizations. In this case, we defined the domain as a set of attributes that statistically describe the variables involved in that domain.

Specifically, we have included in the meta-model a new class named *DomainVariable*, which represent data variables that are part of the data domain. This class is associated with a domain (in the meta-model, the class *Domain*) and also with the class *Variable*, which represent a variable that belongs to a dataset to be displayed through the visualization. The *Variable* entity is seen as a sample of the *DomainVariable* entity.

The domain variable enables us to perform analytic tasks on the visualization and reach insights, because we have information about which values are normal, which values are outliers, or which tendency is being developed.

If users see a visualization with information from a domain which they don't fully understand, their conclusions might be wrong. But also, if practitioners don't fully understand the data domain of a visualization they are developing, they could end up with a misleading graphic. Another example about this concern is given. When visualizing information in X, Y coordinates it is necessary to select the domain of the scales in both axes; different scale extents might distort the whole data story being told. Figure 3 shows an example of the same data visualized through different Y-axis scales.

If we are not aware of the data domain, the first graph can be seen as misleading for not starting the Y-axis at the zero value. Starting the Y-axis at the zero value (as in the second graph), gives us the impression that the temperature change along time is very small and, indeed it is (in absolute terms) [11].

However, in this case, being aware that the tendency and average temperatures of the world over the last decades provides the context to understand that a change of 1 ºC in average temperature is a huge increment in this domain, conveying a whole different story. So, although the first graph does not comply to Tufte's lie factor [12], it is more honest than the second in terms of representing data framed in this domain.

For these reasons, we included as abstract attributes of the *DomainVariable* entity the following characteristics: mean, standard deviation, median, first quartile, third quartile,

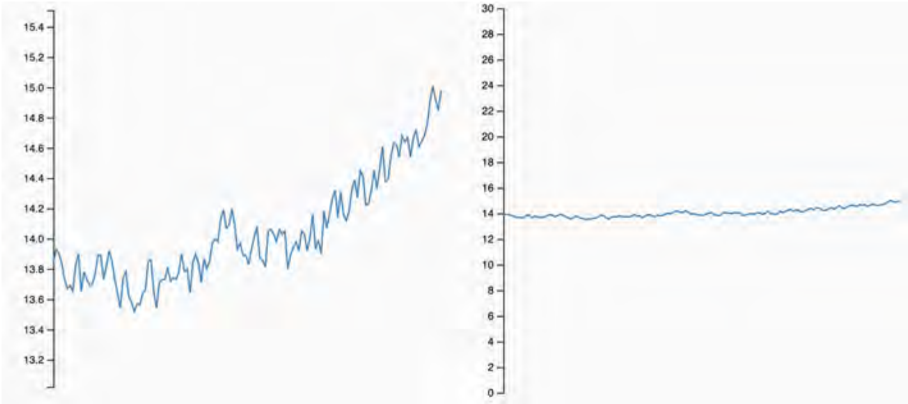


Fig. 3. Example of the effect of different scales when visualizing data.

interquartile range, maximum and minimum. These values not only describe statistically the variable, but also they help in characterizing their distribution [13], as they give notion of its dispersion, skewness and what values can be considered as outliers (Fig. 4).

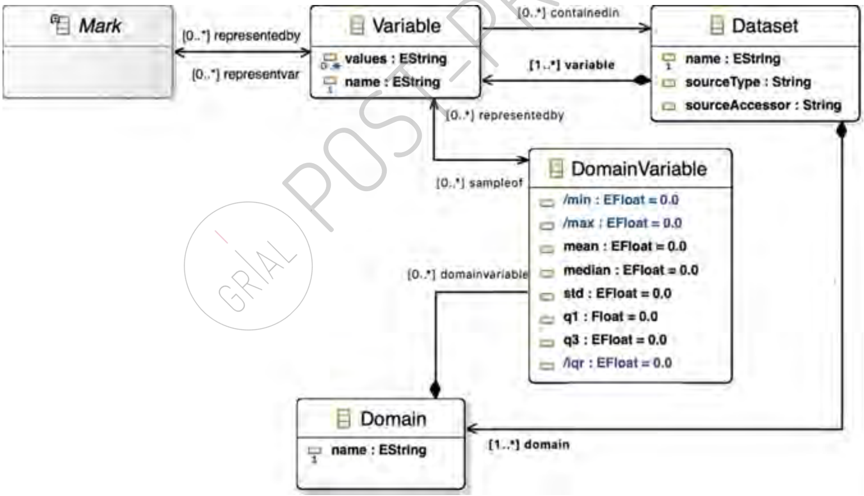


Fig. 4. Detail of the included class to characterize the domain in information visualizations.

3.2 Context Inclusion

We also included another association regarding the *DomainVariable* entity to consider the possibility of representing context in a visualization. In this case, we identify context as additional information related with a variable. For example, income household could be related with the COVID-19 incidence rate [14], and including that information in a visualization about COVID-19 incidence rate provides context to the data to be displayed.

By including a reflexive association on the *DomainVariable* class, we enable the possibility to identify and materialize relationships among variables from a different or the same domain (for example, because they are correlated).

The inclusion of this relationship to provide the notion of context allows the accountability of potentially relevant variables to include in a visualization before selecting its technical features (Fig. 5).

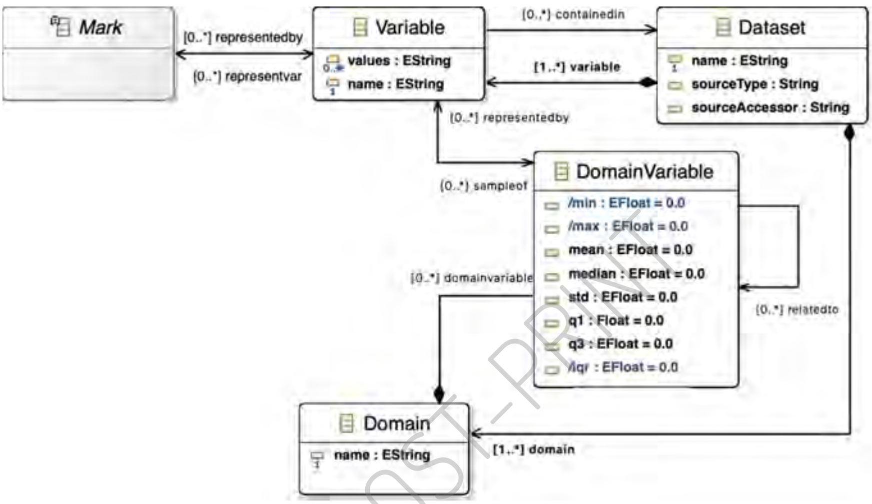


Fig. 5. Detail of the included reflexive association to represent data context in information visualizations.

4 Meta-model Instantiation Example

This section provides an instantiation example of the meta-model to illustrate the role of the included elements. Figure 6 shows the instantiation of a scatter plot for an hypothetical data domain and Fig. 7 the shows a generated visualization according to the instance.

The *DomainVariable* instance provides knowledge to select the Y-axis scale range in a way that data is not exaggerated. In this case, although the Y-axis does not start from zero (which can be seen as misleading in some domains), the domain knowledge provides us the justification: according to the domain characteristics for that variable, it is not likely to find values below 30, so it would make no sense to start the axis at zero. The visualized variable (VariableA in Fig. 6) is seen as a sample of the domain variable.

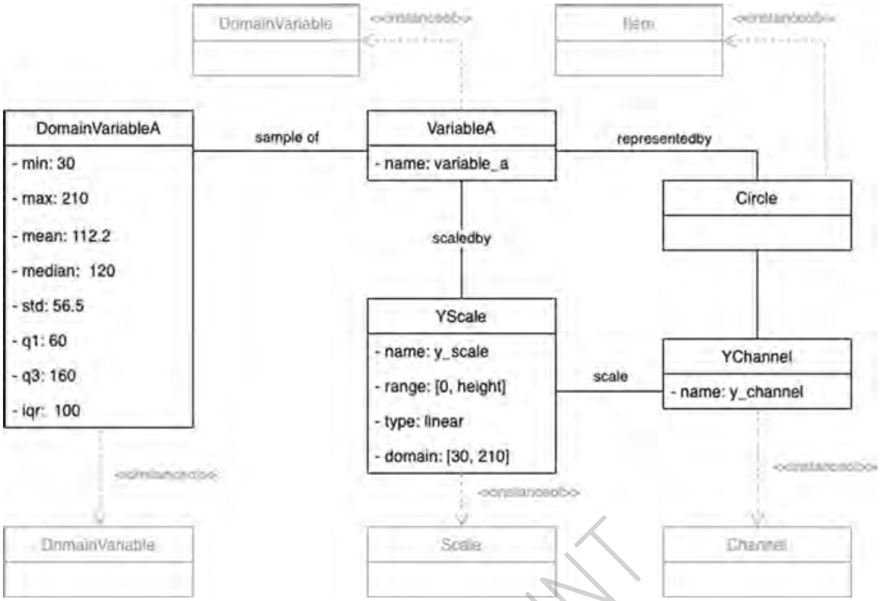


Fig. 6. Excerpt of the example visualization instantiation (Y-axis channel and scale).

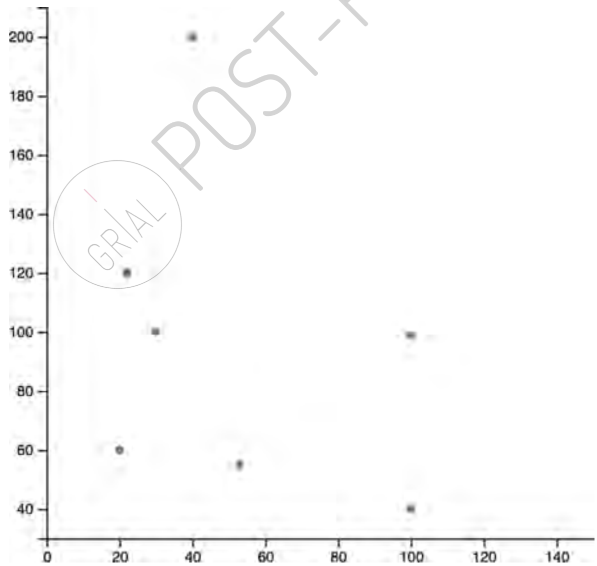


Fig. 7. Generated visualization through the meta-model's instantiation parameters.

5 Discussion

The results of this work set the foundations for characterizing conceptual concepts such as domain expertise and data context when designing information visualizations. Specifically, the testing process has yielded a new version of the meta-model that includes important factors such as the data domain characteristics. The use of a meta-model not only provides a theoretical framework to work with, but also a skeleton to instantiate real products adapted to a specific context.

We selected statistical features to define the domain variables involved in a visualization. More specifically, we included as attributes the characteristics that define box plots [13]. Although the values distribution's is better characterized by its probability density function or cumulative density function, these two functions are more difficult to incorporate to the meta-model than a set of values such as the mean or the maximum and minimum values. However, there is a limitation that will be explored in subsequent works: the inclusion of the notion of uncertainty in data (because the *DomainVariable* represents knowledge about the population), which is also a significant factor when visualizing data [15–17].

It is also important to take into account that there could exist different domain variables with different values' distributions but the same summary statistics [18]. In this case, we don't intend to use the *DomainVariable* values to compare domain variables among them, but to define visual features (such as scales) regarding that variable solely.

Moreover, the goal of the visualization must be accounted for too. In the generation example, the generated visualization would not be useful for detecting outliers, because they would fall outside the scale's domain. It is important to find balance with the visualization goal when including the domain knowledge.

Finally, this approach is limited to domain expertise: if the *DomainVariable* values are populated with wrong values, the whole process would be affected in the same way that a user can reach wrong conclusions if his or her notion of the domain differs from reality.

6 Conclusions

This work presents a meta-modeling approach to incorporate the notion of domain knowledge and data context into information visualizations. The approach is supported by a code generator that materialize the meta-model's abstract features into specific visualizations. The meta-model proposal enables not only a set of rules and guidelines to define no misleading visualizations, but also supports the automatic generation of information visualizations.

The study based on the automatic generation of datasets of visualizations has supported the identification process of the features, especially those related to the data domain, that influence in the development of no misleading visualizations. We introduced the results of the experiments as part of the meta-model to define dashboards.

Future research lines will involve in-depth testing of the influence of domain expertise and data context on the visual elements of a dashboard with the goal of including this knowledge into the generation pipeline.

Acknowledgements. This research work has been supported by the Spanish *Ministry of Education and Vocational Training* under an FPU fellowship (FPU17/03276). This research has been partially funded by the Spanish Government Ministry of Economy and Competitiveness throughout the DEFINES project (Ref. TIN2016-80172-R).

References

1. Pandey, A.V., Manivannan, A., Nov, O., Satterthwaite, M., Bertini, E.: The persuasive power of data visualization. IEEE Trans. Visual Comput. Graph. **20**(12), 2211–2220 (2014)
2. Vázquez-Ingelmo, A., García-Peñalvo, F.J., Therón, R., Conde, M.Á.: Representing data visualization goals and tasks through meta-modeling to tailor information dashboards. Appl. Sci. **10**(7), 2306 (2020)
3. Vázquez-Ingelmo, A., García-Holgado, A., García-Peñalvo, F.J., Therón, R.: A meta-model integration for supporting knowledge discovery in specific domains: a case study in healthcare. Sensors **20**(15), 4072 (2020)
4. Vázquez-Ingelmo, A., García-Peñalvo, F.J., Therón, R.: Capturing high-level requirements of information dashboards' components through meta-modeling. In: The 7th International Conference on Technological Ecosystems for Enhancing Multiculturality (TEEM 2019), León, Spain (2019)
5. Pleuss, A., Wollny, S., Botterweck, G.: Model-driven development and evolution of customized user interfaces. In: Proceedings of the 5th ACM SIGCHI Symposium on Engineering Interactive Computing Systems, pp. 13–22. ACM (2013)
6. Kleppe, A.G., Warmer, J., Bast, W.: MDA Explained. The Model Driven Architecture: Practice and Promise. Addison-Wesley Longman Publishing Co., Inc., Boston, MA (2003)
7. Clements, P., Northrop, L.: Software product lines. Addison-Wesley, Boston, MA, USA (2002)
8. Pohl, K., Böckle, G., Van der Linden, F.J.: Software product line engineering: foundations, principles and techniques. Springer-Verlag, New York Inc, New York, NY, USA (2005)
9. Vázquez-Ingelmo, A., García-Peñalvo, F.J., Therón, R.: Addressing fine-grained variability in user-centered software product lines: a case study on dashboards. In: World Conference on Information Systems and Technologies, pp. 855–864. Springer (2019). https://doi.org/10.1007/978-3-030-16181-1_80
10. Vázquez-Ingelmo, A., García-Peñalvo, F.J., Therón, R.: Taking advantage of the software product line paradigm to generate customized user interfaces for decision-making processes: a case study on university employability. PeerJ Comput. Sci. **5**, e203 (2019). https://doi.org/10.7717/peerj-cs.203
11. Boer, H.D.: The lie-factor/baseline paradox. Medium. https://medium.com/@hiljedeboer/the-lie-factor-baseline-paradox-ec5955393d19. Accessed (2020)
12. Tufte, E., Graves-Morris, P.: The visual display of quantitative information 1983, ed. Cheshire. Graphics Press, CT, USA (2014)
13. Williamson, D.F., Parker, R.A., Kendrick, J.S.: The box plot: a simple visual method to interpret data. Ann. Intern. Med. **110**(11), 916–921 (1989)
14. Finch, W.H., Finch, M.E.H.: Poverty and Covid-19: rates of incidence and deaths in the United States during the first 10 weeks of the pandemic. Front. Sociol. **5**, 47 (2020)
15. Kale, A., Nguyen, F., Kay, M., Hullman, J.: Hypothetical outcome plots help untrained observers judge trends in ambiguous data. IEEE Trans. Visual Comput. Graph. **25**(1), 892–902 (2018)
16. Hullman, J., Kay, M.: Uncertainty+Visualization, Explained https://medium.com/multiple-views-visualization-research-explained/uncertainty-visualization-explained-67e7a73f031b. Accessed 2019

17. Kay, M., Kola, T., Hullman, J.R., Munson, S.A.: When (ish) is my bus? user-centered visualizations of uncertainty in everyday, mobile predictive systems. In: Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems, pp. 5092–5103 (2016)
18. Matejka, J., Fitzmaurice, G.: Same stats, different graphs: generating datasets with varied appearance and identical statistics through simulated annealing. In: Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems, pp. 1290–1294 (2017)

7.27 Appendix AA. A platform for management and visualization of medical data and medical imaging

A platform for management and visualization of medical data and medical imaging

Andrea Vázquez-Ingelmo
GRIAL Research Group, Department of Computer Science, Research Institute for Educational Sciences. University of Salamanca, Salamanca, Spain
andreavazquez@usal.es

Jesús Sampedro-Gómez
Cardiology Department, Hospital Universitario de Salamanca, SACyL. IBSAL, Facultad de Medicina, Universidad de Salamanca, and CIBERCV (ISCiii), Salamanca, Spain,
jmsampedro@saludcastillayleon.es

Antonio Sánchez-Puente
Cardiology Department, Hospital Universitario de Salamanca, SACyL, IBSAL, Facultad de Medicina, Universidad de Salamanca, and CIBERCV (ISCiii), Salamanca, Spain
asanchezpu@saludcastillayleon.es

Víctor Vicente-Palacios
Philips Healthcare, Spain
victor.vicente.palacios@philips.com

P. Ignacio Dorado-Díaz
Cardiology Department, Hospital Universitario de Salamanca, SACyL, IBSAL, Facultad de Medicina, Universidad de Salamanca, and CIBERCV (ISCiii) Salamanca, Spain
acho@usal.es

Pedro L. Sanchez
Cardiology Department, Hospital Universitario de Salamanca, SACyL, IBSAL, Facultad de Medicina, Universidad de Salamanca, and CIBERCV (ISCiii), Salamanca, Spain
plsanchez@usal.es

Francisco José García-Peñalvo
GRIAL Research Group, Department of Computer Science, Research Institute for Educational Sciences. University of Salamanca, Salamanca, Spain
fgarcia@usal.es

ABSTRACT

The application of artificial intelligence algorithms to medical data has gained relevance over the years. These algorithms can enable disease detection, image segmentation, assessment of organ functions, among other research tasks. However, to effectively apply and benefit from artificial intelligence in this context, it is important to tackle the heterogeneity and diversity of data structures and data sources. For these reasons, it is important to rely on information systems that unify data found in medical domains. This work outlines the features of an online platform that allow different roles to upload, process and research on structured medical data and medical imaging.

CCS CONCEPTS

• **Information systems** → Data management systems; Information retrieval; • **Human-centered computing** → Human computer interaction (HCI); • **Computing methodologies** → Artificial intelligence.

TEEM'20, October 21–23, 2020, Salamanca, Spain

ACM ISBN 978-1-4503-8850-4/20/10…$15.00
https://doi.org/10.1145/3434780.3436652

KEYWORDS

Data management, Structured medical data, Medical imaging, Health platform, Artificial Intelligence

ACM Reference Format:
Andrea Vázquez-Ingelmo, Jesús Sampedro-Gómez, Antonio Sánchez-Puente, Víctor Vicente-Palacios, P. Ignacio Dorado-Díaz, Pedro L. Sanchez, and Francisco José García-Peñalvo. 2020. A platform for management and visualization of medical data and medical imaging. In *Eighth International Conference on Technological Ecosystems for Enhancing Multiculturality (TEEM'20), October 21–23, 2020, Salamanca, Spain.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3434780.3436652

1 INTRODUCTION

Artificial intelligence algorithms are becoming a standard in the study of medical data with research purposes [1]. Among them, deep learning models applied to medical imaging have enabled the automation of complex tasks such as disease detection, segmentation of structures, or the assessment of organ functions [2] with similar performance compared to human skill [3].

Nevertheless, these advances are generally limited to specific tasks and datasets in which algorithms are trained and validated. Therefore, one of the main challenges in seeing these technologies applied in real medical scenarios consists of achieving reliable performance for different sources of image data (cohorts, machine brands, operators, etc.). In this sense, informatic tools are still required to collect, organize, manipulate and apply artificial intelligence algorithms for medical data in a friendly, secure and anonymized way. In this study, we present the design parameters, architecture and functions of an online platform for the management and visualization of structured medical data and medical imaging.

The paper is organized as follows. Sec. 2 outlines the starting point and background of the project. Sec. 3 describes the platform's architecture, while Sec. 4 presents the platform's features. Finally, Sec. 5 outlines the conclusions.

2 BACKGROUND

The starting point of this project stems from the need to use a web-based collaborative platform to gather structured and medical imaging data for research purposes in the field of cardiology, with the possibility of integrating artificial intelligence algorithms within the platform. Cardiac imaging is particularly interesting for the application of AI algorithms, since many tasks are related to the assessment of volumes, distances and motion of different structures in the heart and, in this regard, deep learning techniques have been proven to achieve good results [4].

Cardiac imaging is usually composed of DICOM (Digital Imaging and Communication On Medicine) files [5] from echocardiographic, magnetic resonance, and computed tomography. Other online medical imaging platforms implement the DICOM protocol and are available for these purposes, some of them even in an open-source format [6].

On the other hand, there exist solutions based on application programming interfaces with pretrained models repositories for use in the medical imaging field, but they are mostly oriented towards advanced users with expertise in programming and data science [7].

The solution is the development of a technological ecosystem [8, 9] that merges both functionalities into a user-friendly web-based interface, which would support all the tasks required to store structured and imaging data from clinical research studies and enable both healthcare professionals and data scientists to apply deep learning models to the stored images.

In this sense, the platform should be capable of collecting data grouped by multicentric research projects. In addition, structured data must support different levels of information associated with patients, image studies, or files; sometimes presenting a longitudinal structure (repeated measurements or studies for the same patient over time).

Users should be associated with their center and have different permissions depending on their role, defined as: administrators, principal researchers, IA developers and data collectors.

Regarding imaging data, the platform must handle the DICOM format, read some of the metadata tags, check the integrity of data anonymization of each file and allow the visualization and manipulation of the images.

Deep learning models can also be stored by users with the AI developer role in the form of Python scripts with different output formats such as a diagnostic or a segmentation map. Hence, these scripts can be executed through the web interface by any other user interested in analyzing the image.

3 ARCHITECTURE

The platform relies on two pillars: structured data and image data management. To achieve an effective management, different technologies and frameworks are integrated through a client-server approach.

Figure 1 presents a schematic overview of the platform's involved technologies. The front end is based on HTML, CSS, and JavaScript. The DICOM viewer and editor, also located at the front end, relies on the Cornerstone.js library, as it will be explained in Sec. 4.2.

The back end holds technologies that support the main functionalities of the platform: data storage, data processing and an Artificial Intelligence environment.

The web application is built on Django, a Python-based web framework [10]. The web application is the entry point of the platform, providing users with different services. This web application is connected to other technologies (databases, file systems and external tools such as REDCap) to manage the different data structures that will be part of the projects.

The back end also relies on libraries such as OpenCV or Tensorflow to support the execution and application of Artificial Intelligence scripts.

4 PLATFORM FEATURES

There are three main functionality blocks on the platform: data management, image edition and Artificial Intelligence applications. These blocks provide an ecosystem of collaborative features that allow the achievement of complex goals.

4.1 Data management

As introduced in the previous section, the backbone of the platform is data management. Significant quantities of data need to be uploaded into the platform in order to analyze, view, edit or apply algorithms to them. Moreover, data schemas could vary among different projects, adding another complexity layer to the definition of the platform's data structures.

For these reasons, the platform must be very flexible in terms of data organization to allow the definition and modification of data schemas across projects.

Two methods are integrated and employed to tackle data management. First, the Django ORM (a database-abstraction API) provides access to a relational database with the structure shown in Figure 2. The platform is mainly organized through projects, which will hold data from different patients. Patient's structured data is stored as JSON, because of the varying nature of the data schemas among projects.

Three entities are provided to manage image data: studies, series and files. These entities contain data about the DICOM images that will be uploaded into the platform. A study can contain different series, which would reference different files.

The actual image files are referenced through these database entities, and are stored through filesystems (the type of filesystem can be interchanged to allow the selection of different storage technologies).

On the other hand, the platform integrates an external tool: a REDCap instance. REDCap (Research Electronic Data Capture) is an

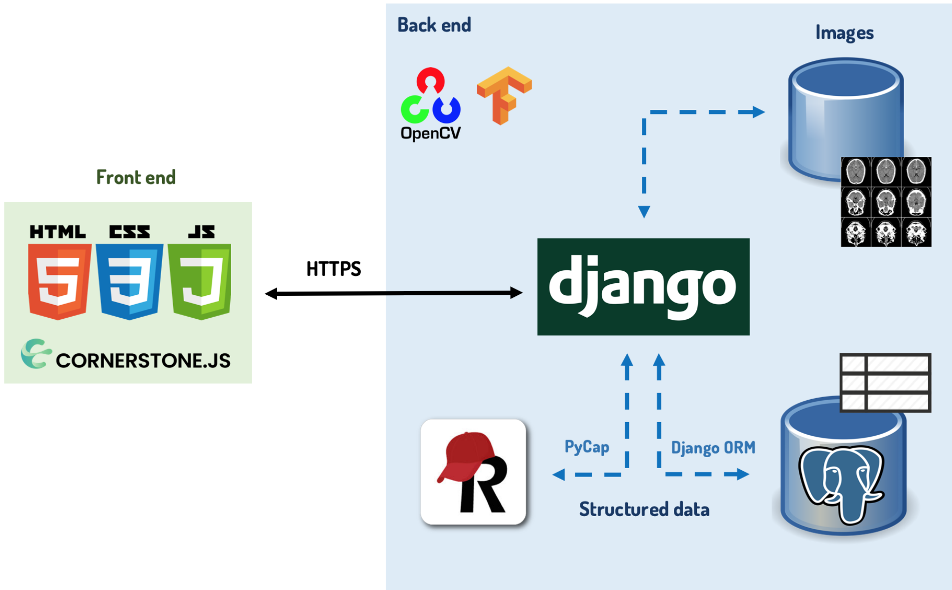


Figure 1: Schematic structure of the platform's architecture

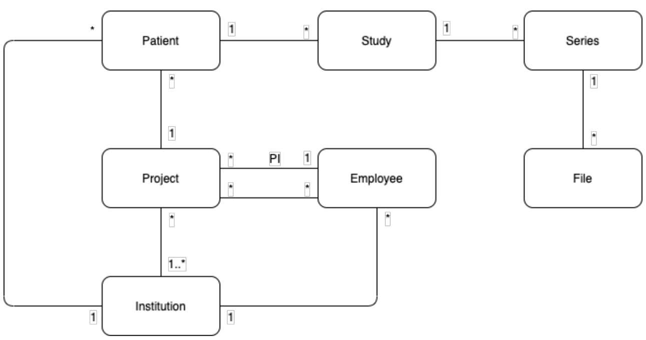


Figure 2: Overview of the platform's data organization

electronic data capture (EDC) software and workflow methodology for creating and designing clinical research databases [11].

Given its integration with other programming languages, such as Python, REDCap data can be imported through API calls (by using the PyCap package, https://pypi.python.org/pypi/PyCap), unifying all data in the platform's storage.

This integration not only provides a method to retrieve data from the REDCap instance, but also the collaboration of both technologies to obtain a more complete tool.

On the other hand, data upload processes are time-consuming tasks, especially image uploads, because DICOM studies can involve several heavy files. For these reasons, the data upload process must be robust and usable.

Job queues are employed to tackle data uploads asynchronously as background tasks, allowing users to navigate the platform and perform other duties while their data is uploading.

```
{
  "BrushTool":{
    "pixelData":[ ],
    "segmentsOnLabelmap":[
      0,
      7,
      1
    ]
  },
  "ArrowAnnotate":{
    "data":[
      {
        "text":"this is an example",
        "uuid":"e2245445-368d-425c-ab28-f429f5de07a7",
        "active":false,
        "handles":{ },
        "visible":true,
        "invalidated":true
      }
    ]
  }
}
```

Figure 3: Example of the storage of image annotations and segmentations

4.2 Image visualization and edition

Another of the main features of the platform is the possibility to process the uploaded images in place, without the necessity of any other external tool. As explained in section 3, image edition takes place in the browser through the Cornerstone.js (https://github.com/cornerstonejs/cornerstone) framework, an open-source library to parse and render DICOM files.

Users can locally edit the images they are currently viewing. To make the annotations and modifications persistent, the viewer

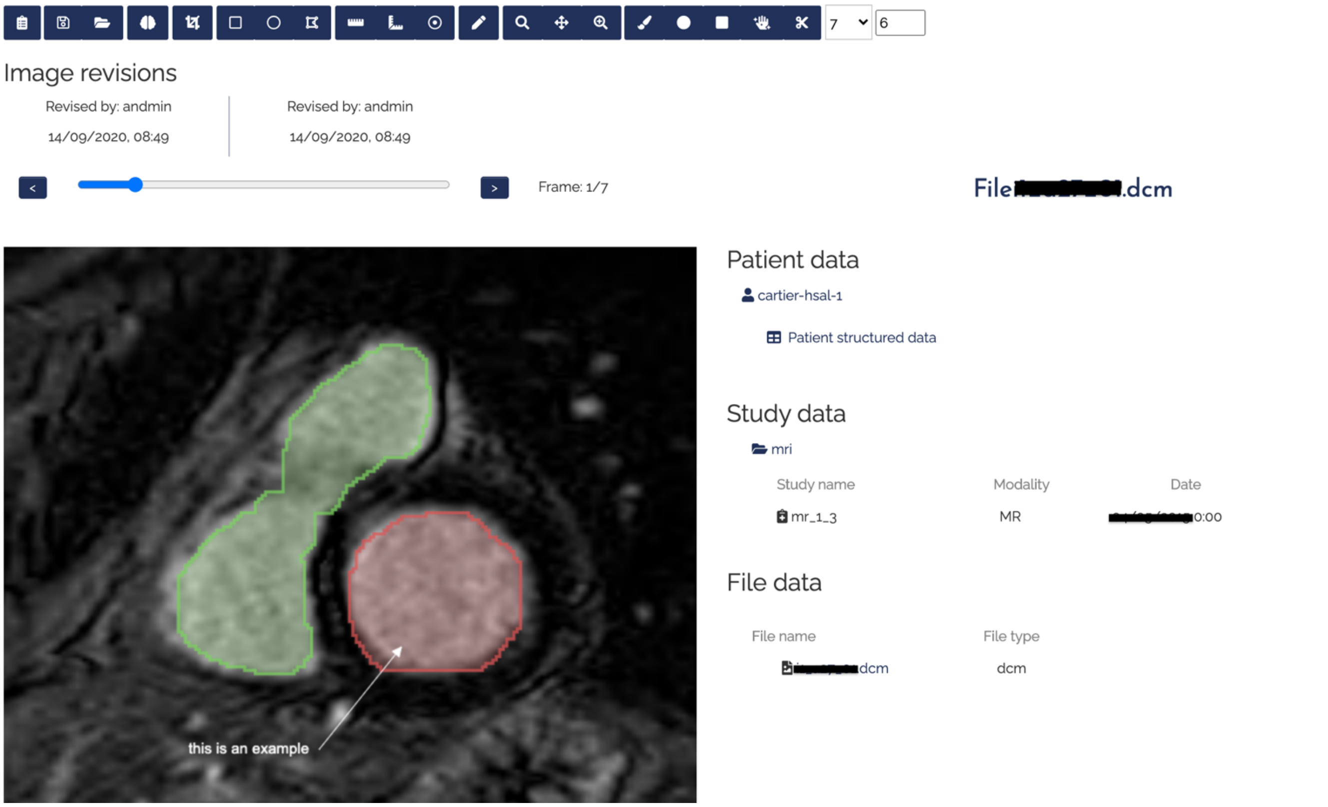


Figure 4: Example of an image segmentation on the platform's image viewer and editor. Image revisions can be saved and navigated to see other researchers' annotations

provides a button that sends the modifications to the server, storing all changes in the database.

The image itself is never modified; annotations or segmentations are stored as raw data (Figure 3). This approach allows the storage of annotations from different users or from different dates, thus enabling comparisons or even annotations' version control. Also, storing image revisions as JSON objects avoids the necessity of duplicating DICOM images to store the edited versions, saving space and making the retrieval of file annotations and segmentation masks more efficient.

The viewer's image edition (Figure 4) tools (except the crop tool and artificial intelligence tool) are supported by another open-source framework, which provides an extensible solution for creating tools on top of Cornerstone.js (https://github.com/cornerstonejs/cornerstoneTools). The following tools are available through the image editor:

- Brush and scissors tools for image segmentation.
- Segmentation layers and size selectors to ease the segmentation process of the images.
- Length and area tools for measure image fragments.
- Annotation tools.
- Zoom tools.
- A crop tool.
- A tool to apply uploaded and validated AI scripts.

4.3 Artificial Intelligence application

The last main feature of the platform is the integration of Artificial Intelligence (AI) algorithms.

This feature allows researchers to upload their AI scripts into the platform and make them available to other users. The platform provides an uploader to define the algorithm's meta-data. It is important to maintain a generic structure on the algorithms' structure to enable a proper integration with the platform. For example, the scripts' inputs and outputs must be standardized for the platform to correctly execute the scripts.

Algorithms' meta-data is important, because they could yield different outputs (another image, a diagnosis, a measure, a segmentation mask, etc.), or their application might be limited to specific DICOM modalities.

Once uploaded, algorithms are available at the viewer component. The application process is straightforward: a button provides information about the available scripts for the current image and the user only needs to select an algorithm and confirm their application.

Depending on the algorithm's output type, a new image or text block will be shown with the result, making the AI application process transparent to the users.

5 CONCLUSIONS

This paper presents a collaborative platform that allows the management of medical data, including medical images data. The platform supports not only plain upload of these data, but also integrate tools to analyze, visualize and process them. Among these tools there is the possibility of integrating artificial intelligence scripts to make the application process transparent to those users that do not have programming skills.

Future research lines will involve usability testing of the platform, as well as the integration of customized data visualizations [12, 13] to ease the analysis of patient's structured data based on the data schema and user goals.

ACKNOWLEDGMENTS

This research work has been supported by the Spanish *Ministry of Education and Vocational Training* under an FPU fellowship (FPU17/03276). This work was also supported by national (PI14/00695, PIE14/00066, PI17/00145, DTS19/00098, PI19/00658, PI19/00656 Institute of Health Carlos III, Spanish Ministry of Economy and Competitiveness and co-funded by ERDF/ESF, "Investing in your future") and community (GRS 2033/A/19, GRS 2030/A/19, GRS 2031/A/19, GRS 2032/A/19, SACYL, Junta Castilla y León) competitive grants.

REFERENCES

[1] A. Rajkomar, J. Dean, and I. Kohane. 2019. Machine Learning in Medicine. N Engl J Med 380, 14 (04), 1347-1358. DOI:10.1056/NEJMra1814259.

[2] G. Litjens, T. Kooi, B. E. Bejnordi, A. A. A. Setio, F. Ciompi, M. Ghafoorian, J. A. W.M Van Der Laak, B. Van Ginneken, and C. I. Sánchez. 2017. A survey on deep learning in medical image analysis. Med Image Anal 42(Dec), 60-88. DOI:10.1016/j.media.2017.07.005.

[3] Xiaoxuan Liu, Livia Faes, Aditya U Kale, Siegfried K Wagner, Dun Jack Fu, Alice Bruynseels, Thushika Mahendiran, Gabriella Moraes, Mohith Shamdas, and Christoph Kern. 2019. A comparison of deep learning performance against health-care professionals in detecting diseases from medical imaging: a systematic review and meta-analysis. The lancet digital health 1, 6, e271-e297.

[4] O. Bernard, A. Lalande, C. Zotti, F. Cervenansky, X. Yang, P. A. Heng, I. Cetin, K. Lekadir, O. Camara, M. A. Gonzalez Ballester, G. Sanroma, S. Napel, S. Petersen, G. Tziritas, E. Grinias, M. Khened, V. A. Kollerathu, G. Krishnamurthi, M. M. Rohe, X. Pennec, M. Sermesant, F. Isensee, P. Jager, K. H. Maier-Hein, P. M. Full, I. Wolf, S. Engelhardt, C. F. Baumgartner, L. M. Koch, J. M. Wolterink, I. Isgum, Y. Jang, Y. Hong, J. Patravali, S. Jain, O. Humbert, and P. M. Jodoin. 2018. Deep Learning Techniques for Automatic MRI Cardiac Multi-Structures Segmentation and Diagnosis: Is the Problem Solved? IEEE Trans Med Imaging 37, 11 (Nov), 2514-2525. DOI:10.1109/TMI.2018.2837502.

[5] Peter Mildenberger, Marco Eichelberg, and Eric Martin. 2002. Introduction to the DICOM standard. European radiology 12, 4, 920-927.

[6] D. S. Marcus, T. R. Olsen, M. Ramaratnam, and R. L. Buckner. 2007. The Extensible Neuroimaging Archive Toolkit: an informatics platform for managing, exploring, and sharing neuroimaging data. Neuroinformatics 5, 1, 11-34. DOI:10.1385/ni:5:1:11.

[7] E. Gibson, W. Li, C. Sudre, L. Fidon, D. I. Shakir, G. Wang, Z. Eaton-Rosen, R. Gray, T. Doel, Y. Hu, T. Whyntie, P. Nachev, M. Modat, D. C. Barratt, S. Ourselin, M. J. Cardoso, and T. Vercauteren. 2018. NiftyNet: a deep-learning platform for medical imaging. Comput Methods Programs Biomed 158(May), 113-122. DOI:10.1016/j.cmpb.2018.01.025.

[8] A. García-Holgado and F. J. García-Peñalvo. 2017. Preliminary validation of the metamodel for developing learning ecosystems. In Fifth International Conference on Technological Ecosystems for Enhancing Multiculturality (TEEM'17) (Cádiz, Spain, October 18-20, 2017) J.M. Dodero, M.S. Ibarra Sáiz and I. Ruiz Rube Eds. ACM, New York, NY, USA. DOI:10.1145/3144826.3145439.

[9] Alicia García-Holgado and Francisco José García-Peñalvo. 2019. Validation of the learning ecosystem metamodel using transformation rules. Future Generation Computer Systems 91, 300-310. DOI:10.1016/j.future.2018.09.011.

[10] Django Software Foundation. 2015. Django Web Framework. https://www.djangoproject.com/

[11] Paul A Harris, Robert Taylor, Robert Thielke, Jonathon Payne, Nathaniel Gonzalez, and Jose G Conde. 2009. Research electronic data capture (REDCap)—a metadata-driven methodology and workflow process for providing translational research informatics support. Journal of biomedical informatics 42, 2, 377-381.

[12] Andrea Vázquez-Ingelmo, Francisco José García-Peñalvo, and Roberto Therón. 2019. Taking advantage of the software product line paradigm to generate customized user interfaces for decision-making processes: a case study on university employability. PeerJ Computer Science 5(2019/07/01), e203. DOI:10.7717/peerj-cs.203.

[13] Andrea Vázquez-Ingelmo, Francisco José García-Peñalvo, Roberto Therón, Daniel Amo Filvà, and David Fonseca Escudero. 2020. Connecting domain-specific features to source code: towards the automatization of dashboard generation. Cluster Computing. The Journal of Networks, Software Tools and Applications, In Press. DOI:10.1007/s10586-019-03012-1.

7.28 Appendix AB. A platform to support the visual analysis of the SALMANTICOR study outcomes: conveying cardiological data to lay users

A platform to support the visual analysis of the SALMANTICOR study outcomes: conveying cardiological data to lay users

Andrea Vázquez-Ingelmo

GRIAL Research Group, Department of Computer Science, Research Institute for Educational Sciences. University of Salamanca, andreavazquez@usal.es

Francisco José García-Peñalvo

GRIAL Research Group, Department of Computer Science, Research Institute for Educational Sciences. University of Salamanca, fgarcia@usal.es

Antonio Sánchez-Puente

Cardiology Department, Hospital Universitario de Salamanca, SACyL. IBSAL, Facultad de Medicina, Universidad de Salamanca, and CIBERCV (ISCiii), asanchezpu@saludcastillayleon.es

Víctor Vicente-Palacios

Philips Healthcare, victor.vicente.palacios@philips.com

Ana Laffond

Cardiology Department, Hospital Universitario de Salamanca, anaelaffond@gmail.com

Alberto García-Galindo

IBSAL, alberto_26796@usal.es

Jesús Sampedro-Gómez

Cardiology Department, Hospital Universitario de Salamanca, SACyL. IBSAL, Facultad de Medicina, Universidad de Salamanca, and CIBERCV (ISCiii), jmsampedro@saludcastillayleon.es

P. Ignacio Dorado-Díaz

Cardiology Department, Hospital Universitario de Salamanca, SACyL. IBSAL, Facultad de Medicina, Universidad de Salamanca, and CIBERCV (ISCiii), acho@usal.es

Pedro L. Sánchez

Cardiology Department, Hospital Universitario de Salamanca, SACyL. IBSAL, Facultad de Medicina, Universidad de Salamanca, and CIBERCV (ISCiii), plsanchez@usal.es

Cardiovascular diseases are the largest risk factor for mortality in developed countries, and hospitals are facing an increasing workload in relation to cardiovascular diseases. In this regard, policymaking related to these diseases and based on epidemiological knowledge of the population can be a powerful means to prevent and treat them. The SALMANTICOR study was conceived in this context. This study had the purpose of collecting data concerning the prevalence and incidence of structural heart disease in the province of Salamanca (Spain). However, the amount of data and variables collected (more than 300), can make the understanding of the results cumbersome for non-experience users. This work overviews the design and architecture of a data visualization platform to support the exploration of the SALMANTICOR study results, with a special focus on conveying this information to lay users.

CCS CONCEPTS • Human-centered computing ~ Visualization ~ Visualization application domains ~ Information visualization • Human-centered computing ~ Human computer interaction (HCI) • Applied computing ~ Life and medical sciences ~ Health care information systems

Additional Keywords and Phrases: Data Visualization, Health domain, Cardiology, Human-Computer Interaction

ACM Reference Format:

First Author's Name, Initials, and Last Name, Second Author's Name, Initials, and Last Name, and Third Author's Name, Initials, and Last Name. 2018. The Title of the Paper: ACM Conference Proceedings Manuscript Submission Template: This is the subtitle of the paper, this document both explains and embodies the submission format for authors using Word. In Woodstock '18: ACM Symposium on Neural Gaze Detection, June 03–05, 2018, Woodstock, NY. ACM, New York, NY, USA, 10 pages. NOTE: This block will be automatically generated when manuscripts are processed after acceptance.

1 INTRODUCTION

Cardiovascular diseases are the largest risk factor for mortality in developed countries [1] and although the number of deaths related to cardiovascular disease has fallen in recent years, the number of people affected has not. This fact makes it more evident that hospitals are facing an increasing workload in relation to cardiovascular diseases. This is the reason why the planning of relevant public policies, based on epidemiological knowledge of the population, can be of great help in the prevention and treatment of this type of diseases [2]. This planning is even more important in ageing populations [3] where the prevalence of cardiovascular diseases is even higher, as is the case in the province of Salamanca (Spain), where the average age is over 48 years old. In this sense, the SALMANTICOR study was conceived: to obtain data concerning the prevalence and incidence of structural heart disease and therefore to be able to find relevant patterns with which to build appropriate public policies and/or public health campaigns.

However, the number of total variables collected (more than 300) can make the analysis of these data complex for non-experienced users. It is therefore important to provide practitioners, who are the ones with the medical knowledge, with a suitable platform to visualise these data in a simple way and thus confirm or disprove their hypotheses, as well as to find interesting patterns of analysis. In addition, these premises can help experienced users in further analysis.

This work outlines the design decisions and the architecture followed to build the first version of a visualization platform for supporting the exploration and exploitation of the collected variables from the SALMANTICOR study. This platform is not only focused on providing basic visualizations to summarize the results, but also on offering a good user experience to reach insights regarding the study.

The rest of this paper is organized as follows. Section 2 provides context to understand the usefulness of data visualizations. Section 3 describes the SALMANTICOR study. Section 4 outlines the proposed platform's architecture. Section 5 presents the results derived from the development of the platform. Finally, section 6 discusses the results and concludes the work.

2 BACKGROUND

Data visualization comprises different methods to represent raw or processed data through visual marks [4]. This discipline has always been relevant, given its numerous benefits when conveying information, but it is continuously growing in popularity due to the exponential use and generation of data in several tasks [5].

Visual representations enable better understanding of complex sets of data, supporting informed decision-making and knowledge generation. Data visualizations can be present in very different context, from scientific publications to newspapers and social media [6-8]. The reason of the extended use of data visualizations is that they can be designed fit a variety of purposes [9-12].

For example, data visualizations can be designed to convey scientific results (i.e., scientific publications), but also to engage readers or raise awareness regarding certain topics (i.e., newspapers). Moreover, they can be (and should be) adapted to the audience to which the visualization is focused.

All these characteristics make visualizations and dashboards powerful tools to analyse and convey data in the health domain [13, 14]. These tools are not intended to replace previous analysis methodologies, but to complement them through visual exploration, and, to make data more accessible and understandable to unskilled users. Moreover, data visualizations and dashboards can be employed to raise awareness regarding certain topics, which could include health awareness campaigns [14, 15].

For all these reasons, creating a visualization platform to disseminate the SALMANTICOR study results could lead to significant benefits, from conveying the results to a ground audience to enabling the possibility of performing deeper analyses on the SALMANTICOR data.

3 THE SALMANTICOR STUDY

The SALMANTICOR study [16] is a population-based cross-sectional descriptive study on the prevalence of structural heart disease and its risk factors. A total of 2,400 individuals, stratified by age, sex and place of residence (rural and urban), were recruited in the province of Salamanca (Spain). The study took place in the period between 2015 and 2018.

The province of Salamanca has a population density of 27.01 inhabitants/km² (2020) in a total area of 12,349.06 km², divided into 362 municipalities out of which more than half have populations of less than 300 inhabitants. However, 10 of these municipalities concentrate more than 67% of the total population of the province, with at least 5000 inhabitants. This contrast clearly differentiates the rural areas from the urban ones. In this context, out of a total of 35 primary health care centers in the province, 18 are considered urban health centers.

After obtaining written consent from study subjects, each participant underwent a baseline examination at their primary care centre. Afterwards, the participants filled in different questionnaires focused on socio-demographic data, cardiovascular risk factors, medical history, medication, dietary habits, and physical activity. In addition, empirically based data were collected to assess the health of the study subjects, among them: Echocardiographic and vascular function assessment, electrocardiographic examination, and a laboratory test.

The SALMANTICOR study was approved by the ethical committee of the healthcare community (Trial registration number: NCT03429452).

4 ARCHITECTURE PROPOSAL

The proposed platform is set to provide an overview of the study in a friendly way with the goal of engaging users to seek more details regarding the study's results and raise awareness regarding cardiovascular diseases, but also to provide an exploratory data analysis (EDA) tool for researchers looking for advanced data exploitation and analyses.

In this sense, the platform needs the flexibility to support the exploration of every variable involved in the study and provide summaries and a set of interactive visualizations for a broader audience.

For these reasons, the architecture is composed of modules that oversee the retrieval, visualization, and analysis of each study section. Dividing the architecture into individual but related modules enables flexibility to extend the platform with more functionalities, analyses, and visualizations.

The platform is developed as a web service. The front-end provides a usable interface and data visualizations to navigate the data, and the back-end offers data computation, storage, and retrieval functionalities through API calls.

Figure 1 shows the structure of the risk factors module, which is the generic structure for each study section. Each module communicates with the client to retrieve the data displayed through the D3.js framework [17, 18].

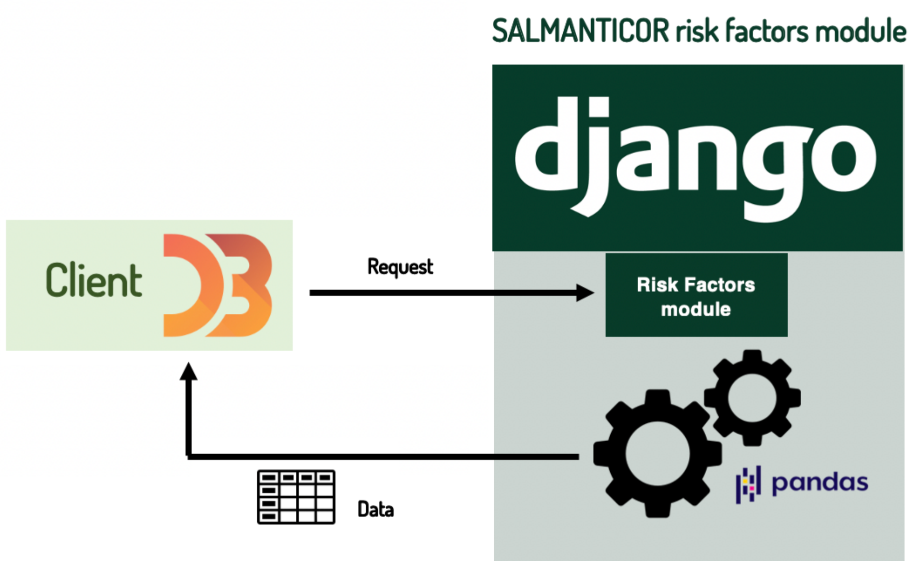


Figure 1: Outline of the platform's modules architecture.

Using this approach, it is possible to modify the interface and the visualizations without a negative impact on the business logic. Moreover, data visualizations can also be generated on demand through drag and drop interactions involving the study variables by using a dashboard generation service [19] (Figure 2). This feature is still under development, but it is aimed at advanced users that want to explore certain variables through specific visualizations.

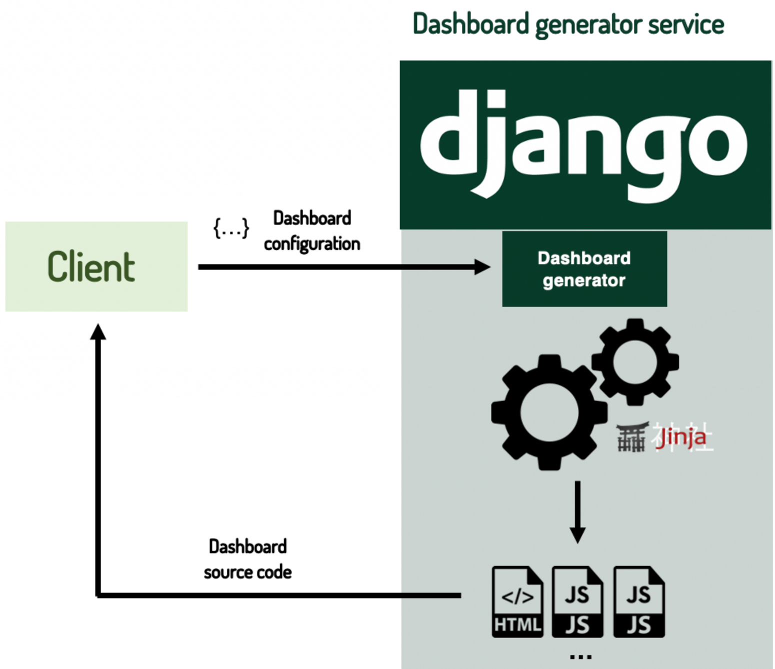


Figure 2: Outline of the integrated dashboard generator service that provides visualizations on demand [19].

5 THE PLATFORM

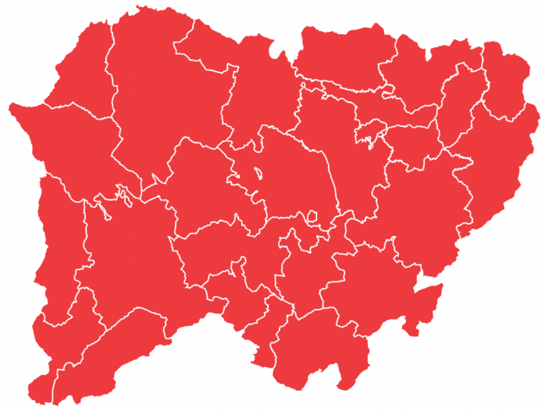
5.1 Study overview

As introduced before, one of the platform's goals is to disseminate the study and its results in an understandable manner to engage ground users. Since users could be overwhelmed by the significant amount of data, we divided the portal into different sections following the SALMANTICOR data organization.

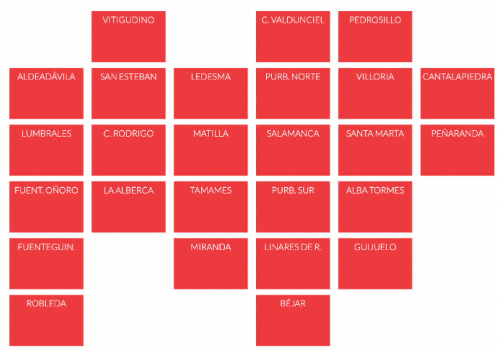
However, although dividing the study into smaller pieces makes the information more manageable, there could still be a lack of context that can confuse users. For these reasons, the main page contains a scroll visualization explaining the whole study and its purpose, including the population, sample, and geographical depiction of Salamanca's province.

Besides providing a graphical summary of the study, the scroll visualization is also aimed at engaging users through the scroll animations and interactivity (Figure 3). Maps are shown using a tile grid layout to allow combinations with other visual marks that summarize data from the study overview [20].

El estudio **SALMANTICOR** [1] es un estudio poblacional de incidencia y prevalencia de enfermedades cardíacas estructurales y factores de riesgo cardiovascular llevado a cabo en la provincia de Salamanca.



La provincia salmantina cuenta con un total de 36 zonas básicas de salud



Se trata de un estudio transversal donde se realizó un muestreo aleatorio estratificado por edad, sexo y el tipo de residencia de **2063 pacientes** a lo largo de todo el territorio.

Figure 3: First sections of the scroll visualization.

5.2 Visualizing and exploring the SALMANTICOR study results

Due to the vast number of variables involved in the study, it is necessary to provide end-users with a helpful tool to explore the different facets of data instead of static summary visualizations. However, displaying variables in a map without further details (such as sociodemographic factors or sampling details) could lead users to a superficial view of the results.

For these reasons, the exploration view provides controls to select specific variables to be inspected and a Sankey visualization that includes sociodemographic variables to overview the participants' characteristics (Figure 4). By interacting with the Sankey visualization (Figure 5), it is possible to filter the displayed results by different sociodemographic categories.

The map visualizations show two views of Salamanca's territory: rural and urban health districts. The size of each bubble represents the number of participants within that district, while their color represents the average of the selected variable per 100 inhabitants. This visual codification tries to avoid potential biases derived from the aggregation of data [21, 22] by highlighting the sample's size.

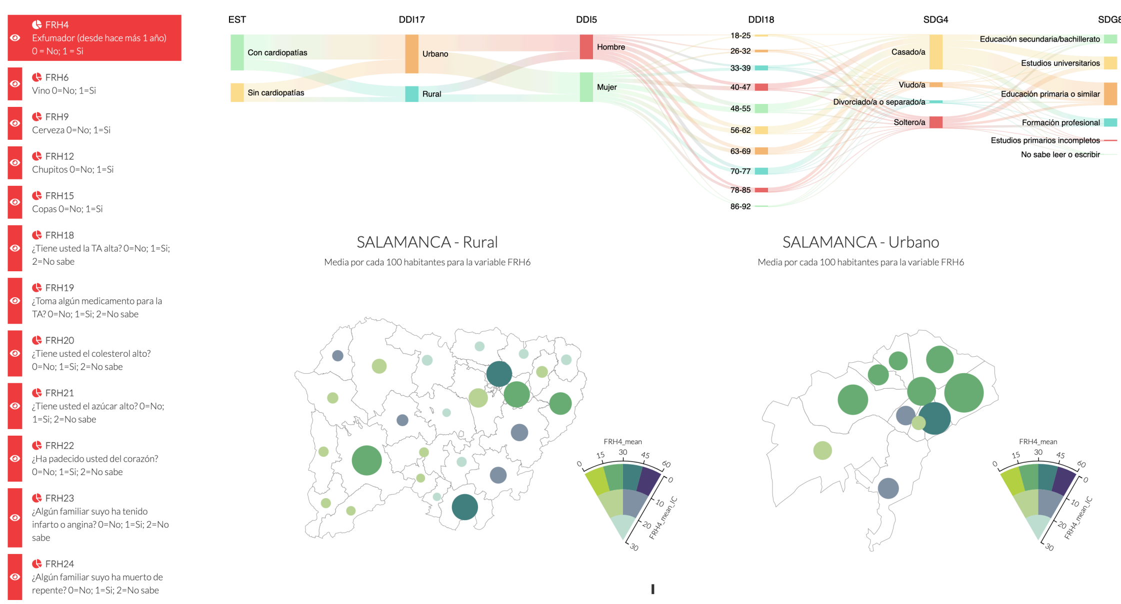


Figure 4: Risk factors exploration view.

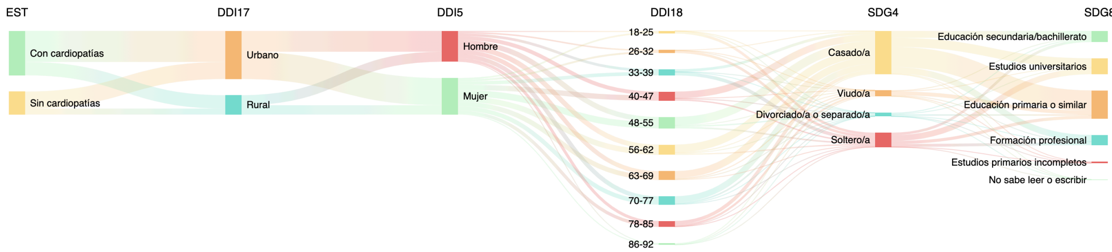


Figure 5: Sankey diagram with sociodemographic filters.

Conveying the sampling proportion error is crucial to avoid wrong conclusions. On the other hand, it is also essential to emphasize the uncertainty associated with the average calculations. We employed a value-suppressing uncertainty palette [23] to encode the average value and associated sampling proportion error. In this regard, the color intensity is higher for those values with a small error and vice versa, providing a straightforward way of identifying values with significant sampling proportion errors.

A map without uncertainty can also be unveiled through a slider in the bottom section of the page. This view is provided to allow comparisons among average values with or without uncertainty considered (Figure 6), which is useful to understand the impact of sampling errors.

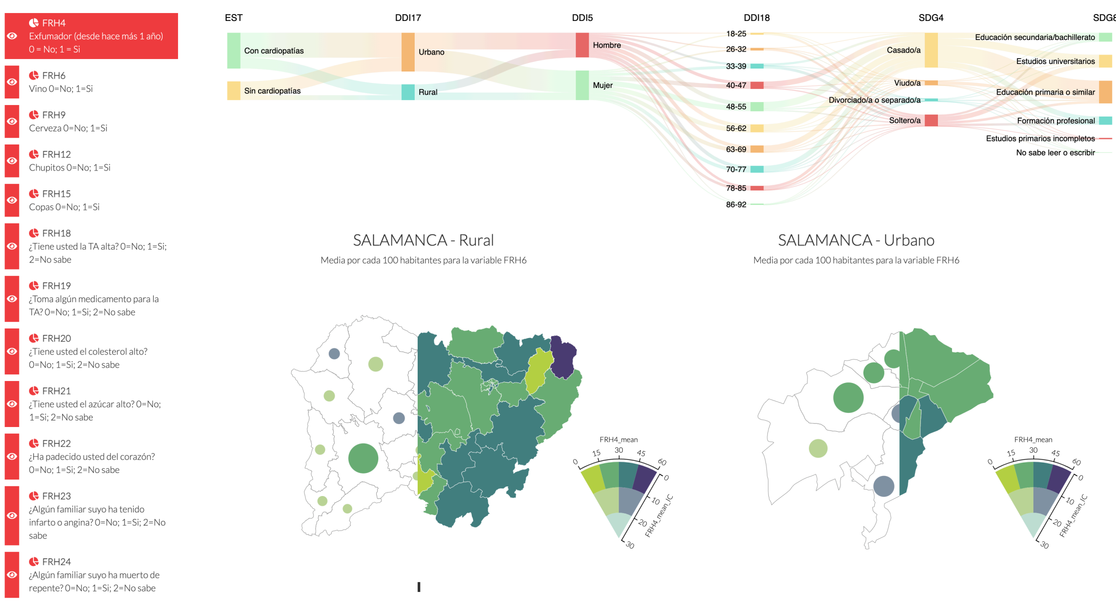


Figure 6: Map visualizations without uncertainty considered.

5.3 Advanced exploration

The last feature of the platform is an advanced workspace in which the summaries of every variable are displayed; numeric variables can be inspected through histograms, while categorical variables can be examined through bar charts with the count of every unique category involved (Figure 7).

Besides this individual summary of each variable, this feature allows users to drag and drop variables into a workspace and combine them to obtain different information visualizations. Visualizations are automatically generated through the generation service explained in section 4.

Although this section is still under development, it provides more insights regarding data and more flexibility to visualize results with other types of charts.

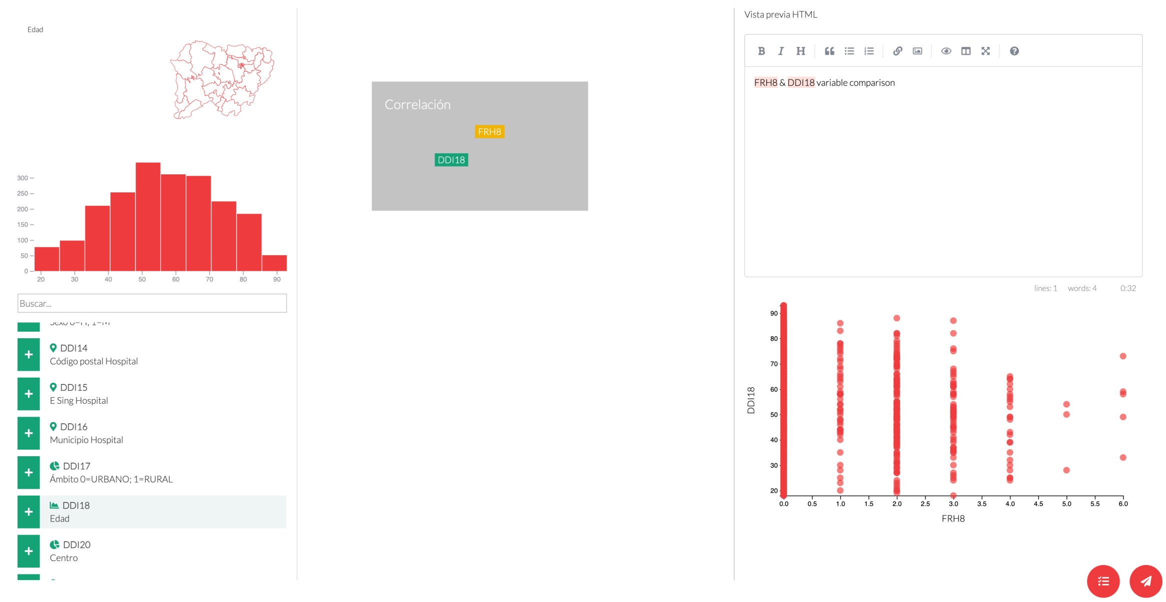


Figure 7: Advanced exploration of variables. The leftmost section provides a summary of each variable and the possibility of dragging them into the middle section to craft a visualization. The rightmost section displays the generated visualizations.

6 DISCUSSION AND CONCLUSIONS

Data visualizations are powerful tools to analyze complex datasets and convey information in understandable manners for broader audiences. Applying data visualizations to the health field has several benefits, including identifying patterns, clustering, comparisons, etc. [9] visually.

We developed a customized platform that relies on data visualizations to disseminate, analyze and convey data related to the SALMANTICOR study. The platform's backbone is a flexible architecture composed of individual but related modules that offer retrieval, storage, and computation features of the different study sections.

The platform provides the results of the study's sections through interactive visualizations and filters, which are helpful to explore the sociodemographic influence on the results and the different facets of data.

The main page offers an interactive and descriptive overview of the study. This page is crucial to understand the study purposes, population, samples, and methodology. Results can be consulted through individual pages in which enable users to freely select the variables to inspect and to apply sociodemographic filters to them, following the Shneiderman's mantra "overview first, zoom and filter, then details-on-demand" [24]. Finally, an advanced tool has been also proposed as a part of the platform, providing the possibility of crafting personalized visualization through interactions with the user interface.

Although the first version of the platform is promising, we aim to test the interface design in-depth to continue improving the visualizations and interactions to provide a good user experience and engagement with the SALMANTICOR data. Our future work will involve user testing to research the platform's usability and any impact on awareness regarding cardiovascular diseases.

ACKNOWLEDGMENTS

This research work has been supported by the Spanish *Ministry of Education and Vocational Training* under an FPU fellowship (FPU17/03276). We thank all primary care physicians and personnel helping with the development of the study. We thank Philips Iberica and Obra Social 'La Caixa' for their support. We especially thank participants in the study and apologize for any inconvenience we could have caused. We thank the involvement of the Salamanca patient organisation 'El paciente experto', for providing counselling to Salmanticor and for further promoting the dissemination of the results to the society and to the regional government.

REFERENCES

[1] A. Timmis *et al.*, "European Society of Cardiology: Cardiovascular Disease Statistics 2019," (in eng), *Eur Heart J,* vol. 41, no. 1, pp. 12-85, Jan 1 2020, doi: 10.1093/eurheartj/ehz859.

[2] T. A. Pearson *et al.*, "American Heart Association Guide for Improving Cardiovascular Health at the Community Level, 2013 update: a scientific statement for public health practitioners, healthcare providers, and health policy makers," (in eng), *Circulation,* vol. 127, no. 16, pp. 1730-53, Apr 23 2013, doi: 10.1161/CIR.0b013e31828f8a94.

[3] G. A. Roth *et al.*, "Global, Regional, and National Burden of Cardiovascular Diseases for 10 Causes, 1990 to 2015," (in eng), *J Am Coll Cardiol,* vol. 70, no. 1, pp. 1-25, Jul 4 2017, doi: 10.1016/j.jacc.2017.04.052.

[4] C. O. Wilke, *Fundamentals of data visualization: a primer on making informative and compelling figures*. O'Reilly Media, 2019.

[5] D. Patil and H. Mason, *Data Driven*. " O'Reilly Media, Inc.", 2015.

[6] A. Cairo, *How Charts Lie: Getting Smarter about Visual Information*. WW Norton & Company, 2019.

[7] A. Cairo, *The Functional Art: An introduction to information graphics and visualization*. New Riders, 2012.

[8] A. Cairo, *The truthful art: Data, charts, and maps for communication*. New Riders, 2016.

[9] M. Brehmer and T. Munzner, "A multi-level typology of abstract visualization tasks," *IEEE transactions on visualization and computer graphics,* vol. 19, no. 12, pp. 2376-2385, 2013.

[10] H. Lam, M. Tory, and T. Munzner, "Bridging from goals to tasks with design study analysis reports," *IEEE transactions on visualization and computer graphics,* vol. 24, no. 1, pp. 435-445, 2017.

[11] T. Munzner, "A nested process model for visualization design and validation," *IEEE Transactions on Visualization Computer Graphics,* no. 6, pp. 921-928, 2009.

[12] T. Munzner, *Visualization analysis and design*. Boca Raton, FL, USA: AK Peters/CRC Press, 2014.

[13] D. Dowding *et al.*, "Dashboards for improving patient care: review of the literature," *International journal of medical informatics,* vol. 84, no. 2, pp. 87-100, 2015.

[14] A. Sarikaya, M. Correll, L. Bartram, M. Tory, and D. Fisher, "What Do We Talk About When We Talk About Dashboards?," *IEEE Transactions on Visualization Computer Graphics,* vol. 25, no. 1, pp. 682 - 692, 2018.

[15] A. V. Pandey, A. Manivannan, O. Nov, M. Satterthwaite, and E. Bertini, "The persuasive power of data visualization," *IEEE transactions on visualization and computer graphics,* vol. 20, no. 12, pp. 2211-2220, 2014.

[16] J. I. Melero-Alegria *et al.*, "SALMANTICOR study. Rationale and design of a population-based study to identify structural heart disease abnormalities: a spatial and machine learning analysis," (in eng), *BMJ Open,* vol. 9, no. 2, p. e024605, Feb 13 2019, doi: 10.1136/bmjopen-2018-024605.

[17] M. Bostock, V. Ogievetsky, and J. Heer, "D³ data-driven documents," *IEEE transactions on visualization and computer graphics,* vol. 17, no. 12, pp. 2301-2309, 2011.

[18] E. Meeks, *D3. js in Action: Data Visualization with JavaScript*. Manning, 2018.

[19] A. Vázquez-Ingelmo, F. J. García-Peñalvo, and R. Therón, "Towards a Technological Ecosystem to Provide Information Dashboards as a Service: A Dynamic Proposal for Supplying Dashboards Adapted to Specific Scenarios," *Applied Sciences,* vol. 11, no. 7, p. 3249, 2021. [Online]. Available: https://www.mdpi.com/2076-3417/11/7/3249.

[20] S. Ribecca. "Chart Combinations: Tile Grid Maps." DataViz Catalogue. http://datavizcatalogue.com/blog/chart-combinations-tile-grid-maps/ (accessed 2021).

[21] T. V. Pollet, G. Stulp, S. P. Henzi, and L. Barrett, "Taking the aggravation out of data aggregation: A conceptual guide to dealing with statistical issues related to the pooling of individual-level observational data," *American journal of primatology,* vol. 77, no. 7, pp. 727-740, 2015.

[22] A. Vázquez-Ingelmo, F. J. García-Peñalvo, and R. Therón, "Aggregation Bias: A Proposal to Raise Awareness Regarding Inclusion in Visual Analytics," in *World Conference on Information Systems and Technologies*, 2020: Springer, pp. 409-417.

[23] M. Correll, D. Moritz, and J. Heer, "Value-suppressing uncertainty palettes," in *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, 2018, pp. 1-11.

[24] B. Shneiderman, "The eyes have it: A task by data type taxonomy for information visualizations," in *Proceedings 1996 IEEE symposium on visual languages*, 1996: IEEE, pp. 336-343.

7.29 Appendix AC. Application of Artificial Intelligence Algorithms Within the Medical Context for Non-Specialized Users: the CARTIER-IA Platform

Application of Artificial Intelligence Algorithms Within the Medical Context for Non-Specialized Users: the CARTIER-IA Platform

Francisco José García-Peñalvo[1], Andrea Vázquez-Ingelmo[1], Alicia García-Holgado[1], Jesús Sampedro-Gómez[2], Antonio Sánchez-Puente[2], Víctor Vicente-Palacios[3], P. Ignacio Dorado-Díaz[2], Pedro L. Sánchez[2] *

[1] GRIAL Research Group, University of Salamanca (Spain)
[2] Cardiology Department, Hospital Universitario de Salamanca, SACyL. IBSAL, Facultad de Medicina, University of Salamanca, and CIBERCV (ISCiii) (Spain)
[3] Philips Healthcare (Spain)

Received 1 March 2021 | Accepted 19 April 2021 | Published 13 May 2021

LA UNIVERSIDAD EN INTERNET

Abstract

The use of advanced algorithms and models such as Machine Learning, Deep Learning and other related approaches of Artificial Intelligence have grown in their use given their benefits in different contexts. One of these contexts is the medical domain, as these algorithms can support disease detection, image segmentation and other multiple tasks. However, it is necessary to organize and arrange the different data resources involved in these scenarios and tackle the heterogeneity of data sources. This work presents the CARTIER-IA platform: a platform for the management of medical data and imaging. The goal of this project focuses on providing a friendly and usable interface to organize structured data, to visualize and edit medical images, and to apply Artificial Intelligence algorithms on the stored resources. One of the challenges of the platform design is to ease these complex tasks in a way that non-AI-specialized users could benefit from the application of AI algorithms without further training. Two use cases of AI application within the platform are provided, as well as a heuristic evaluation to assess the usability of the first version of CARTIER-IA.

Keywords

Information System, Medical Data Management, Medical Imaging Management, Artificial Intelligence, Health Platform.

DOI: 10.9781/ijimai.2021.05.005

I. Introduction

Artificial Intelligence (AI) algorithms have grown in popularity and increased its range of uses over the years. The possibility of applying them to different problems and contexts provide a wide support in complex scenarios in which data is continuously being generated.

One of these complex scenarios is the medical context. These algorithms and approaches are becoming very relevant when analyzing medical data [1]. However not only structured or tabular data can be involved in this context; medical imaging are also crucial resources within the medical domain.

The analysis of medical imaging involves complex tasks such as disease detection, segmentation, assessment of organ functions, etc. [2]-[4]. In this sense, artificial intelligence algorithms can provide support to these tasks with similar performance compared to human skills [5].

However, as introduced, data is being continuously generated in medical scenarios, which makes its management a convoluted responsibility. In fact, not only several data sources can be involved, but also different data structures. This data heterogeneity is a challenge both for its management and the application of AI algorithms.

Because of this, one of the main challenges of applying AI algorithms in real medical scenarios relies on the unification and accessibility of the generated data. For this reason, information systems are required to gather, clean, organize and structure data in order to apply AI algorithms in a friendly, secure and anonymized manner.

This work presents a platform for the management of structured data and imaging resources in the medical context with advanced features such as their visualization, edition and application of AI on the stored resources.

Powerful tools such as information dashboards can be easily integrated in the platform [6], [7] to explore structured data. This kind of tools provide support to knowledge generation, which is very relevant in this context [8].

On the other hand, DICOM editors and AI integration are also crucial components, which allow the modification and advanced exploration of imaging data.

* Corresponding author.

E-mail addresses: fgarcia@usal.es (F. J. García-Peñalvo), andreavazquez@usal.es (A. Vázquez-Ingelmo), aliciagh@usal.es (A. García-Holgado), jmsampedro@saludcastillayleon.es (J. Sampedro-Gómez), asanchezpu@saludcastillayleon.es (A. Sánchez-Puente), victor.vicente.palacios@philips.com (V. Vicente-Palacios), pidorado@saludcastillayleon.es (P. I. Dorado-Díaz), plsanchez@saludcastillayleon.es (P. L. Sánchez)

The starting point of this project stems from the need of using a collaborative platform to gather these heterogeneous types of data. Unifying medical data sources through a collaborative platform eases their exploration and analysis, as well as enabling the possibility of sharing knowledge across different projects.

In this case, the platform was built for research purposes in the field of cardiology, but its flexibility enables its use for other fields in which structured data and medical imaging need to be unified.

In fact, the features of this platform can also provide support to educational purposes, in which the application of AI scripts is guided and explained to novice or non-specialized users [9], [10].

Relying on a web-collaborative platform also allows the integration of artificial intelligence algorithms. In fact, cardiac imaging is particularly interesting for the application of AI algorithms, since many tasks are related to the assessment of volumes, distances and motion of different structures in the heart and, in this regard, deep learning techniques have been proven to achieve good results [11].

Cardiac imaging is usually composed of DICOM (Digital Imaging and Communication On Medicine) files [12] from echocardiographic, magnetic resonance, or computed tomography, among the most important. Other online medical imaging platforms implement the DICOM protocol and are available for these purposes, some of them even in an open-source format [13].

On the other hand, there exist solutions based on application programming interfaces with pretrained models' repositories for use in the medical imaging field. Nevertheless, these repositories are mostly oriented towards advanced users with expertise in programming and data science knowledge [14].

As can be seen, different solutions arise to manage medical imaging and execute AI algorithms over them. However, in this scenario, it is necessary not only to unify data sources, but also these kinds of services.

For these reasons, the development of a technological ecosystem [15], [16] is an appropriate solution to merge both functionalities into a user-friendly web-based interface.

The CARTIER-IA platform can be seen as a technological ecosystem that support all data-management related tasks (including structured data and medical imaging collection) and also enable both healthcare professionals and data scientists to apply AI models to the stored images.

Deep learning and machine learning models can be stored by AI developers through Python scripts, as it will be detailed in section 2.C. Using this approach, scripts can be executed through the web interface by any user interested in analyzing the image, with the goal of providing the benefits of these scripts without requiring python-programming skills nor advanced knowledge regarding Artificial Intelligence algorithms.

In this respect, given the fact that not every user is skilled in programming AI algorithms, the platform needs a user-centered approach to provide friendly interfaces and bring AI-driven tasks closer to non-specialized users.

In this work we present the integration of AI algorithms into the CARTIER-IA platform's DICOM viewer and editor. A heuristic evaluation of the tool is provided to test its usability and improve the image processing and AI application workflow with the goal of offering better user experience.

The structure of this paper is as follows: section II outlines the technical details of the platform as well as the heuristic evaluation methodology, section III explains the main functionality blocks of the CARTIER-IA platform, section IV describes two use cases of the AI integration within the platform and section V provides the heuristic evaluation results regarding the image editor and script application tool. Finally, section VI and section VII discuss the results and present the conclusions, respectively.

II. Methodology

A. Technical Details of the Platform

The platform relies on different technologies and frameworks which are integrated using a client-server architecture.

The front-end employs HTML, CSS, and JavaScript to send data to the server. On the other hand, the DICOM viewer and editor is also located at the front end, and it is implemented through the Cornerstone.js library.

On the other hand, the back end performs more complex tasks to fulfil the requirements of the platform, such as the data storage, data processing and an Artificial Intelligence environment.

The technology employed to implement this client-server approach as a web application is Django, a Python-based web framework [17]. The web application is also connected through web requests to other services such as a REDCap instance to manage additional projects and information.

Due to the necessity of pre-validate DICOM images and structured data, upload processes can be time-consuming tasks. For this reason, job queries have been implemented to carry out these data uploads asynchronously as background jobs. This allow users to navigate the platform while their data is uploading.

Finally, to implement the integrated AI environment, the back end is supported by libraries such as OpenCV and TensorFlow, in order to enable the execution of deep learning models and other AI-related scripts.

B. Usability Study: Heuristic Evaluation

Integrating complex tasks such as AI algorithms in a web interface in which other diverse functionalities are involved is a challenge, especially regarding providing a good user experience.

For this reason, the platform needs to be thoroughly tested in terms of its usability. One of the preliminary studies that has been carried out to identify interface design weaknesses in the platform's first version is a heuristic evaluation.

Although there are several heuristics sets to perform heuristic evaluations, the most popular are the ten heuristics by Nielsen [18]. There are also specific heuristics related to the medical domain, but there aren't focused on this kind of platforms (they are mostly related to the evaluation of Electronic Health Records [19], [20]).

However, due to the fact that CARTIER-IA platform is mainly focused on research tasks, image edition and AI algorithms application, the previous heuristics are not the best fit for this usability study.

For these reasons, the Nielsen's heuristics were the selected instrument to perform the heuristic evaluation on the CARTIER-IA platform. This set is composed of ten heuristics, which are listed below [18].

HR1: Visibility of system status.

HR2: Match between system and the real world.

HR3: User control and freedom.

HR4: Consistency and standards.

HR5: Error prevention.

HR6: Recognition rather than recall.

HR7: Flexibility and efficiency of use.

HR8: Aesthetic and minimalist design.

HR9: Help users recognize, diagnose, and recover from errors.

HR10: Help and documentation.

A total of six experts were involved in the heuristic evaluation. Four of these experts were HCI experts (web developers and researchers), and two of them both HCI experts and domain experts (a Ph.D.

student and clinical data scientist) [21]. In fact, these double experts had used the CARTIER-IA platform as users before performing the heuristic evaluation.

The heuristic evaluation was carried out using a template with guidelines to support the evaluation and issues' reporting. Each evaluator had only access to his/her own report, in order to avoid biases. The evaluation template had three fields to collect the evaluator's name, the name of the tool evaluated, and the browser that they employed to access the platform.

Finally, the template provided a table with three columns (heuristic name, score from 1 to 10 and problems detected) and one row per problem detected within each heuristic.

III. The CARTIER-IA Platform

A. Data Collection

As introduced in section I, one of the motivations of developing the platform is to unify data from different sources and arrange them into a more friendly structure. Due to this requirement, the CARTIER-IA platform provides two types of data upload processes.

First, a structured data uploader. The platform allows users to upload spreadsheets of data at different levels, containing information associated to patients, image studies or files. The platform also supports longitudinal structures (repeated measurements or data for the same patient over time).

In addition, data schemas are flexible to vary among different projects, so a project might contain a structured data schema completely different from another.

This flexibility is accomplished through the Django ORM (a database-abstraction API). The Django ORM API provides access to a relational database with the structure shown in Fig. 1. The platform is mainly organized through projects, which will hold data from different patients. Structured data is stored as JSON object at different levels (patient, study or file), which provides the support to modify the data schemas across projects.

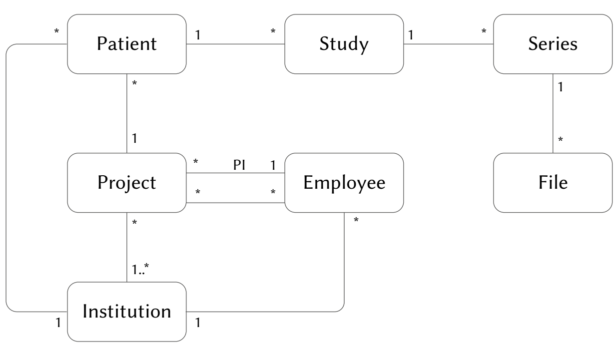


Fig. 1. Schematic overview of the platform's data structuration.

On the other hand, the platform provides a second data uploader; an image data uploader. This uploader allows users to upload a set of image studies through compressed files. The service handles the DICOM format, read some of the metadata tags, check the integrity of data anonymization of each file and validates the metadata against already stored structured data, linking them if applicable.

In this regard, imaging data relies on three entities: studies, series, and files. These entities provide the structure to manage data about the DICOM images that will be uploaded to the platform.

The image files are referenced through these entities and stored through file systems. The type of file system can be modified, allowing flexibility in the selection of the storage technology.

Finally, the platform integrates an external tool to make the data management more powerful. A REDCap instance is connected to the platform to enable the importation of its data, thus providing another layer of data unification. REDCap (Research Electronic Data Capture) is an electronic data capture (EDC) software and workflow methodology for creating and designing clinical research databases [22].

B. Image Edition

Another main functionality of the platform is its image editor. When DICOM images are uploaded, users can explore them more closely through this tool. One of the benefits of this image editor is that is fully integrated within the platform, so it is not necessary to use external tools to carry out image modifications.

This is possible because, as explained in section II, image edition takes place in the browser through the Cornerstone.js (https://github.com/cornerstonejs/cornerstone) framework, an open-source library to parse and render DICOM files.

Thanks to this approach, users can edit the images they are currently exploring, and decide later if they can make these annotations and modifications persistent.

These modifications are not stored along the image itself, but as JSON objects containing all the necessary meta-data regarding the carried-out modifications or annotations. By storing the modifications as standalone objects, it is possible to explore the annotations made by other users, compare them against each other or even to have a version control of the modifications on each image.

The majority of image edition tools and functionalities are supported by another open-source framework, which provides an extensible solution for creating tools on top of Cornerstone.js (https://github.com/cornerstonejs/cornerstoneTool). Specifically, the following tools are available through the image editor:

- Brush and scissors tools for image segmentation
- Segmentation layers and brush size selectors to ease the segmentation process of the images
- Length and area tools to measure image fragments
- Annotation tools
- Zoom tools
- A crop tool (computed on the backend)
- A tool to apply uploaded and validated AI scripts (computed on the backend)

C. Artificial Intelligence Support

The feature in which this paper is focused is the Artificial Intelligence integration within the platform. This feature has two main motivations:

- To offer the benefits from Artificial Intelligence algorithms in situ, without the necessity of leaving the platform to applying these algorithms
- To provide a friendly interface to apply AI scripts and open their use to non-specialized users

This feature allows researchers to upload their AI scripts into the platform and make them available to other users. Only researchers with privileges can add new scripts, which need to be thoroughly tested by the corresponding researcher before integrating them into the platform to ensure a reliable functionality.

The platform also provides an uploader to define the algorithm's meta-data. In this case, algorithms' meta-data is highly important to properly integrate the scripts within the platform. These meta-data provide information about the algorithm's output (a modified image, a set of measures, a segmentation mask, etc.), its applicability (as their

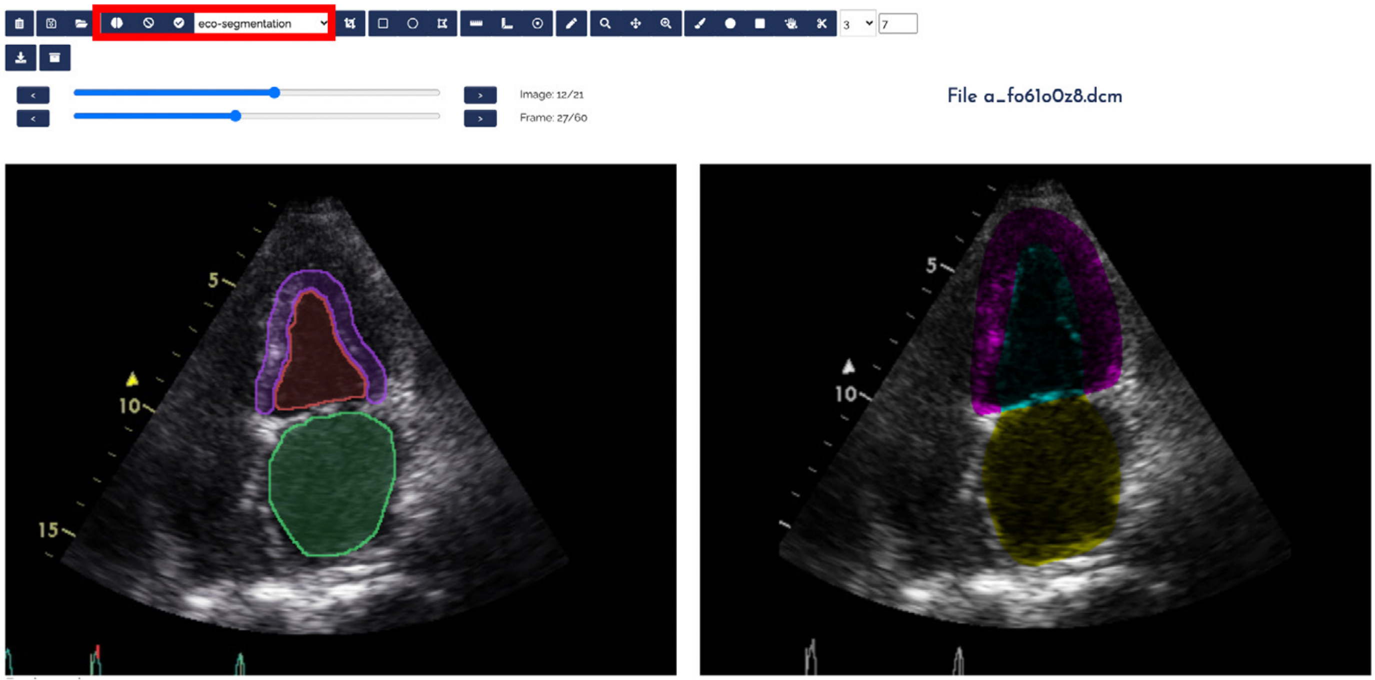


Fig. 2. Screenshot of a manual segmentation (left) and the AI algorithm output (right).

application might be limited to specific DICOM modalities) or other parameters depending on the output.

It is important to clarify that the algorithms need to be pre-trained before their integration into the platform. For this reason, the uploader also provides a field to upload the exported model or the models' weights depending on the type of AI algorithm employed.

To sum up, to integrate an algorithm into the platform it is necessary to provide the pre-trained model, and the script that makes use of the pre-trained model with the goal of enabling their invocation by the platform's AI module.

Once an algorithm has been integrated, it will be available at image editor. To apply an algorithm, the user just needs to click the AI button and select one of the available scripts for the current image being displayed. When the user confirms the application, the platform will yield the result which, depending on the algorithm's output type, could result in displaying a new image, an inferred diagnosis or the addition of AI-driven measurements as new structured data.

IV. Use Cases

This section provides two application uses of the AI integration within the CARTIER-IA platform as an example of how the platform behaves when dealing with different types of AI algorithms.

A. Manual vs. Artificial Intelligence Segmentation

Segmentation of medical images is a relevant procedure within the field of medical image processing. Its ultimate goal is to identify different elements and features in medical images to detect abnormalities or other characteristics of interest.

For this reason, one of the most relevant features of the image viewer is the possibility of performing the segmentation of the stored DICOM images in place.

As explained in the previous section, the image viewer relies on different tools to provide a complete set of image processing functionalities. Among them, the platform offers different brushes to perform image segmentations manually and store them as JSON objects that can be retrieved and further processed.

But along with the manual segmentation, researchers can integrate deep learning models whose outputs are automatically generated segmentations. To do that, users can select among the available AI algorithms in the platform and simply confirm their choice (red rectangle in Fig. 2). The algorithm choice is processed in the back end, which consults the algorithm's meta-data and, depending on the output, performs different actions. In this case, the output is a segmented image, so this result is sent back to the client and displayed in the viewer next to the original image (Fig. 2).

This interface organization allows users to compare their manual segmentation with the algorithm's result, which could provide new information or assist the user with their own image segmentation. And it can also be used in the reverse scenario. If it is a trained physician or technician who performs the manual segmentation, this interface could be used to improve the artificial intelligence algorithm by active learning.

B. Measurements

The application of AI scripts is not only limited to image segmentation. As explained throughout this work, the platform manages both imaging data and structured data. Structured data also provides crucial information regarding patients and their monitoring, diagnoses, treatments, etc.

In this context, the CARTIER-IA platform also supports the execution of AI algorithms that, based on the input image or even input structured data, return a dataset containing new inferred information. The algorithm's results are persistently stored along with the rest of the patient's, study's or file's structured data, making them available for other users when exploring the project.

The process to apply these kinds of algorithms is exactly the same. However, in this case, instead of returning a new image, the back end executes the algorithm, stores the newly generated variables and sends a confirmation to the client (Fig. 3).

After the confirmation, users can see the measures yielded by the algorithm at the specified level. Fig. 4 shows a new variable generated by the algorithm "ai_DummyECO-script", which is stored under a new

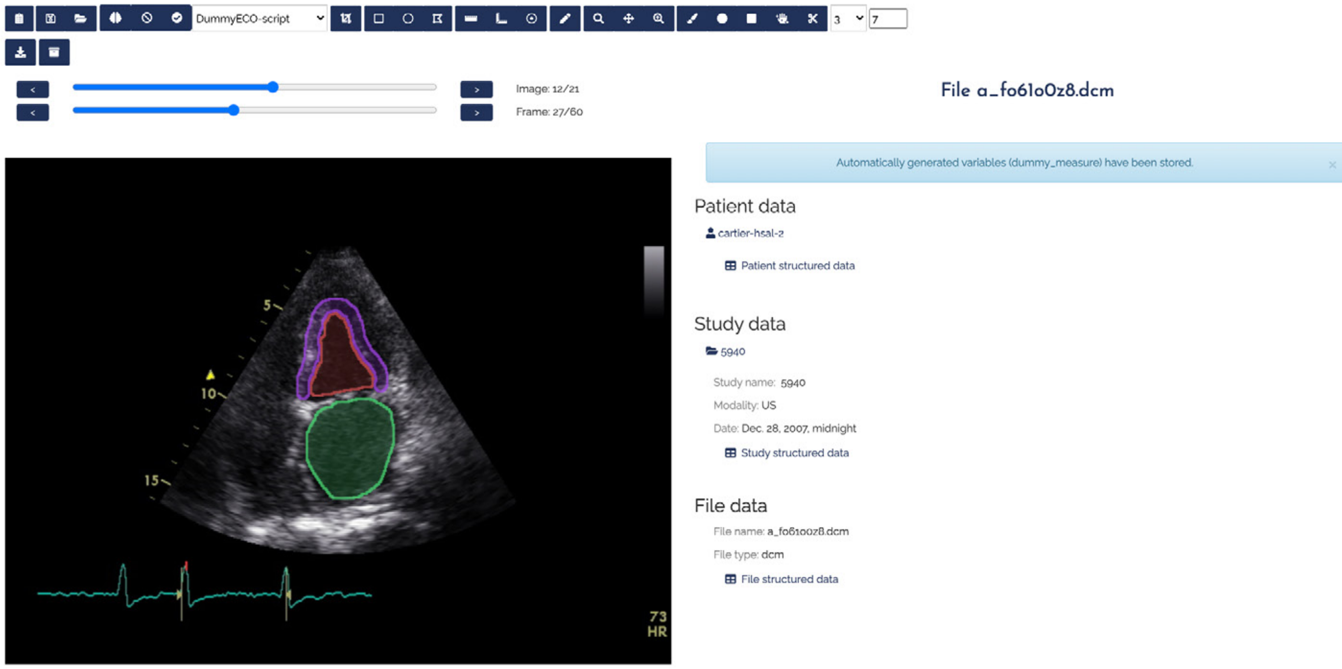


Fig. 3. Screenshot of AI algorithms' measurements output.

Fig. 4. Automatically generated variables after applying a measurement AI algorithm.

category with the format "<algorithm_name>_vars" to differentiate them from the original study variables.

V. Heuristic Evaluation Results

Each expert was identified by a number (E1, E2, E3, E4, E5, E6) in order to present the outputs of the heuristic evaluation. The heuristic evaluation was performed on the whole platform, but only the DICOM editor and AI tool-related issues are being discussed given the focus of this work.

Fig. 5 shows the total average value assigned to the problems identified under each heuristic. Values close to 1 indicate that experts detected non-relevant issues, and values close to 10 implies that the issues are relevant and severe. A zero value represents that experts did not identify any problems in that heuristic. Not only is the severity of the problems important, but also the absolute number of issues to solve in each heuristic (Fig. 6).

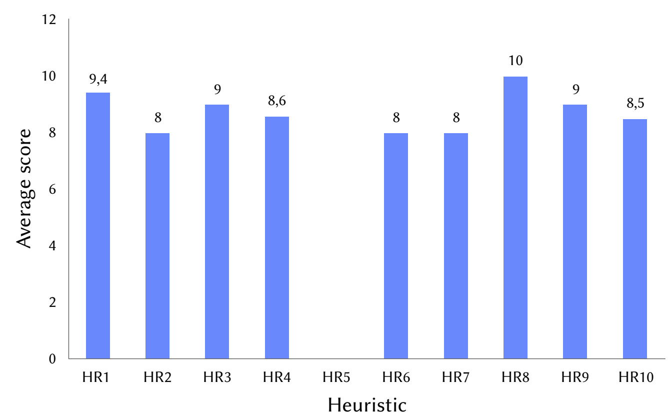


Fig. 5. Average score for each heuristic rule regarding the DICOM editor and AI tool.

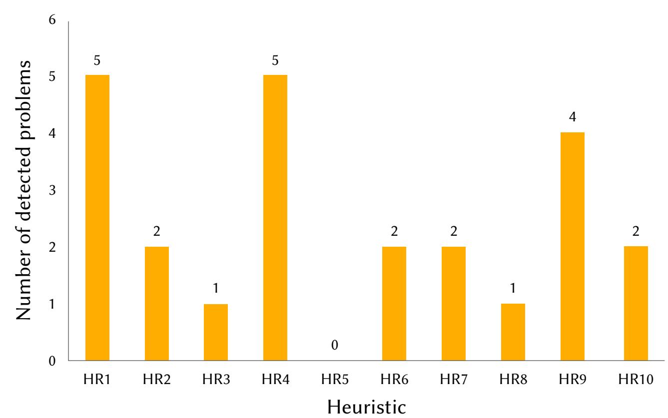


Fig. 6. The total number of detected problems regarding the DICOM editor and AI tool per Nielsen's heuristic.

The heuristics with the largest number of usability issues identified for the DICOM viewer and AI tool were HR1 (*Visibility of system status*) and HR4 (*Consistency and standards*), both with 5 problems detected. The average severity scores for these heuristics are 9.4 and 8.6, respectively.

The HR1 problems are mainly related to the absence of progress bars and the actions that can take place within the image editor, which are not clearly explained.

Regarding HR4-related issues, they are focused on the correctness of the metaphors used for the icons that represent each functionality button. In addition, some misfunctions on these image processing functionalities were also identified.

On the other hand, there are three heuristics that also obtained high severity ratings: HR3 and HR9 with a score of 9 and HR8 with a score of 10.

Regarding HR8 (*Aesthetic and minimalist design*), this high score is due to the fact that only one issue was encountered within this category (also related with the DICOM editor's icons), but E6 assigned a score of 10 because of its relevance. Specifically, this issue pointed out the great quantity of icons employed for the editor toolbar and their difficulty to clearly convey their meaning.

Only one issue was identified under HR3 (*User control and freedom*) but also with a high severity rating (9). In this case, E6 identified the impossibility of undo or redo actions taken place within the tool.

Four issues were identified in the HR9 (*Help users recognize, diagnose, and recover from errors*), obtaining a score of 9, too. In this case, experts identified the lack of information when an AI script or a DICOM image fails and the impossibility of recovering from this kind of errors. This heuristic also includes the issue that the DICOM editor does not support the reset of the modifications made on the images.

Finally, a lower number of issues were encountered in the rest of heuristics:

- Better explanations regarding the editor's functionalities (HR2 and HR6)
- Better explanations regarding the results yielded by the functionalities supported on the editor (HR6)
- Keyboard shortcuts and AI integration for advanced users (HR7)
- Lack of documentation, specifically regarding the AI tool, which could be complex to understand (HR10)

VI. Discussion

The heuristic evaluation identified different design issues regarding the image editor and AI algorithms' application. These usability evaluations are crucial to iteratively provide more robust and friendly interfaces to perform complex tasks such as the ones supported by the CARTIER-IA platform.

The results derived from the heuristic evaluation shown very high scores. This is due the great relevance that experts gave to usability in the image editor tool. The image editor is a powerful component of the CARTIER-IA platform, because not only provides edition functionalities, but also is the integration point for applying AI algorithms. For these reasons, offering good user experience in the image editor interface is crucial, and thus every usability issue encountered has high relevance.

The majority of issues were related to the toolbar, which relies on several icons to depict the supported functionalities. However, these icons were not very clear to the experts. In addition to this topic, some experts also pointed out the necessity of explaining the functionalities more thoroughly, especially the AI algorithms.

In this version of the platform, algorithms are listed in the interface and the user can apply them directly. However, only the names of the algorithms are displayed, which can be confusing, as the algorithm's name could provide little or no information at all regarding its outputs and results.

One of the design parameters for the AI integration was to make the application process straightforward both for skilled and non-skilled users. However, simplifying too much this process can also have drawbacks. Non-skilled users could question the algorithms' outputs if there are no further explanations regarding the process nor the interpretation of the results, because the AI tool works as a black box in its current version.

It is crucial to find balance between implementing a simple interface but also displaying enough information to understand the actions carried out within the platform.

Providing user-friendly interfaces in the health domain could make convoluted tasks more straightforward and thus, save time for physicians. As it has been shown, this kind of interfaces could also bring closer tasks for which users are not specialized nor trained (such as AI algorithms programming).

Another important benefit from integrating AI algorithms in a medical data management platform is that the trained models can be improved. Although in its current version the platform only provides an interface for executing AI algorithms (because models need to be pre-trained before their integration into the platform), this approach sets the foundations for future improvements, including the possibility of training AI models directly from the platform.

For example, manual segmentations can be carried-out by the users within the platform, which results in new data to train the existing models. On the other hand, users can also label the algorithms' outputs depending on their performance, thus laying the foundations for improving the models through active learning.

On the other hand, there is room for improvement regarding the algorithms' validation. Currently, researchers are responsible of the validation of their scripts, but another validation layer can be implemented to analyze and test these scripts automatically before carrying out the integration. The metrics obtaining from the testing of the scripts could complement the information of each algorithm to generate more confidence regarding the platform's AI support.

Finally, we want to mention that a heuristic evaluation does not ensure identifying all the problems that could affect a real user in a real context while using the platform. There are studies that point out that the problems detected by experts are not necessarily the actual problems that will affect the end users of the platform [23]. To alleviate this limitation, we included one expert with extensive knowledge of the platform's domain, although subsequent research will explore usability from the researchers and physicians' point of view.

VII. Conclusions

This paper presents a collaborative platform for the management of medical data and imaging. The platform has several features to provide support for a variety of functionalities, such as a DICOM viewer and editor. Among these tools there is the possibility of integrating artificial intelligence scripts to make the application process straightforward to non-specialized users.

Given the implication of all these features within the platform, a heuristic evaluation has been carried out to identify usability issues of the DICOM viewer and AI algorithms' integration in the current version of the platform. This evaluation gives hints on the aspects that need to be improved to provide better user experience to researchers and physicians.

Future research lines will involve the resolution of every usability issue identified, as well further usability tests including other techniques such as the PSSUQ questionnaire or usability labs.

Acknowledgment

This research work has been supported by the Spanish *Ministry of Education and Vocational Training* under an FPU fellowship (FPU17/03276). This work was also supported by national (PI14/00695, PIE14/00066, PI17/00145, DTS19/00098, PI19/00658, PI19/00656 Institute of Health Carlos III, Spanish Ministry of Economy and Competitiveness and co-funded by ERDF/ESF, "Investing in your future") and community (GRS 2033/A/19, GRS 2030/A/19, GRS 2031/A/19, GRS 2032/A/19, SACYL, Junta Castilla y León) competitive grants.

References

[1] A. Rajkomar, J. Dean, and I. Kohane, "Machine Learning in Medicine," (in eng), *N Engl J Med,* vol. 380, no. 14, pp. 1347-1358, 04 2019, doi: 10.1056/NEJMra1814259.

[2] G. Litjens *et al.*, "A survey on deep learning in medical image analysis," (in eng), *Med Image Anal,* vol. 42, pp. 60-88, Dec 2017, doi: 10.1016/j.media.2017.07.005.

[3] S. González Izard, R. Sánchez Torres, Ó. Alonso Plaza, J. A. Juanes Méndez, and F. J. García-Peñalvo, "Nextmed: Automatic Imaging Segmentation, 3D Reconstruction, and 3D Model Visualization Platform Using Augmented and Virtual Reality," (in eng), *Sensors (Basel),* vol. 20, no. 10, p. 2962, 2020, doi: 10.3390/s20102962.

[4] S. G. Izard, J. A. Juanes, F. J. García Peñalvo, J. M. G. Estella, M. J. S. Ledesma, and P. Ruisoto, "Virtual Reality as an Educational and Training Tool for Medicine," *Journal of Medical Systems,* vol. 42, no. 3, p. 50, 2018/02/01 2018, doi: 10.1007/s10916-018-0900-2.

[5] X. Liu *et al.*, "A comparison of deep learning performance against health-care professionals in detecting diseases from medical imaging: a systematic review and meta-analysis," *The lancet digital health,* vol. 1, no. 6, pp. e271-e297, 2019.

[6] A. Vázquez-Ingelmo, F. J. García-Peñalvo, and R. Therón, "Information Dashboards and Tailoring Capabilities - A Systematic Literature Review," *IEEE Access,* vol. 7, pp. 109673-109688, 2019, doi: 10.1109/ACCESS.2019.2933472.

[7] A. Vázquez-Ingelmo, F. J. García-Peñalvo, R. Therón, D. A. Filvà, and D. F. Escudero, "Connecting domain-specific features to source code: towards the automatization of dashboard generation," *Cluster Computing. The Journal of Networks, Software Tools and Applications,* p. In Press, 2020, doi: 10.1007/s10586-019-03012-1.

[8] A. Sarikaya, M. Correll, L. Bartram, M. Tory, and D. Fisher, "What Do We Talk About When We Talk About Dashboards?," *IEEE Transactions on Visualization Computer Graphics,* vol. 25, no. 1, pp. 682 - 692, 2018.

[9] Y. Zhonggen, "Visualizing Artificial Intelligence Used in Education Over Two Decades," *Journal of Information Technology Research (JITR),* vol. 13, no. 4, pp. 32-46, 2020, doi: 10.4018/JITR.2020100103.

[10] J. C. Sánchez-Prieto, J. Cruz-Benito, R. Therón Sánchez, and F. J. García Peñalvo, "Assessed by Machines: Development of a TAM-Based Tool to Measure AI-based Assessment Acceptance Among Students," *International Journal of Interactive Multimedia and Artificial Intelligence,* vol. 6, no. 4, p. 80, 2020.

[11] O. Bernard *et al.*, "Deep Learning Techniques for Automatic MRI Cardiac Multi-Structures Segmentation and Diagnosis: Is the Problem Solved?," *IEEE Trans Med Imaging,* vol. 37, no. 11, pp. 2514-2525, Nov 2018, doi: 10.1109/TMI.2018.2837502.

[12] P. Mildenberger, M. Eichelberg, and E. Martin, "Introduction to the DICOM standard," *European radiology,* vol. 12, no. 4, pp. 920-927, 2002.

[13] D. S. Marcus, T. R. Olsen, M. Ramaratnam, and R. L. Buckner, "The Extensible Neuroimaging Archive Toolkit: an informatics platform for managing, exploring, and sharing neuroimaging data," (in eng), *Neuroinformatics,* vol. 5, no. 1, pp. 11-34, 2007, doi: 10.1385/ni:5:1:11.

[14] E. Gibson *et al.*, "NiftyNet: a deep-learning platform for medical imaging," (in eng), *Comput Methods Programs Biomed,* vol. 158, pp. 113-122, May 2018, doi: 10.1016/j.cmpb.2018.01.025.

[15] A. García-Holgado and F. J. García-Peñalvo, "Preliminary validation of the metamodel for developing learning ecosystems," in *Fifth International Conference on Technological Ecosystems for Enhancing Multiculturality (TEEM'17) (Cádiz, Spain, October 18-20, 2017)* J. M. Dodero, M. S. Ibarra Sáiz, and I. Ruiz Rube Eds., (ACM International Conference Proceeding Series (ICPS). New York, NY, USA: ACM, 2017.

[16] A. García-Holgado and F. J. García-Peñalvo, "Validation of the learning ecosystem metamodel using transformation rules," *Future Generation Computer Systems,* vol. 91, pp. 300-310, 2019, doi: 10.1016/j.future.2018.09.011.

[17] Django Software Foundation. "Django Web Framework." https://www.djangoproject.com/ (accessed 15/03/2015.

[18] J. Nielsen, "Heuristic evaluation," in *Usability inspection methods*, vol. 17, J. Nielsen and R. L. Mack Eds., no. 1): John Wiley & Sons, Inc., 1994, pp. 25-62.

[19] A. Tarrell, L. Grabenbauer, J. McClay, J. Windle, and A. L. Fruhling, "Toward improved heuristic evaluation of EHRs," *Health Systems,* vol. 4, no. 2, pp. 138-150, 2015/07/01 2015, doi: 10.1057/hs.2014.19.

[20] D. Armijo, C. McDonnell, and K. Werner, *Electronic health record usability: Evaluation and use case framework. AHRQ Publication No. 09(10)-0091-1-EF.* Rockville, MD: Agency for Healthcare Research and Quality, 2009.

[21] J. Nielsen, "Finding usability problems through heuristic evaluation," in *CHI '92: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems.* New York, NY, USA: ACM, 1992, pp. 373–380.

[22] P. A. Harris, R. Taylor, R. Thielke, J. Payne, N. Gonzalez, and J. G. Conde, "Research electronic data capture (REDCap)—a metadata-driven methodology and workflow process for providing translational research informatics support," *Journal of biomedical informatics,* vol. 42, no. 2, pp. 377-381, 2009.

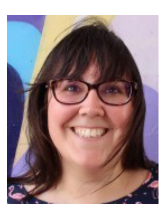
[23] R. Khajouei, A. Ameri, and Y. Jahani, "Evaluating the agreement of users with usability problems identified by heuristic evaluation," *International Journal of Medical Informatics,* vol. 117, pp. 13-18, 2018/09/01/ 2018, doi: https://doi.org/10.1016/j.ijmedinf.2018.05.012.

Francisco José García-Peñalvo

He received the degrees in computing from the University of Salamanca and the University of Valladolid, and a Ph.D. from the University of Salamanca (USAL). He is Full Professor of the Computer Science Department at the University of Salamanca. In addition, he is a Distinguished Professor of the School of Humanities and Education of the Tecnológico de Monterrey, Mexico. Since 2006 he is the head of the GRIAL Research Group GRIAL. He is head of the Consolidated Research Unit of the Junta de Castilla y León (UIC 81). He was Vice-dean of Innovation and New Technologies of the Faculty of Sciences of the USAL between 2004 and 2007 and Vice-Chancellor of Technological Innovation of this University between 2007 and 2009. He is currently the Coordinator of the PhD Programme in Education in the Knowledge Society at USAL. He is a member of IEEE (Education Society and Computer Society) and ACM.

Andrea Vázquez-Ingelmo

Andrea Vázquez-Ingelmo received the bachelor's degree in computer engineering from the University of Salamanca, Salamanca, in 2016 and the master's degree in computer engineering from the same university in 2018. She is a member of the Research Group of Interaction and eLearning (GRIAL), where she is pursuing her PhD degree in computer sciences. Her area of research is related to human-computer interaction, software engineering, information visualization and machine learning applications.



Alicia García-Holgado

She received the degree in Computer Sciences (2011), a M.Sc. in Intelligent Systems (2013) and a Ph.D. (2018) from the University of Salamanca, Spain. She is member of the GRIAL Research Group of the University of Salamanca since 2009. Her main lines of research are related to the development of technological ecosystems for knowledge and learning processes management in heterogeneous contexts, and the gender gap in the technological field. She has participated in many national and international R&D projects. She is a member of IEEE (Women in Engineering, Education Society and Computer Society), ACM (and ACM-W) and AMIT (Spanish Association for Women in Science and Technology).

Jesús Sampedro-Gómez

Jesús Sampedro-Gómez is industrial engineer by the Universidad Politécnica of Madrid. He is also a PhD student by the University of Salamanca and works as data scientist in the Cardiology Department of the University Hospital of Salamanca.

Antonio Sánchez-Puente

Antonio Sánchez Puente, PhD, is a junior researcher from CIBER working at the cardiology department of the University Hospital of Salamanca as a data scientist. He was awarded his doctorate in physics by the University of Valencia for his study of gravity theories before turning his career around the application of artificial intelligence in medicine.

Víctor Vicente-Palacios

Víctor Vicente-Palacios holds a PhD from the University of Salamanca in Statistics and works as a Data Scientist at Philips Healthcare in the area of AI applied to Medicine. He is also an alumnus of the Data Science for Social Good program (University of Chicago) and organizer of PyData Salamanca.

P. Ignacio Dorado-Díaz

P. Ignacio Dorado-Díaz holds a PhD from the University of Salamanca in Statistics. He is currently working as a research coordinator in the Cardiology Department of the Hospital de Salamanca in addition to being a professor at the Universidad Pontificia de Salamanca.

Pedro L. Sánchez

Pedro Luis Sánchez holds a doctorate in medicine from the University of Salamanca. He is currently the head of the Cardiology Department at the University Hospital of Salamanca, in addition to being a professor at the University of Salamanca.

7.30 Appendix AD. User-Centered Design Approach for a Machine Learning Platform for Medical Purposes

User-Centered Design Approach for a Machine Learning Platform for Medical Purpose

Alicia García-Holgado[1](✉), Andrea Vázquez-Ingelmo[1], Julia Alonso-Sánchez[1], Francisco José García-Peñalvo[1], Roberto Therón[1], Jesús Sampedro-Gómez[2], Antonio Sánchez-Puente[2], Víctor Vicente-Palacios[3], P. Ignacio Dorado-Díaz[2], and Pedro L. Sánchez[2]

[1] GRIAL Research Group, Research Institute for Educational Sciences, University of Salamanca, Salamanca, Spain
{aliciagh,andreavazquez,juliaalonso,fgarcia,theron}@usal.es
[2] Cardiology Department, Hospital Universitario de Salamanca, SACyL. IBSAL, Facultad de Medicina, University of Salamanca, and CIBERCV (ISCiii), Salamanca, Spain
{jmsampedro,asanchezpu}@saludcastillayleon.es, {acho, plsanchez}@usal.es
[3] Philips Healthcare, Alicante, Spain
victor.vicente.palacios@philips.com

Abstract. Machine learning is increasingly present in different sectors. Decision-making processes that occur in all types of companies and entities can be improved with the use of AI algorithms and machine learning. Furthermore, the application of machine learning algorithms enables the possibility of providing support to automate the undertaking of complex tasks. However, not all users who want to use machine learning are skilled enough from a technological and data science point of view to use many of the tools that are already available on the market. In particular, the health sector is taking advantage of AI algorithms to enhance the decision-making processes and to support complex common activities. Nonetheless, physicians have the domain knowledge but are not deeply trained in data science. This is the case of the cardiology department of the University Hospital of Salamanca, where the large amount of anonymized data makes it possible to improve certain tasks and decision-making. This work describes a machine learning platform to assist non-expert users in the definition and application of ML pipelines. The platform aims to fill data science gaps while automatizing ML pipelines and provides a baseline to integrate it with other developed applications for the cardiology department.

Keywords: Machine learning · User-centered design · Cardiology · Focus group · ML pipelines

1 Introduction

Machine Learning (ML) applications are continuously growing in different fields. The application of ML algorithms enables the possibility of providing support to automate the undertaking of complex tasks.

P. H. Ruiz et al. (Eds.): HCI-COLLAB 2021, CCIS 1478, pp. 237–249, 2021.
https://doi.org/10.1007/978-3-030-92325-9_18

One of the main fields which is benefiting from the application of these algorithms is the health field. The health context involves several tasks including diagnosis, classification, disease detection, segmentation, assessment of organ functions, etc. [1–3], that can be partly automated and enhanced through artificial intelligence support.

Moreover, datasets are constantly being generated through several sources, which provides enough volume of data to apply these kinds of algorithms. However, the application of ML is not trivial; it is necessary to rely on data science and programming skills to select the proper algorithm and preprocess the datasets accordingly.

Given the benefits that can be derived from the application of ML in the medical domain, it is crucial to democratize knowledge regarding algorithms and pipelines, as it could enable non-expert users to support their decision-making processes with AI.

In this work we present a ML platform to assist non-expert users in the definition and application of ML pipelines. The main challenge of this platform is to provide a proper user interface to ease the understanding of the outcomes and processes involved in ML pipelines. For these reasons, we followed a user-centered design approach to capture requirements and needs from a variety of users profiles, including AI experts and non-experts.

The rest of the paper is structured as follows. Section 2 outlines related applications and works to assist users in ML and data science tasks. Section 3 describes the ML platform proposal. Section 4 details the user study carried out to validate the conceptual application, followed by Sect. 5, in which the results are summarized. Finally, Sect. 6 presents the conclusions derived from this work.

2 Related Works

Several tools are already developed for supporting ML processes. We can organize them in three categories. First, tools for developers and data scientist which provide libraries for creating ML applications. For example, TensorFlow, which is a ML system that operates at large scale and in heterogeneous environments [4], helping researchers push the state-of-the-art in ML and developers easily build and deploy ML powered applications. Another example is Apache Mahout, a library for scalable ML on distributed dataflow systems [5]. In this category, we can also include libraries for Python such as PyTorch, Scikit-learn or Keras.io, and cloud services such as Google Colab, which is a serverless Jupyter notebook environment [6].

Second, there are applications that target experts and at the same time provide tools for non-specialist users. There several applications provide visual environments that support the visual definition of ML models.

For example, Weka, a collection of ML algorithms for data mining tasks. It has four environments, specifically, it has a visual interface, Knowledge Flow, that enables users to specify a data stream by graphically connecting components representing data sources, preprocessing tools, learning algorithms, evaluation methods, and visualization tools [7, 8].

RapidMiner Studio provides tools for building ML workflows in a comprehensive data science platform. It has the Visual Workflow Designer tools to create ML workflows, each step is documented for complete transparency. This part of the tool allows to connect the data source, automated in-database processing, data visualization as well as the Model Validation process [9].

SPSS by IBM includes a product for supporting visual data science and ML. The main work area in SPSS Modeler is the stream canvas, an interface to build ML streams connecting nodes [10].

Another example is KNIME Analytics Platform. It provides tools for creating visual workflows for data analytics with a graphical interface, without the need for coding. KNIME is a modular environment, which enables easy visual assembly and interactive execution of a data pipeline [11].

On the other hand, the ML has started to introduce in primary and secondary education. This has prompted the development of tools to help non-expert users, such as children, to perform simple ML processes using a visual interface. In particular, there are two tools that are noteworthy. LearningML [12], a tool to foster computational thinking skills through practical AI Projects, and Machine Learning for Kids. Both tools are based on a simple pipeline used to train models and an integration with Scratch to use the trained model.

As can be seen, there are plenty of powerful applications focused on easing the application of ML algorithms as well as educational tools to understand these complex workflows. However, our application context asks for a customized tool with specific requirements related to the health sector and more emphasis on providing an educational experience to those unskilled users while using the platform. The next sections will outline these requirements and our approach to address the necessities regarding the automatization of ML pipelines in this domain.

3 Platform Definition

3.1 The Problem

As introduced before, the health sector is taking advantage of AI algorithms to enhance the decision-making processes and to support complex common activities such as image segmentation, disease detection, identification of risk factors, etc. However, to benefit from these algorithms, it is important to rely on robust data science skills. In fact, in the health sector, it is necessary to rely both on data science skills as well as on domain knowledge to get the most out of the application of AI pipelines in the health sector.

Having both data science skills and domain knowledge is very powerful, but it is also very difficult, as physicians are not usually (deeply) trained in data science, and data scientists might not be specialized in specific domains. For these reasons, it is necessary to fill these skills and domain knowledge gaps, which is usually addressed by having multidisciplinary teams.

However, bringing AI and data science concepts closer to physicians would be more efficient, as they would be less dependent of data scientists. The presented tool lay its foundations on this specific issue: to fill data science gaps while automatizing ML pipelines.

Section 2 proved that there are already plenty of tools that address the automatization of AI and data science pipelines, but the necessity of adapting them to the user needs identified in the medical sector asks for a customized tool. In fact, although these tools are mostly generic and powerful, we want to focus on enhanced interactivity to improve the engagement of physicians while still providing all the benefits derived from the introduction of ML pipelines in medical departments, as well as an integrated on-going training during the use of the tool's features.

Moreover, by developing this customized platform, we can also focus on the integration of these services with other developed applications for the cardiology department of the University Hospital of Salamanca (such as the CARTIER-IA platform [13, 14]), fostering the creation of a robust technological ecosystem [15, 16].

3.2 User Needs

The development of the tool follows a user-center approach. The users have been categorized into two groups. The main user group is physicians. They have knowledge or interest in AI/ML and have data. Moreover, they do not have enough knowledge of programming or using AI algorithms. These users are also characterized by a wide age range and different levels of digital competence, which also influence in the user needs.

The secondary user groups are students and ML experts. Medical students neither do they have knowledge of programming, but they are mostly young, and it is supposed that they have more experience with digital tools. Regarding ML experts, they have knowledge of AI/ML as well as programming skills, however they are not domain experts.

The needs of physicians and medical students are mainly focused on:

- Getting a tool to apply AI/ML in medical datasets without technical expertise and with limited knowledge of ML.
- Visualizing and analyzing medical data.
- Learning about data analysis and its usefulness and benefits in medicine.
- Learning about ML algorithms in a practical way.
- Being able to use a ML application that allows a detailed data visualization and help in interpreting the data.

On the other hand, the needs of ML experts lie in facilitating the integration of algorithms in ML pipelines. Furthermore, we also identify customization as a need. They need to modify the parameters and heuristics of a ML algorithm and learn from the results of their tests.

3.3 Main Scenarios

User scenarios are stories about people and their activities [17]. We have identified five scenarios that cover the goals and questions to be achieved through the tool. The scenarios are mainly focused on supplying the needs of the physicians, the main user group.

First, the pipelines definition. The platform interface will allow the definition of ML pipelines through graphical elements and interactions, to materialize the tasks defined using a programming approach such as those develop using Luigi, the Python module to build complex pipelines of batch jobs. Users will be able to customize the pipeline and access the intermediate results, as well as save configurations for sharing or later use.

The second scenario covers the algorithm training. Users of the platform will be able to train various algorithms by providing some input data. The platform will allow the user to choose the type of algorithm and configure the parameters associated with it. It is proposed that the algorithm and parameter selection process will be guided by a series of heuristics based on existing literature (although it is also proposed that expert users can add or modify these heuristics) depending on the scheme, the problem to be solved and the volume of data entered.

The third scenario is focused on visualization and interpretation of the results. Various metrics and results will be obtained after training is completed depending on the algorithm or ML model used. These results can be ROC curve plots, cut-off points with specificity/sensitivity/etc. values, variable importance, etc. The platform will assist users in the process of interpreting these results through visualizations, annotations, and explanations.

The next identified scenario will support the data checking and visualization. Users will also be able to check and visualize the input data, to explore it. Moreover, the platform will provide feedback regarding the potential problems of the dataset and the feasibility of using different algorithms on the input data.

Finally, the last scenario is related to the use of heuristics. This scenario addresses two objectives, one focused on medical students without ML knowledge and other for ML experts. First, the use of heuristics through a rule-based recommender will be a functionality to use the platform as a didactic tool. The feature will provide a guided process during the definition of pipelines, selection and training of algorithms, interpretation of the results, etc., to provide an educational component in the platform itself for those users who are not experts in AI or programming.

Secondly, the heuristics are also for ML expert users, who may be physicians, data scientists or developers. Particularly, we have identified customization as the main need for ML experts. For this reason, the platform will allow the modification of heuristics. The basic heuristics will be based on existing literature [18, 19], but expert users will be able to modify them through XML documents or a graphical interface, being able to store several versions of the heuristics used in the platform.

4 Concept Validation

The test phase was focused on validating the application to make sure works well for the people who will use it. In particular, the process was focused on and identifying the design requirements using the feedback of the primary and secondary users (physicians and ML experts).

The concept validation includes the development of a high-fidelity prototype [20] using Adobe XD together a remote focus group [21] with final users testing the digital prototype remotely and answering questions related to usability in order to identify problems and discover new requirements.

4.1 The Prototype

The ML platform aims to assist non-expert users in the definition and application of ML pipelines. The platform will also support the integration with other developed tools.

There are three profiles when logging in: regular user, expert, and administrator. The regular user can perform all the tasks that we have mentioned above that represents the main functionality of the application. The expert also has access to the heuristics used by the application and can review and modify them. The administrator can perform all these tasks and manages user accounts and validate them.

The tool is divided into different screens that bring together all the functionality (Fig. 1). We can distingue three groups of screens. First, the homepage which provides information about the platform and the user access. The platform is totally private, so that access to it is protected by user registration and login. In addition, the user sign-in involves a validation process, so users apply for a new account and administrators have to approve it.

The next set of functionality focuses on project management within the platform. Users (both regular and experts) will create ML projects. Each project is mainly a pipeline with one or more inputs and one or more outputs. The platform allows saving the projects in a personal area (Fig. 2), so user can reuse them so many times as needed. Users can also download projects to have a copy or even send them to another user using any other media (email, shared folder, etc.). Users can upload those projects to their accounts

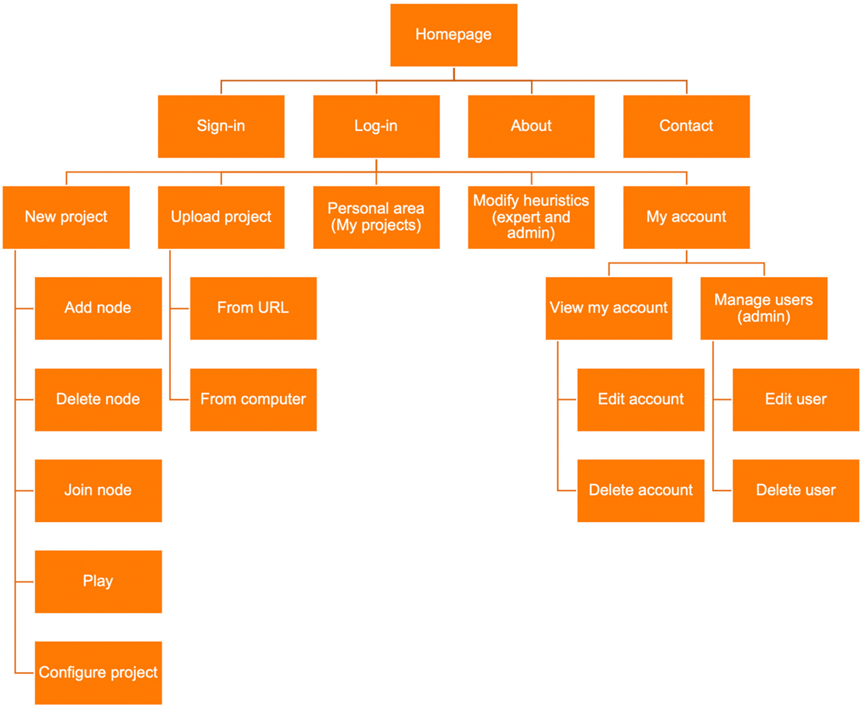


Fig. 1. Main site flows.

through their personal area using an URL to a shared space or a zip file located in the computer.

Furthermore, the personal area also contains a main functionality for expert users and administrators; an interface to manage the heuristics. Experts can modify the parameters and heuristics of a ML algorithm and learn from the results of their tests.

Finally, most of the functionality is in the project editor (Fig. 3). The editor allows creating a new pipeline or editing an existing one. This interface is divided into three main parts. First, a top bar with provides access to the personal area and user account. It also has a button to reset the project.

Second, a toolbar containing all the nodes that can be added to the pipeline grouped by categories: data, data preparation, data processing, visualization, ML algorithms and evaluation. The design is conceived to be able to add new nodes over time. Moreover, the toolbar provides tools for join nodes, configure, execute, and save the project.

The toolbar has two views, a simplified view for expert users or those with more experience in the use of the platform, and a detailed view showing the name of all nodes and categories.

The last and principal part of this interface is the workspace, a blank area to build the pipeline connecting different nodes (Fig. 3). Each node has inputs and outputs; it produces an output, a set of results, which are the input to the next node. A node can be connected with several nodes, for example, a visualization node to get a visual analysis

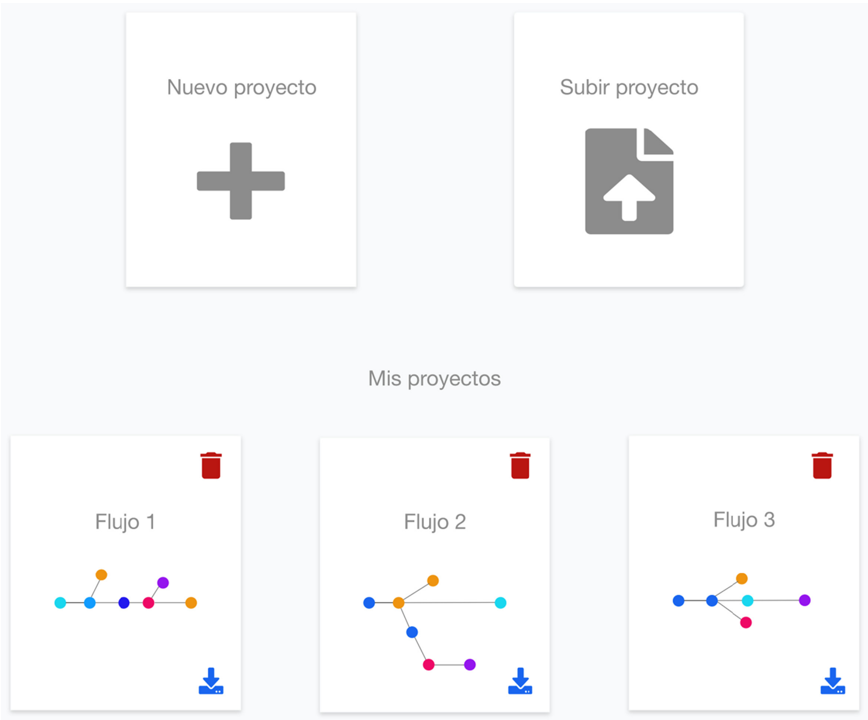


Fig. 2. Saved ML pipelines (projects) in the user's personal area.

of the intermediate results and a node to apply an algorithm to that dataset. The nodes have also different visual marks to provide useful information:

- A warning mark in those cases in which there are missing information, a problem in the execution of the node or a recommendation to improve the configuration of the node.
- A result mark to indicate that this node produces as output a set of intermediate results that can be retrieved when the pipeline is executed.
- A suggestion mark to provide additional information and support regular users without ML knowledge.

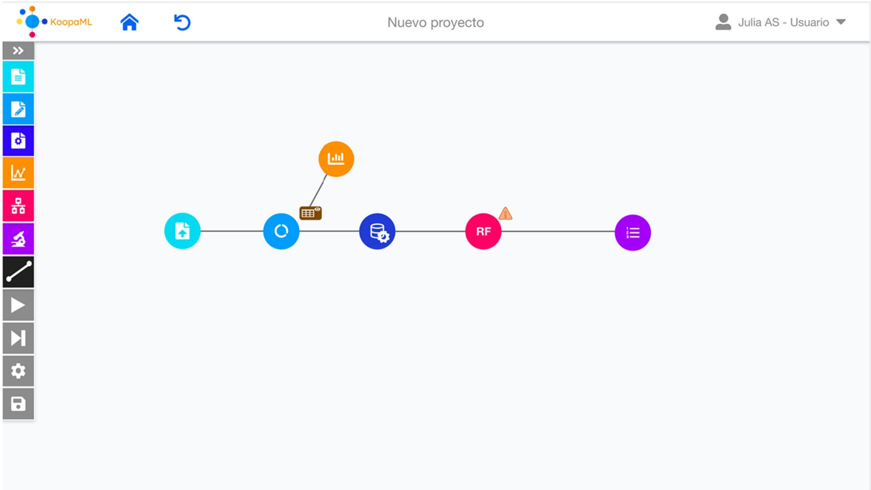


Fig. 3. Pipeline example different tasks involved in the process.

4.2 The Focus Group

The focus group is a qualitative technique that involves a group of participants for discussing on a topic directed by the researcher. Through a focus group, we can learn about users' attitudes, beliefs, desires, and reactions to concepts [22]. It provides valuable assistance to the specification of the interaction and visual design concept of the product under consideration [23].

The objective of the focus group is identifying the design requirements and testing the digital prototype. The focus group was organized fully online to facilitate user participation and to meet the social distancing measures established by the COVID-19 health crisis [24, 25]. We used Zoom as communication tool because not only support grid view, but also allows remote control. During the focus group we gave remote control to some of the participants to perform some tasks within the digital prototype.

The participants in the focus group were the three types of users previously identified:

- CE1: A cardiologist with some knowledge of ML.
- CE2: A cardiologist and researcher involved in ML projects.
- CE3: A cardiologist with a positive attitude towards ML.
- DE1: A physics and data scientist.
- DE2: An industrial engineer and data scientist.
- DE3: A software engineer and data scientist.
- S1: A computer science student working on a ML project related to cardiology.

The focus group was in Spanish, it took 60 min, and it was moderated by three women researchers related to software engineering and HCI. The roundtables were divided into two phases. A first phase in which we introduce the platform and explained the user access and roles; and a second phase in which participants answered different questions related to the main parts of the digital prototype (the personal area and the workspace).

There were common questions for each screen:

- Can you identify briefly what can be done on this screen?
- Can you describe it?
- What is your opinion regarding the visual aesthetics of the screen?

After these questions, the moderators shared some specific questions for each part. For the personal area (Fig. 2):

- How would you delete a project?
- How would you download a project?
- How would you create a project?

Some of the questions also involved the interaction of a volunteer with the digital prototype using the remote-control tool. For the workspace with a new project:

- How would you start a ML pipeline?
- Do you know how to continue the workflow from this first node?
- How would you modify a previous/already created project?

Later, we showed a workspace with a ML pipeline example (Fig. 3):

- Could you describe what you see on this screen?
- Once you have the ML pipeline defined, how would you get the results?
- How would you explore the results of the ML pipeline?

Finally, some general questions:

- Have you been able to see how to navigate between screens?
- What positive aspects of the platform have caught your attention?
- What negative aspects of the platform have caught your attention?
- What tasks would you like to perform with the platform?

5 Results

The analysis of the focus group was focused on identified positive and negative aspects across the different screens. Moreover, we identified a set of suggestions.

5.1 Positive Comments

The ML experts and cardiologist has similar opinions about the workspace. Both indicate that it is an intuitive interface (CE3), notably the first step for uploading data to begin a pipeline (CE2, DE3) and the interaction to add nodes into the pipeline (CE2). Furthermore, CE1 highlight the general order of steps suggested by the category organization in the toolbar. All user groups have underlined the tools to facilitate the use of the platform for users with basic knowledge of ML, such as the tooltips (DE3) and the information about errors and recommendation associated to our pipeline (CE2).

Likewise, domain and ML experts emphasize the design. CE1 indicates that the look & feel is good, and DE3 likes the appearance with few well-organized buttons and with plenty of white space to get started.

Cardiologists pay more attention to pipeline execution. CE1 and CE2 comment the option of executing each node step by step and seeing results from initial stages without waiting to get the final results of the pipeline.

On the other hand, ML experts pay more attention to flexibility. The platform allows creating pipelines with few (or many nodes) connecting them in any order (DE1, DE3). Moreover, DE2 remarks that the platform allows adding different ML models and visualizing many interim results.

Finally, the participants highlighted several general positive aspects of the digital prototype. Both ML experts and cardiologists emphasized the interface design, the navigation, and the simplicity of the tool. Moreover, the ML experts point out that the tool is quite scalable.

5.2 Negative Comments

Regarding the negative aspects, we have identified the following:

- The way the toolbar is expanded is not intuitive. (DE1).
- The "save data" category of nodes causes confusion about their function. (CE1, DE1, DE3).
- Not fully self-explanatory. It is not clear how to proceed to create and join nodes the first time you use the tool if you have not used a similar tool before. Invites you to follow a trial-and-error strategy. (CE1).
- It is not clear which tasks are in progress and which have been completed during the pipeline execution. (DE3).

5.3 Suggestions

Throughout the focus group, the participants suggested several improvements to reduce the identified negative aspects:

- Support new users with the platform indicating which are the first elements of a ML pipeline (DE3) and a set of guidelines or instructions to start the definition of the pipeline (DE3). According to CE1, it would be useful an example (initial simulation) showing how the platform works.
- Improve the toolbar with tooltips (CE1) and using the expanded view as default (CE2).
- Provide a set of short video tutorials on how to use the tool, illustrating basic operation and more concrete things (CE1, DE3).
- Restrict the options for defining the pipeline to beginners and allowing full flexibility for experts (DE3). For example, a default pipeline structure for people with basic ML skills, so they can start a project with two clicks and learn how to do it (DE2).
- Changing the name of the subcategory "save data" to "download or export data" (DE3).
- Add a visual help to know how to join the nodes within the pipeline (CE1).
- Enable execution not only from the toolbar, but also using the right mouse button because this is a common interaction with this type of tool (DE3).
- Include an execution mark to indicate whether the execution of the node within the pipeline was successful or there was a problem (DE1).

6 Conclusions

The health sector produces a huge amount of dataset with useful information to support decision-making processes and complex common activities. However, physicians are not deeply trained in data science although they have the domain knowledge. On the other hand, ML experts are not usually experts in the domain. This work describes the design process of a ML platform to bridging the data science gaps of physicians while automatizing ML pipelines.

Although there are already different tools that allow users to build and execute ML pipelines, the requirements found in the medical context asked for a customized platform with the goal of offering a tool adapted to the necessities of the end users found in this context: physicians with lack of ML and programming skills that are interested in taking advantage from the application of these algorithms.

In addition, the development of a customized tool opens the path for the integration of other already developed tools for the cardiology department at the University Hospital of Salamanca. By implementing communication mechanisms, it is possible to connect different platforms to foster the creation of a technological ecosystem with data science features specifically adapted to the medical context requirements.

The focus group session has provided a considerable amount of data concerning usability functions and their validation by domain experts and ML experts. It has provided information to understand users' current situation and needs, getting the perspective of multiple users discussing the same requirement or functionality. On the other hand, we have collected the reactions to the digital prototype.

The results of the focus group have served as an input to develop a new version of the digital prototype to solve the main detected problems and improve it including some suggestions.

References

1. Litjens, G., et al.: A survey on deep learning in medical image analysis. Med. Image Anal. **42**, 60–88 (2017). https://doi.org/10.1016/j.media.2017.07.005
2. González Izard, S., Sánchez Torres, R., Alonso Plaza, Ó., Juanes Méndez, J.A., García-Peñalvo, F.J.: Nextmed: automatic imaging segmentation, 3D reconstruction, and 3D model visualization platform using augmented and virtual reality. Sensors (Basel) **20**(10), 2962 (2020). https://doi.org/10.3390/s20102962
3. Izard, S.G., Juanes, J.A., García Peñalvo, F.J., Estella, J.M.G., Ledesma, M.J.S., Ruisoto, P.: Virtual reality as an educational and training tool for medicine. J. Med. Syst. **42**(3), 1–5 (2018). https://doi.org/10.1007/s10916-018-0900-2
4. Abadi, M., et al.: TensorFlow: a system for large-scale machine learning. In: 12th USENIX Symposium on Operating Systems Design and Implementation OSDI 16, pp. 265–283. USENIX Association, Savannah (2016)
5. Anil, R., et al.: Apache mahout: machine learning on distributed dataflow systems. J. Mach. Learn. Res. **21**(127), 1–6 (2020)
6. Bisong, E.: Google colaboratory. In: Building Machine Learning and Deep Learning Models on Google Cloud Platform: A Comprehensive Guide for Beginners, pp. 59–64. Apress, Berkeley (2019). https://doi.org/10.1007/978-1-4842-4470-8_7
7. Frank, E., et al.: Weka-a machine learning workbench for data mining. In: Maimon, Oded, Rokach, Lior (eds.) Data Mining and Knowledge Discovery Handbook, pp. 1269–1277. Springer, Boston (2010). https://doi.org/10.1007/978-0-387-09823-4_66
8. Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I.H.: The WEKA data mining software: an update. SIGKDD Explor. Newsl. **11**(1), 10–18 (2009). https://doi.org/10.1145/1656274.1656278
9. Bjaoui, M., Sakly, H., Said, M., Kraiem, N., Bouhlel, M.S.: Depth insight for data scientist with RapidMiner «an innovative tool for AI and big data towards medical applications». Paper presented at the Proceedings of the 2nd International Conference on Digital Tools & Uses Congress, Virtual Event, Tunisia (2020)
10. McCormick, K., Salcedo, J.: IBM SPSS Modeler Essentials: Effective Techniques for Building Powerful Data Mining and Predictive Analytics Solutions. Packt Publishing Ltd., Birmingham (2017)
11. Berthold, M., et al.: KNIME - the Konstanz information miner: version 2.0 and beyond. ACM SIGKDD Explor. Newsl. **11**(1), 26–31 (2009). https://doi.org/10.1145/1656274.1656280
12. Rodríguez-García, J.D., Moreno-León, J., Román-González, M., Robles, G.: Evaluation of an online intervention to teach artificial intelligence with LearningML to 10–16-Year-old students. In: Proceedings of the 52nd ACM Technical Symposium on Computer Science Education. Association for Computing Machinery, pp 177–183 (2021). https://doi.org/10.1145/3408877.3432393
13. García-Peñalvo, F., et al.: Application of artificial intelligence algorithms within the medical context for non-specialized users: the CARTIER-IA platform. Int. J. Interact. Multimed. Artif. Intell. **6**(6), 46 (2021)
14. Vázquez-Ingelmo, A., et al.: Usability study of CARTIER-IA: a platform for medical data and imaging management. In: Zaphiris, P., Ioannou, A. (eds.) HCII 2021. LNCS, vol. 12784, pp. 374–384. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-77889-7_26
15. García-Holgado, A., García-Peñalvo, F.J.: Validation of the learning ecosystem metamodel using transformation rules. Futur. Gener. Comput. Syst. **91**, 300–310 (2019). https://doi.org/10.1016/j.future.2018.09.011
16. García-Holgado, A., García-Peñalvo, F.J.: A metamodel proposal for developing learning ecosystems. In: Zaphiris, P., Ioannou, A. (eds.) LCT 2017. LNCS, vol. 10295, pp. 100–109. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-58509-3_10

17. Carroll, J.: Making Use: Scenario-Based Design of Human-Computer Interactions. The MIT Press, Cambridge (2000)
18. Buitinck, L., et al.: API design for machine learning software: experiences from the scikit-learn project. Paper presented at the ECML PKDD Workshop: Languages for Data Mining and Machine Learning (2013)
19. Scikit-Learn: Choosing the right estimator - Scikit-Learn documentation (2020)
20. Pernice, K.: UX Prototypes: Low Fidelity vs. High Fidelity (2016)
21. Krueger, R.A., Casey, M.A.: Focus Groups: A Practical Guide for Applied Research. Sage publications, Focus group. Usability.gov (2014). https://www.usability.gov/how-to-and-tools/methods/focus-groups.html. Accessed 12 June 2021
22. Kuhn, K.: Problems and benefits of requirements gathering with focus groups: a case study. Int. J. Hum.-Comput. Interact. **12**(3–4), 309–325 (2000). https://doi.org/10.1080/10447318.2000.9669061
23. García-Peñalvo, F.J., Corell, A., Abella-García, V., Grande-de-Prado, M.: Online assessment in higher education in the time of COVID-19. Educ. Knowl. Soc. **21** (2020). https://doi.org/10.14201/eks.23086
24. Fardoun, H., González-González, C.S., Collazos, C.A., Yousef, M.: Exploratory study in iberoamerica on the teaching-learning process and assessment proposal in the pandemic times. Educ. Knowl. Soc. **21** (2020). https://doi.org/10.14201/eks.23437
25. García-Peñalvo, F.J., Corell, A., Rivero-Ortega, R., Rodríguez-Conde, M.J., Rodríguez-García, N.: Impact of the COVID-19 on higher education: an experience-based approach. In: García-Peñalvo, F.J. (ed.) Information Technology Trends for a Global and Interdisciplinary Research Community, pp. 1–18. IGI Global, Hershey (2021)

7.31 Appendix AE. Bringing machine learning closer to non-experts: proposal of a user-friendly machine learning tool in the healthcare domain

Bringing machine learning closer to non-experts: proposal of a user-friendly machine learning tool in the healthcare domain

Bringing machine learning closer to non-experts in the healthcare domain

Andrea Vázquez-Ingelmo

GRIAL Research Group, Department of Computer Science, Research Institute for Educational Sciences. University of Salamanca, andreavazquez@usal.es

Julia Alonso-Sánchez

GRIAL Research Group, Department of Computer Science, Research Institute for Educational Sciences. University of Salamanca, juliaalonso@usal.es

Alicia García-Holgado

GRIAL Research Group, Department of Computer Science, Research Institute for Educational Sciences. University of Salamanca, aliciagh@usal.es

Francisco José García-Peñalvo

GRIAL Research Group, Department of Computer Science, Research Institute for Educational Sciences. University of Salamanca, fgarcia@usal.es

Jesús Sampedro-Gómez

Cardiology Department, Hospital Universitario de Salamanca, SACyL. IBSAL, Facultad de Medicina, Universidad de Salamanca, and CIBERCV (ISCiii), jmsampedro@saludcastillayleon.es

Antonio Sánchez-Puente

Cardiology Department, Hospital Universitario de Salamanca, SACyL. IBSAL, Facultad de Medicina, Universidad de Salamanca, and CIBERCV (ISCiii), asanchezpu@saludcastillayleon.es

Víctor Vicente-Palacios

Philips Healthcare, victor.vicente.palacios@philips.com

P. Ignacio Dorado-Díaz

Cardiology Department, Hospital Universitario de Salamanca, SACyL. IBSAL, Facultad de Medicina, Universidad de Salamanca, and CIBERCV (ISCiii), acho@usal.es

Pedro L. Sánchez

Cardiology Department, Hospital Universitario de Salamanca, SACyL. IBSAL, Facultad de Medicina, Universidad de Salamanca, and CIBERCV (ISCiii), plsanchez@usal.es

Applying Machine Learning to solve or support complex tasks is growing in popularity in a lot of different contexts. One of these contexts is the medical domain. Through Machine Learning, specific problems such as diagnosis, classification, disease detection, segmentation, assessment of organ functions, etc., can be eased by assisting physicians with useful models and their outcomes. However, understanding the application of Machine Learning and Artificial Intelligence algorithms requires expert knowledge and significant data science skills. This work presents a proposal for a user-friendly Machine Learning tool, focusing on providing a good user experience for physicians as well as an educative context for understanding the tasks involved in Machine Learning pipelines, their configuration, and their outputs.

CCS CONCEPTS • Human-centered computing ~ Human computer interaction (HCI) • Applied computing ~ Life and medical sciences ~ Health care information systems • Computing methodologies ~ Machine learning

Additional Keywords and Phrases: Machine Learning, Health domain, User-centered design, Human-Computer Interaction

ACM Reference Format:

First Author's Name, Initials, and Last Name, Second Author's Name, Initials, and Last Name, and Third Author's Name, Initials, and Last Name. 2018. The Title of the Paper: ACM Conference Proceedings Manuscript Submission Template: This is the subtitle of the paper, this document both explains and embodies the submission format for authors using Word. In Woodstock '18: ACM Symposium on Neural Gaze Detection, June 03–05, 2018, Woodstock, NY. ACM, New York, NY, USA, 10 pages. NOTE: This block will be automatically generated when manuscripts are processed after acceptance.

1 INTRODUCTION

The application of Machine Learning (ML) algorithms has become a powerful tool and resource to solve complex tasks in a lot of different fields. ML algorithms provide the means to tackle problems in which a significant quantity of data is involved, as they enable their classification, clustering, identification of patterns and other useful tasks. However, although powerful, the application of ML is not trivial; the selection of a suitable algorithm and the configuration of its parameters needs a deep analysis of the problem to solve, as well as quality data.

One of the domains that is taking advantage of the application of ML in its workflows is the healthcare sector. In this context, a huge number of complex tasks are involved, including diagnosis, classification, disease detection, segmentation, assessment of organ functions, etc. [1-3]. While these tasks are mostly driven by humans, the introduction of ML provides a complementary support to automate and enhance these processes by using the algorithms' outputs with the goal of improving decision-making.

Another characteristic of this domain is the fact that data is continuously being generated, whether because of medical tests, image studies, or medical trials, among other. The availability of data fosters the application of ML algorithms and the development of infrastructure and workflows to make the most out of them. But, as mentioned before, the use of these algorithms is not straightforward, as it is required to rely on data science and programming skills to obtain quality outputs. While physicians have enormous domain knowledge, they could lack these skills, and thus, need to rely on experts to apply ML to their data.

Democratizing knowledge regarding ML algorithms and pipelines could enable non-expert users to support their diagnoses and analyses with AI, enhancing their decision making and obtaining more benefits from them. For all these reasons, in this work we present a ML platform proposal that has the goal of providing intuitive and educational interfaces to ease the interpretation and understanding of ML workflows applied to the medical domain. We followed a user-centered design approach to capture relevant requirements and necessities from potential user profiles that could be involved in this context. This paper provides an overview of the platform and its first development stages.

The remaining text is structured as follows. Section 2 provides an overview of existing tools for automating ML and data science pipelines. Section 3 describes the requirements elicitation process conducted to identify user needs. Section 4 presents the architecture proposal for the platform, while section 5 provides details of the developed prototype. Finally, section 6 discusses the results of the first development stages and concludes this work.

2 RELATED WORK

There are plenty of tools that allow the automation and generation of machine learning pipelines. These tools can fall in three different categories: programming libraries for coding ML applications, applications for experts and non-specialist users and tools that support the understanding of how ML applications work.

A lot of well-known programming libraries fall in this category, such as TensorFlow [4], Apache Mahout [5], and other Python frameworks like PyTorch, Scikit-learn or Keras.io.

In the second category we can find visual environments that support the definition of machine learning pipelines and data science tasks. For example, Weka is a collection of algorithms for data mining tasks. One of its environments enables users to define data streams by connecting nodes that represent data sources, preprocessing tasks, evaluation methodologies, visualizations or algorithms, among other [6, 7].

Orange Data Mining [8] is another powerful tool that, through a user-friendly interface, allows the definition of data mining workflows, with several methodologies, operations and visualizations available. There are more solutions very similar to these mentioned tools, including Rapid Miner [9] which follows the same node-and-link philosophy to specify and define ML workflows. These applications focus on providing robust and complex features through intuitive interfaces and interaction methods.

Finally, there exist applications whose main goal is to provide educational resources to learn and understand how to create ML pipelines, as well as to properly interpret the derived outcomes from ML models. These tools try to explain these complex algorithms through very simple interfaces and without deep diving in technical details. Some examples include Machine Learning for Kids (https://machinelearningforkids.co.uk/) or LearningML (https://web.learningml.org/).

While there are a lot of solutions for applying and automating ML, it is difficult to adapt them to specific contexts in which very particular necessities are involved. For these reasons, our context of application asks for a customized tool that emphasizes the provenance of an educational experience through intuitive interfaces for unskilled physicians that want to start employing ML to support their jobs.

3 REQUIREMENTS ELICITATION

We carried out a requirements elicitation process to identify the main functionalities and scenarios that the system must have. In this regard, we interviewed potential users of the application, including physicians, AI experts and managers.

Through this process, we ended up with the description of the system's main functional blocks: ML pipelines definition, ML algorithms training, results interpretation and visualization, data validation and recommendation heuristics management.

The ML pipelines definition functionality allows users to define ML pipelines through a graphic interface using visual elements and interactions. Users can customize the pipeline and inspect the intermediate results of each pipeline stage, as well as saving the pipeline schema and configuration to share it with colleagues.

The ML algorithms training scenario offers the possibility of training different ML algorithms using the user defined input data. The platform will let users choose the algorithm and configure its parameters. This functionality block is complemented with the proposal of supporting the algorithm selection process with recommendations through a series of heuristics based on the data schema, data volume and problem to solve.

Other crucial requirement of the proposed platform is to include the possibility of visualizing and interpreting the pipeline's results. This process is very important because a bad interpretation of the results could lead to the inefficient use of the pipeline and lose all its potential benefits. For these reasons, the platform is set to provide different metrics once the algorithms training process is done (such as ROC curves, recall, precision, specificity, etc.). In this regard, the platform would assist users in the results interpretation process through visualizations, annotations, and explanations.

On the other hand, the quality of the training process not only depends on the algorithm's configuration, but also on the input data. Therefore, a validation process of the input data must be conducted before training any algorithm. This validation process refers to provide users with information regarding the applicability of the available algorithms to the input data, as well as regarding potential issues related to them (missing values, data volume, data types, etc.).

The last functional scenario is related to recommending suitable algorithms and configuration based on a set of pre-defined heuristics. In this sense, the platform could be employed also as an educational tool to offer a guided process during the pipelines' definition, selection of ML algorithms, outcomes interpretation, etc. These heuristics will be employed to build a rule-based recommender, but the platform will allow the modification and addition of new heuristics to provide more flexibility.

Finally, we also identified two user roles. Identifying user roles is also crucial because it allows the adaptability of the platform depending on the end users' skills, knowledge, and privileges. The following user roles were identified:

- Physicians. These will be the main users of the platform. These users have knowledge regarding the data domain, they are interested in IA and ML, but they don't have enough skills or knowledge to create ML pipelines programmatically.
- IA experts. Experts can also use the platform to build their own pipelines in a straightforward manner, but they will also have privileges to tune the heuristics or modify them based on their own knowledge and experience.

4 ARCHITECTURE PROPOSAL

In this section, we present an architecture proposal to tackle the problem of creating a ML tool that allows expert and non-expert users. The philosophy behind this architecture relies on modular tasks that can be connected to create ML pipelines in a straightforward way.

Although the user interface is a crucial part of the system, it is necessary to analyze the ML and data science domains to identify the main modules of the system to provide a scalable and flexible architecture. In this regard, we followed a domain engineering approach through the previously described requirements elicitation process with potential users as well as through literature reviews.

Following this approach, we identified a set of functional blocks that will interact and collaborate among them to provide support to the goal of the system: to ease and automate the development of ML pipelines. The functional blocks or modules are the following:

- User management module
- Heuristics management module
- Pipelines management module
- Tasks management module

The user management module provides the services related to authentication and sessions. It also supports the management of the system's users' roles described in the previous section. The heuristics management module allows IA experts to modify the recommendation heuristics through a graphic interface. The pipelines management module provides a workspace to create ML pipelines using visual elements. Finally, the tasks management module provides the definition of the operations related to each ML pipeline potential stage.

Figure 1 shows the schematic overview of the platform's architecture with C4 model notation [10]. The platform's architecture is designed to allow flexibility and evolution, due to the evolutive nature of AI and ML algorithms and methods. For this reason, the tasks management module is designed as a loosely coupled module, in which algorithms and operations can be added and modified without impacting the features of the remaining modules.

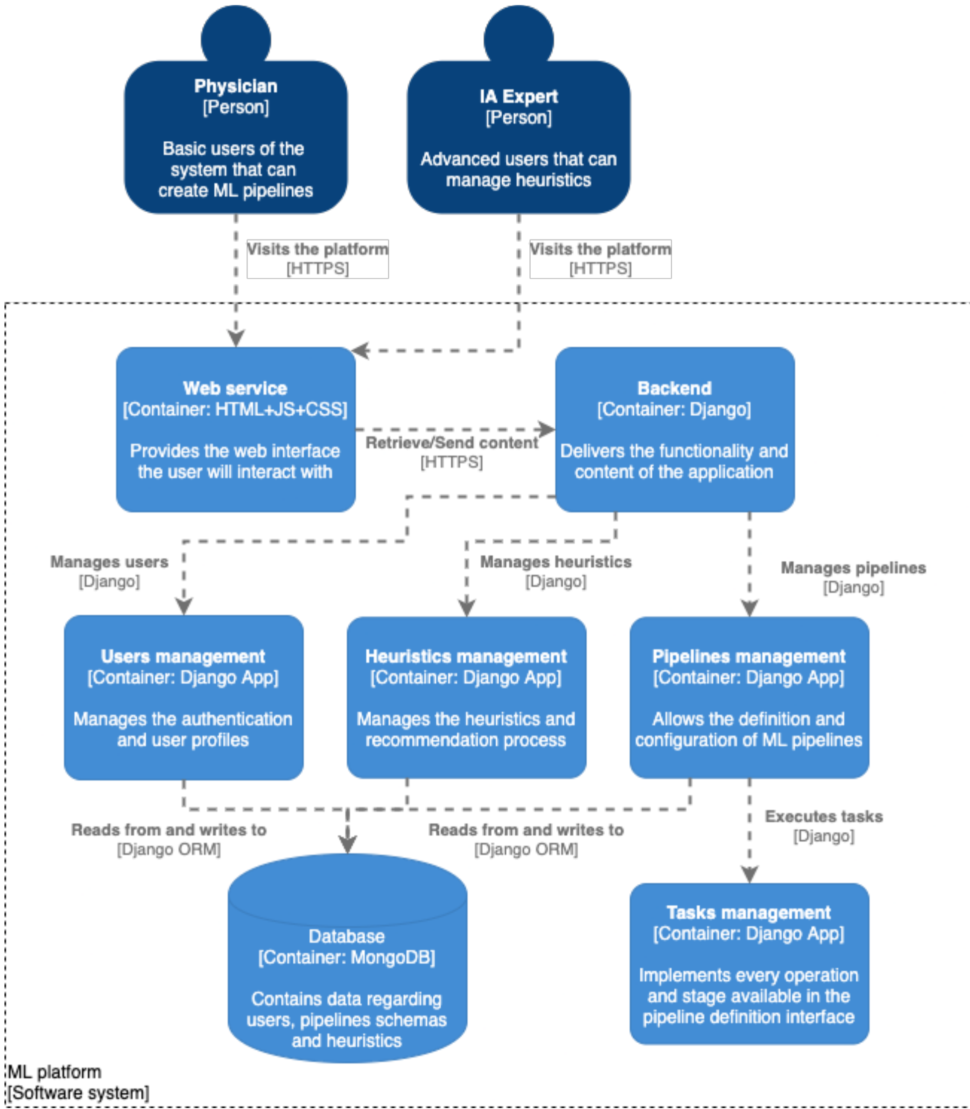


Figure 1: C4 model of the system's architecture.

5 PROTOTYPE

A prototype was developed to identify early design issues as well as new requirements that could went unnoticed during the first requirements elicitation. Following this methodology enables the communication among developers, designers, and end-users.

The prototype was developed with Adobe XD and contains the initial design of the interface for the main functionalities of the platform. Figure 2 shows the initial page or landing page of the platform, in which of the tool's features are introduced. This page also contains the links to sign in and sign up.

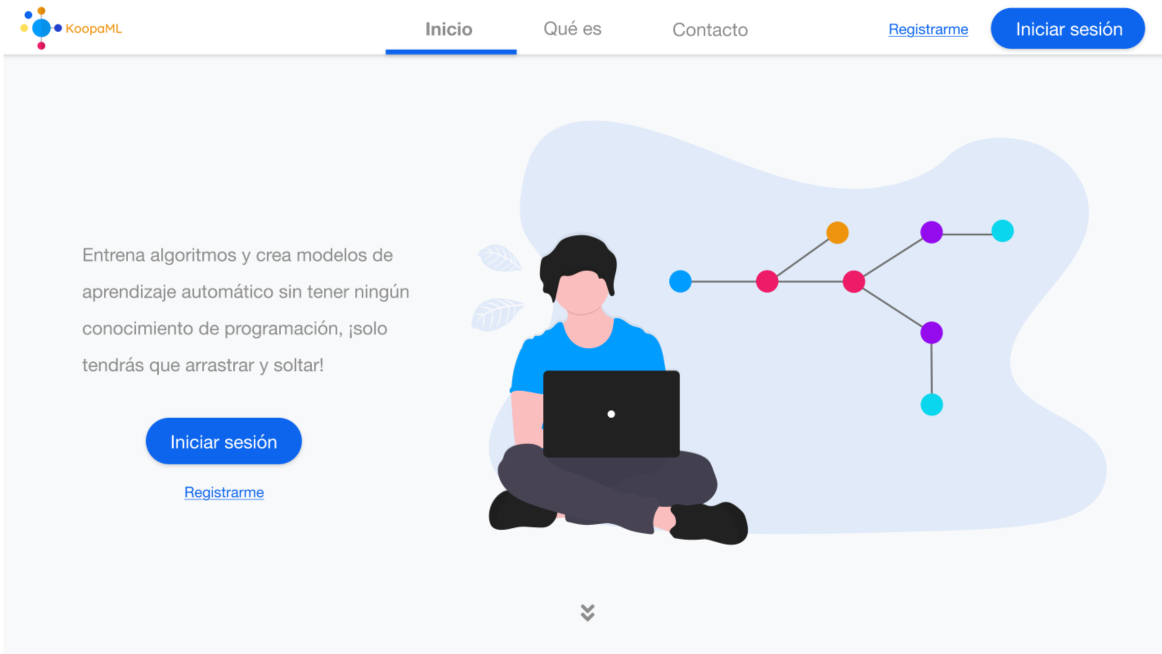


Figure 2: Landing page of the platform

When a user logs in in the platform, his or her personal workspace is retrieved. In this screen a user could start a new project, upload a previously developed project, or continue on-going projects (Figure 3). The “upload project” functionality allows the restoration of previous projects developed with this platform, which also enables users with the possibility of sharing pipelines and importing them into their personal workspace.

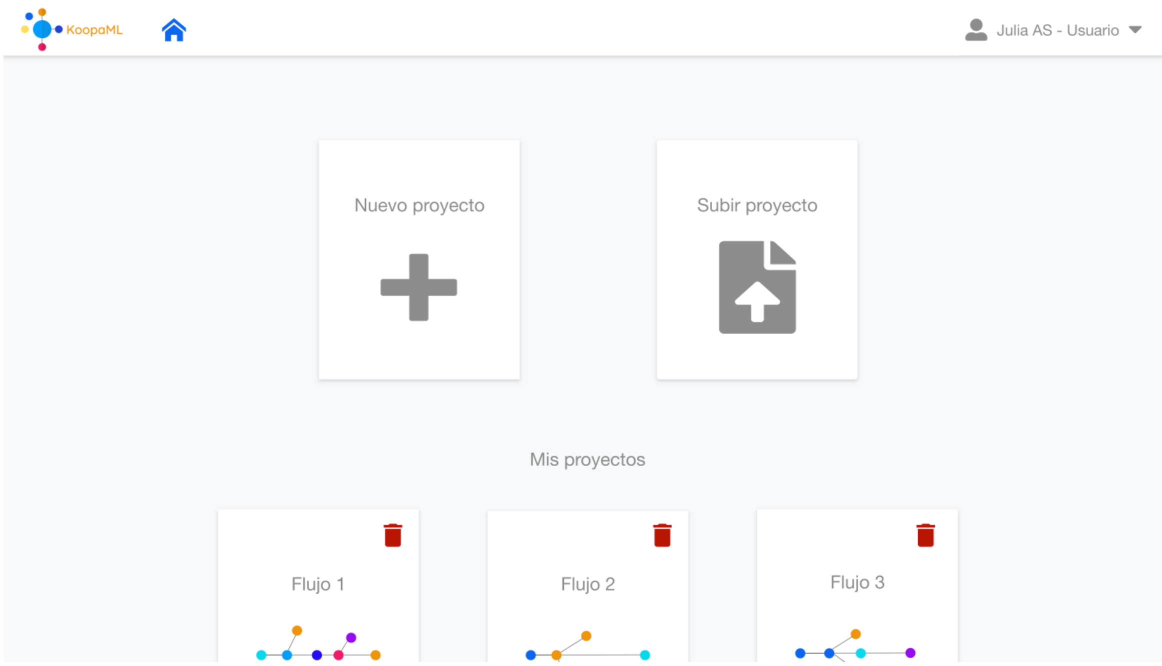


Figure 3: Project creation page.

Starting a new project or selecting an existing one causes the pipeline designer to open. The pipeline designer provides a space to craft pipelines by using a graph layout composed of nodes (data science and AI tasks) and links (input/output connections to the tasks). This approach eases the creation of pipelines through visual elements and drag and drop interactions, making the process more intuitive and straightforward than coding the pipeline programmatically.

In this screen, we also display annotations and recommendations, as well as warnings regarding the misconfiguration of some nodes, with the goal of providing an educational experience during the creation of pipelines.

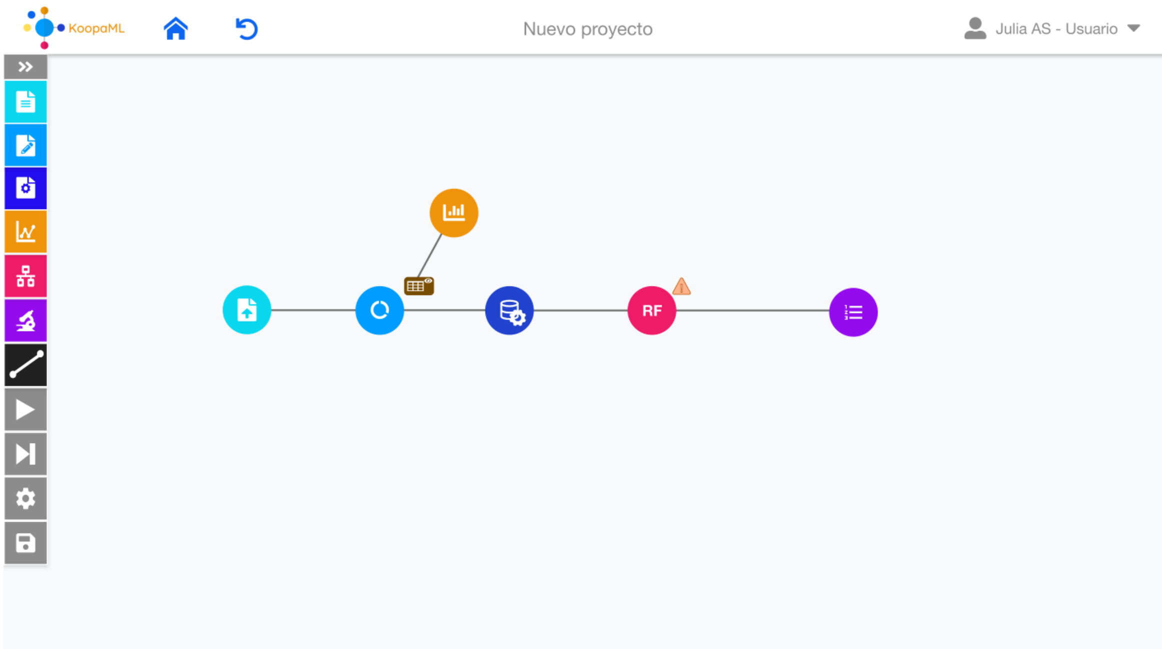


Figure 4: Example pipeline.

Finally, once the pipeline is completed, users can execute it to retrieve its outcomes as well as intermediate results. Retrieving intermediate results was one of the main requirements, as they allow replicability and deeper understanding of the transformations the pipeline is performing on the input data.

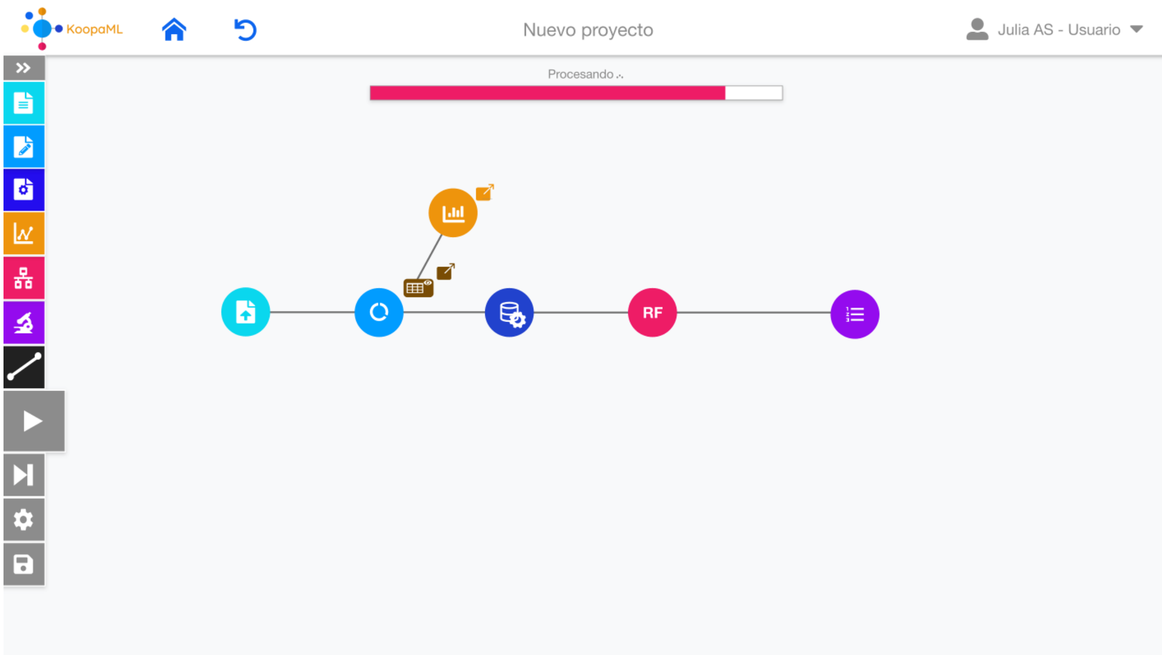


Figure 5: Execution of a pipeline.

6 DISCUSSION AND CONCLUSIONS

We presented the initial stages of the development of a platform for automating and learning the definition of ML pipelines. We followed a user-centered approach for the design and development process, due to the main goal of the system: to ease the application of ML for non-specialized users.

Although there are commercial tools that tackle the automation of these processes, the specific requirements that arise from the medical context asked for a customized platform that aligns with the necessities of specific end users (in this case, physicians with lack of data science skills to actively use these algorithms for their benefit).

Developing a customized tool instead of using an existing tool has its drawbacks, such as a longer development process. However, the benefits derived from the customization of the tool makes this approach suitable for this context. In this regard, we want to conduct more experiments to test the technological acceptance and usability and to continuously improve the platform's features.

On the other hand, another related benefit of the customized tool is the possibility to implement communication mechanisms among other already developed tools for the cardiology department at the University Hospital of Salamanca [11]. Connecting different platforms would foster the creation of a technological ecosystem [12-15] with powerful and transparent data management and data science features adapted to the health sector requirements.

We have conducted a focus group to test and validate the prototype as well as to identify new the requirements. The focus group involved different user profiles, including physicians as well as AI experts related to the health domain. The outcomes of this study are out of the scope of this paper, but the feedback was positive and very useful, providing the necessary information to start the implementation of the tool.

Future research lines will involve the validation of the first version of the platform, as well as in-depth user tests to measure the usability, ease-of-use, and effectiveness of the tool.

ACKNOWLEDGMENTS

This research work has been supported by the Spanish *Ministry of Education and Vocational Training* under an FPU fellowship (FPU17/03276). This work was also supported by national (PI14/00695, PIE14/00066, PI17/00145, DTS19/00098, PI19/00658, PI19/00656 Institute of Health Carlos III, Spanish Ministry of Economy and Competitiveness and co-funded by ERDF/ESF, “Investing in your future”) and community (GRS 2033/A/19, GRS 2030/A/19, GRS 2031/A/19, GRS 2032/A/19, SACYL, Junta Castilla y León) competitive grants.

REFERENCES

[1] G. Litjens *et al.*, "A survey on deep learning in medical image analysis," (in eng), *Med Image Anal,* vol. 42, pp. 60-88, Dec 2017, doi: 10.1016/j.media.2017.07.005.

[2] S. González Izard, R. Sánchez Torres, Ó. Alonso Plaza, J. A. Juanes Méndez, and F. J. García-Peñalvo, "Nextmed: Automatic Imaging Segmentation, 3D Reconstruction, and 3D Model Visualization Platform Using Augmented and Virtual Reality," (in eng), *Sensors (Basel),* vol. 20, no. 10, p. 2962, 2020, doi: 10.3390/s20102962.

[3] S. G. Izard, J. A. Juanes, F. J. García Peñalvo, J. M. G. Estella, M. J. S. Ledesma, and P. Ruisoto, "Virtual Reality as an Educational and Training Tool for Medicine," *Journal of Medical Systems,* vol. 42, no. 3, p. 50, 2018/02/01 2018, doi: 10.1007/s10916-018-0900-2.

[4] M. Abadi *et al.*, "TensorFlow: A System for Large-Scale Machine Learning," in *12th USENIX Symposium on Operating Systems Design and Implementation OSDI 16*. Savannah, GA: USENIX Association, 2016, pp. 265-283.

[5] R. Anil *et al.*, "Apache Mahout: Machine Learning on Distributed Dataflow Systems," *Journal of Machine Learning Research,* vol. 21, no. 127, pp. 1-6, 2020. [Online]. Available: https://jmlr.org/papers/v21/18-800.html.

[6] E. Frank, M. Hall, G. Holmes, R. Kirkby, B. Pfahringer, and I. H. Witten, "Weka-A Machine Learning Workbench for Data Mining," in *Data Mining and Knowledge Discovery Handbook*, O. Maimon and L. Rokach Eds. Boston, MA: Springer, 2009.

[7] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, "The WEKA data mining software: an update," *SIGKDD Explor. Newsl.,* vol. 11, no. 1, pp. 10–18, 2009, doi: 10.1145/1656274.1656278.

[8] J. Demšar and B. Zupan, "Orange: Data mining fruitful and fun-a historical perspective," *Informatica,* vol. 37, no. 1, 2013.

[9] S. Land and S. Fischer, "Rapid miner 5," *Rapid-I GmbH,* 2012.

[10] S. Brown. "The C4 Model for Software Architecture." http://c4model.com/ (accessed.

[11] F. García-Peñalvo *et al.*, "Application of Artificial Intelligence Algorithms Within the Medical Context for Non-Specialized Users: the CARTIER-IA Platform," *International Journal of Interactive Multimedia and Artificial Intelligence,* vol. 6, no. 6, 2021.

[12] A. García-Holgado, "Technological ecosystems in health sector: Rural-Path ecosystem," presented at the International meeting in the area of health and social atention, Zamora, Spain, June 2, 2016, 2016. [Online]. Available: https://goo.gl/Ge7KxR.
[13] A. García-Holgado and F. J. García-Peñalvo, "The evolution of the technological ecosystems: An architectural proposal to enhancing learning processes," in *Proceedings of the First International Conference on Technological Ecosystems for Enhancing Multiculturality (TEEM'13) (Salamanca, Spain, November 14-15, 2013)*, F. J. García-Peñalvo Ed., (ACM International Conference Proceeding Series (ICPS). New York, NY, USA: ACM, 2013, sec. New York, USA, pp. 565-571.
[14] A. García-Holgado, F. J. García-Peñalvo, and M. J. Rodríguez-Conde, "Definition of a technological ecosystem for scientific knowledge management in a PhD Programme," in *Proceedings of the Third International Conference on Technological Ecosystems for Enhancing Multiculturality (TEEM'15) (Porto, Portugal, October 7-9, 2015)*, G. R. Alves and M. C. Felgueiras Eds., (ACM International Conference Proceeding Series (ICPS). New York, NY, USA: ACM, 2015, pp. 695-700.
[15] F. J. García-Peñalvo, "Technological ecosystems for enhancing the interoperability and data flows," *Journal of Information Technology Research,* vol. 11, no. 1, pp. vi-x, 2018.

7.32 Appendix AF. Fostering Decision-Making Processes in Health Ecosystems through Visual Analytics and Machine Learning

Fostering decision-making processes in health ecosystems through visual analytics and machine learning

Francisco José García-Peñalvo, Andrea Vázquez-Ingelmo, Alicia García-Holgado

Grupo de Investigación GRIAL, Departamento de Informática y Automática, Instituto Universitario de Ciencias de la Educación, Universidad de Salamanca, (https://ror.org/02f40zc51), Salamanca, España
{fgarcia, andreavazquez, aliciagh}@usal.es

Abstract. Data-intensive contexts, such as health, use information systems to merge, synthesize, represent, and visualize data by using interfaces to ease decision-making processes. All data management processes play an essential role in exploiting data's strategic value from acquisition to visualization. Technological ecosystems allow the deployment of highly complex services while supporting their evolutionary nature. However, there is a challenge regarding the design of high-level interfaces that adapt to the evolving nature of data. The AVisSA project is focused on tackling the development of an automatic dashboard generation system (meta-dashboard) using Domain Engineering and Artificial Intelligence techniques. This approach makes it possible to obtain dashboards from data flows in technological ecosystems adapted to specific domains. The implementation of the meta-dashboard will make intensive use of user experience testing throughout its development, which will allow the involvement of other actors in the ecosystem as stakeholders (public administration, health managers, etc.). These actors will be able to use the data for decision-making and design improvements in health provision.

Keywords: Domain Engineering; SPL; Meta-modeling; Information Dashboards; Information Systems; Healthcare; Health Domain.

1 Introduction

Information systems have become a critical factor in several contexts. They allow the unification, formatting, processing, and visualization of data through user-friendly interfaces [1]. Moreover, COVID-19 has accelerated the digital transformation in all the business domains such as education [2-6], political decision-making [7, 8], tourism [9, 10], and health systems [11, 12], among others.

Some data-intensive contexts, like the health context [13], present more complexities in their decision-making processes due to the heterogeneity of the data sources and the different formats involved (structured data, echocardiograms, magnetic resonances, etc. [14-17]). For these reasons, it is necessary to tackle these issues with robust methodologies that enable the development and collaboration of complex services.

This work presents a project focused on tackling data and knowledge management issues in complex contexts. The approach leverages meta-modeling, domain engineering, and artificial intelligence to generate information dashboards that assist decision-making processes.

The rest of this paper is organized as follows. Section 2 provides background for the project proposal. Section 3 outlines the objectives of AVisSA. Section 4 details the methodology followed, and section 5 discusses the project's expected outcomes. Finally, section 6 presents the conclusions derived from the work.

2 Background

Since 2012, the GRIAL Research Group [18] has been developing information technology (IT) solutions for the health sector in collaboration with outstanding partners such as Intras Foundation, which provides support for qualitative data analysis and the usability of solutions built around psychiatric care aspects, Salamanca University Hospital, which develops various software systems, and ArSoft, which researches medical image segmentation. Furthermore, the GRIAL Group is affiliated with the Institute of Biomedical Research of Salamanca (IBSAL).

Technological ecosystems are the natural evolution of IT systems. These ecosystems enable the development of highly complex services while maintaining their evolutionary nature. The main goal of the previous R&D project, the DEFINES project (A Digital Ecosystem Framework for an Interoperable NEtwork-based Society, ref.: TIN2016-80172-R), funded by the Spanish National Research Plan (that ended on June 2021), was to develop a framework of a technological environment as support of services for the management of corporate knowledge, dubbed technological ecosystem [19]. In the context of the TE-CUIDA project, this framework was established in a technological ecosystem for the care domain to provide comprehensive and distant assistance for the needs of official and informal care providers (e.g., family members) of dependent older people (TEchnological ecosystem for support for caregivers, ref.: SA061P17) [20].

In terms of technology, the significant outcomes of both projects allow for the solution of the problem through the design and development of a technological ecosystem that is in line with current software architectures. In this approach, we've proposed a generic framework with a solid evolutionary component, whose architecture can be used in various situations and allows for integrating other software tools and developing new services that add value to the ecosystem.

As an example of the strategy's relevance, it's clear that technological ecosystems in the field of health have sparked much interest in recent years, as evidenced by the several European-funded projects that use a similar approach [21].

However, during the previous projects' review of state of the art, a common lack was discovered [22]: despite their immense potential, technological ecosystems in the health sector face challenges when deployed in real contexts, resulting in the majority of concepts never progress beyond pilot testing.

Comparing the health field with other fields in which technological ecosystems are a reality, it was concluded that proposals in the health sector tend to ignore the evolutionary characteristic of the ecosystem. This capacity to incorporate new tools and services means that the data exchanged, and the knowledge generated in this type of solution has a high added value for other potential users of the ecosystem, beyond the users who are the primary objective of the solution. Besides, the challenge remains at the high-level interfaces to create control panels or dashboards that adapt to the evolving and changing nature of data.

In this light, the current project aims to create a fundamental pillar not only for the success of the ecosystem framework proposed in DEFINES and implemented in TE-CUIDA but also for the development of a solid ecosystem architecture that can be extrapolated with guarantees of success when implementing new services and incorporating new actors that contribute to the ecosystem's evolution. This pillar focuses on using information dashboards [23], which are tools for the graphic representation of the leading indicators involved in achieving the objectives within an ecosystem (e.g., degree of success in providing a service) designed to assist in decision-making processes. The goal is to encourage the extraction of knowledge inherent within the information flows between the ecosystem's components and actors by presenting tailored data for each type of user using visual metaphors.

Dashboards are powerful visualization tools capable of combining various heterogeneous data sources from the components of the ecosystem. Still, their development and configuration have a high degree of complexity due to the heterogeneity of the data, the particular needs of the final users of the dashboard itself, and the evolutionary nature of the technological ecosystems that causes the software components to change, be removed or added to meet the changing needs of the ecosystem itself [24, 25]. These characteristics directly impact manually-developed dashboards [26], which need to address the problem from a software perspective and consider the user experience and application (and data) domain.

This project, AVisSA, aims to develop an automatic dashboard generation system for decision-making in Health ecosystems.

3 Objectives

AVisSA will address the development of an automatic dashboard generation system that adapts to data analysis and knowledge management needs in heterogeneous contexts such as the health sector to improve these processes within the health system, impacting decision-making processes.

The automatic generation of these tools will be based on domain engineering and artificial intelligence techniques to obtain customized products with lower development costs.

Identifying the primitive elements of a dashboard and visualization (such as the scales, axes, visual marks, etc.) allows for greater flexibility when setting up an automatic dashboard generation. It determines the influence and usefulness of the various visual elements according to the context.

On the other hand, branches of artificial intelligence as Machine Learning approaches [27-29] can be applied in conjunction with this paradigm to obtain tailored tools. These tools are fed by the particularities of the user, offering the most suitable components to gain insights and for the achievement of their particular objectives [30].

To achieve the main objective of the project, a set of specific goals are proposed:

- O1. Establishing automatic mechanisms for collecting and analyzing knowledge in the technological ecosystem.
- O2. Developing methods for the storage and anonymized (or pseudo-anonymized) treatment of data. Since medical data may be combined with data related to platform usage, best practices will be established to ensure compliance with privacy regulations and legislation.
- O3. Automatic generation of indicators, metrics, and tools that facilitate decision-making for different types of users in the technology ecosystem (e.g., health professionals, managers, and administration members).
- O4. Using artificial intelligence techniques for automatic customization of data visualization forms to suit the application domain and the preferences and characteristics of each user.
- O5. Providing simulation spaces for training or diagnosis processes based on automatically generated dashboards, including machine learning capabilities with the presented data and learning analytics of the educational actions.
- O6. Evaluating dashboard usability and satisfaction (in terms of the capacity of software to be understood, learned, used, and appealed to the user).
- O7. Evaluating the evolutionary and adaptive capacity of the architecture through its implementation in two sufficiently differentiated case studies: the healthcare field and the monitoring of medical tests based on images.

Achieving these objectives will provide the following benefits:

- Providing tools that allow decision-making based on the visualization of heterogeneous data, regardless of the application domain (e.g., medical data, treatment adherence, patient behavioral trends, etc.).
- Attracting new actors to the ecosystem, providing new services, and adding value to the existing solution (public administration, companies in the health sector, etc.).
- Establishing new synergies with other areas of the health sector (e.g., research networks interested in available data).
- Opening new business channels that allow the commercialization of the solution.

4 Methodology

4.1 Coordination

It is essential to define and plan the tasks adequately, to correctly manage the complexity of the project's objectives and its interdisciplinarity and multidisciplinarity. Six activities are proposed in a 48-month window.

Activity 1 is devoted to the project coordination based on the PRINCE2 (PRojects IN Controlled Environments) project management methodology [31].

4.2 Systematic literature review

Activities 2 and 3 represent technological innovation. They will start with in-depth research in the recent advances within the particular context of the automatic generation of dashboards in healthcare environments through SLRs (Systematic Literature Reviews) [32, 33]. Also, the categorization of the data to be extracted, KPIs, and metrics suitable for the different ecosystem actors and users will be addressed in this stage. Based on the best practices and challenges, action-research cycles [34] using SCRUM [35] will be applied. These activities will result in a meta-dashboard implemented and tested in a controlled context, also considering the data life cycle.

4.3 Dashboard generator

For the conceptualization of the dashboard generator, a meta-modeling approach will be used, allowing extraction of the characteristics of the dashboards domain to obtain a dashboard meta-model that contains the abstract and generic features of these tools. Also, the meta-model will focus not only on the dashboards' technical characteristics and functionalities but also on the human factor.

This allows establishing both a practical framework when developing dashboards and a theoretical work on the technical and end user-related elements that should be considered when designing these tools. Among the details related to the end-user, there exist many determining factors for the correct interpretation of the information displayed: level of data domain knowledge, visual literacy, or even potential biases. Although these factors may seem abstract, it is possible to categorize them to obtain user profiles to build visual tools that allow effective visualization and understanding levels.

The Software Product Lines (SPL) paradigm will be used to develop the meta-model. This paradigm offers a development framework of reusable components based on a previous domain analysis (domain engineering phase) to combine them to obtain customized products adapted to different contexts. Two significant benefits can be attained using this paradigm in conjunction with meta-modeling. The first one is the decrease in the development time of these visual tools since they are generated through the composition of previously developed software assets. The second is the flexibility in the generation process.

Specifically, identifying the primitive elements of the dashboards allows improving their scalability. Once the software assets are designed, it is only necessary to assemble them to obtain any dashboard with any number and type of displays. Besides, because the elements identified are generic, the data domain is not a determining factor when generating dashboards. The high-level layer provided by the meta-model allows to abstract more specific aspects of the information to be displayed, allowing this solution to be adapted to any data set. Finally, thanks to this abstraction of information, it is possible to establish connection mechanisms between the dashboard and the elements of any ecosystem [36].

4.4 ·Evaluation

During activity 4, a mixed approach (quantitative-qualitative) was chosen for the evaluations because not all observations are susceptible to quantitative measurement when working in terms of the user experience. Therefore, results analysis requires differentiation not only in terms of quantity (quantitative) but also in terms of quality (qualitative). A priori, neither quantitative nor qualitative research is superior to its counterpart and responds to the same inferential logic: both can be equally systematic and provide similarly helpful information [37]. Moreover, if both types of data are integrated and converge, the validity of the obtained generalizations is reinforced.

Activity 5 is devoted to using the meta-dashboard created in activity 4 to generate simulators for training and diagnosis processes, including machine learning capabilities to help students or professionals analyze the medical data they are visualizing and learning analytics functionalities to have comprehensive information about the simulator usage.

Regarding the experimental phases (activities 4 and 5), this research work will make intensive use of the usability laboratory funded by the network of Networked Infrastructures of Castilla y León (INFRARED - ref. USAL05), of which the GRIAL group is a collaborating member.

The user experience studies will be of an ex-post nature, and within this modality of designs, we will opt to carry out a descriptive study of exploratory nature. On the one hand, combined methods such as Conductor and Thinking Aloud will be employed to test the developed tools' use. Besides, a descriptive study using surveys will be carried out and analyzed using correlation techniques. Thus, the research questions will be answered in descriptive terms and in terms of the relationship between variables and after a systematic information collection. This will ensure the rigor and validity of the obtained information.

For the survey-based research, three phases will be established [38]:

1. Theoretical-conceptual: setting out objectives and research hypotheses
2. Methodological: selecting the sample and the variables under study; formulating and preparing the experimental environment.
3. Conceptual statistics: codification and data analysis to obtain results from which generalizations can be made. Furthermore, integrate the conclusions drawn into the initial theoretical framework.

Finally, and thanks to the specific facilities of the usability laboratory, a logging process of the users' interactions with the system will be carried out. These loggings will be correlated with the results obtained in the other investigation and survey stages. The relevance of the results will be verified, on the one hand, from the point of view of usability by experts of the GRIAL design research team, and on the other hand, from a socio-health point of view by members of the research team with experience in this field.

Activity 6 is oriented to disseminating results and exploiting the developed meta-dashboard.

4.5 Team

The team comprises 16 members (7 in the research team, 9 in the work team – see Table 1). 71.43% of the research team members are males, 28.57% of the members are females. 87.71% of the members of the research team have a Ph.D. degree. 44.44% of the work team members are males, 55.56% of the members are females. 66.67% of the members of the work team have a Ph.D. degree. From the global perspective (16 members, including both research and work teams), 56.25% are males, 43.75% are females, and 75% have a Ph.D. degree. The roles and main profiles are also presented in Table 1. The roles of software engineer, medicine, and neuropsychology mean interdisciplinarity, an expert in educational technologies, and academic researcher suggest multidisciplinary. Finally, manager and data manager roles are the technician profiles for supporting research.

5 Expected results

5.1 Scientific and technical impact

The achievement of the objectives of the present proposal will have a substantial impact on the promotion and generation of frontier knowledge. Specifically, in the following areas:

- Scientific impact and international leadership in the field. The automatic development of dashboards, through a meta-model, allows obtaining visual analysis tools adapted to any domain, facilitating the exploitation of data. Due to the flexibility and adaptability of the dashboards generated using artificial intelligence techniques, the best configuration and design according to the context can be attained. The proposal is at the frontier of state of the art, not only in the health field.
- The innovation of the ideas on which it is based. The systematic studies show no such solution in healthy ecosystems, and meta-dashboards are just developing in other fields. Different techniques have been proposed in the literature to create and adapt dashboards (among them, meta-modeling). Still, most of their applications have not been produced or evaluated in authentic contexts.

- Interdisciplinary and multidisciplinary scientific approach. The approach of the project requires the collaboration of various areas of knowledge. Different experts are involved in domain engineering-based software development, in human-computer interaction for the evaluation of the user experience in the usability lab, and technical staff with expertise in socio-sanitary technologies given the scope of the ecosystem.

More specifically, it is worth noting that the challenges faced by the currently available solutions in the province of dashboards are more accentuated in the field of the health sector. Technological ecosystems and, more specifically, dashboards are a domain that is still in the early stages of development, especially in Spain. With the current health information systems, the static characteristics of performance reporting in the health care sector have resulted in inconsistent, incomparable, time-consuming, and static performance reports that can transparently reflect a round picture of performance and effectively support healthcare managers' decision-making processes. So, the healthcare sector needs interactive performance management tools such as performance dashboards to measure, monitor, and manage performance more effectively.

Software solutions such as dashboards have been commonly related to Business Intelligence (BI) [39]. They allow their users to better manage their data by providing data analysis, information presentation, and integration with other business development environments through metadata management.

To understand the value that this proposal brings by integrating processes and services available in the BI solutions market, it is essential first to identify the challenges that arise within the activities carried out to deploy this type of solution. In general terms, the current BI type solutions, and more specifically, dashboard development, present the following challenges for their deployment:

- Data collection guarantees data quality and consistency among different actors, lines, or processes, including mechanisms to restrict access to information.
- KPIs and Metrics.
- The visual component extrapolates the visualization structure to different processes, lines, or business models.
- To achieve the definition of the dashboard itself, the post-deployment activities are associated with the analysis of the information flows that are established between the ecosystem components and users.

The challenges mentioned above are accentuated when attempting to deploy dashboards in technological ecosystems since they have characteristics that are not present within the areas that traditional BI solutions focus on:

- Its evolutionary component, so that the ecosystem must adapt to changes, both in the context where the ecosystem is deployed and to the changing requirements of users.
- The human component is a fundamental part of the ecosystem at the same level as the software components.

For the above reasons, the success of the current proposal involves a series of improvements over existing solutions:

- The development of a meta-dashboard that can be incorporated into any technological ecosystem so that the definition of the dashboard itself is associated with the analysis of the information flows that are established between the different components of the ecosystem. As well as the interaction of users with the ecosystem technologies.
- The possibility of implementing it in the health environment that has differentiated needs and will improve the knowledge transfer.
- The above will open the market and promote interest in this type of solution from organizations that are part of technological ecosystems or technological solutions in the health sector and have not considered BI a key to their business or activity.
- Provide unified mechanisms to seek the integrity and quality of data from different sources and with varying quality criteria.

5.2 Socio-economic impact

The AVisSA technological ecosystems sought to satisfy the aging and care provision challenge, with the particularity of considering the differential characteristics of the Spanish population consisting of the geographical dispersion. Innovative solutions are needed in active aging, especially to allow for an autonomous life in their home environment for as long as possible. Besides, the implementation of the meta-dashboard will enable the incorporation of other actors as stakeholders (public administration, health managers, etc.) who will be able to access the data generated within the ecosystem for decision making and design improvements in health provision.

The exploitation of the data will make it possible to promote the ecosystem between primary users (doctors, caregivers, and patients) and secondary and tertiary users (managers, public administration, research networks), significantly increasing the viability of the initial proposal. Healthcare dashboards can be used for various purposes, including strategy analysis and execution, performance reviews, performance improvement, data comprehension, and scope opportunity.

On the other hand, from an economic perspective, the dashboard market has evolved from being a resource generally used by large companies to monitor their commercial departments to being used in diverse contexts. Each metric can be broken down, analyzed, and correlated with other information.

The main factor driving the growth of the dashboard software market is the increasing appearance of a large amount of structured and unstructured data that different organizations and companies must manage. One of the fundamental causes of this paradigm shift is that data generated by the interaction of users with companies through digital channels is experiencing a faster growth rate than conventional business data. This, together with the high competition between companies due to an increasingly globalized market, is one of the main factors driving the growth of the global dashboard market. Dashboards allow to show in real-time the trends of the different indicators of interest and offer information that can be used in strategic decisions of businesses and all types of organizations.

6 Conclusions

This work describes the AVisSA project, which will address the development of an automatic dashboard generation system (meta-dashboard) based on the data flow in technological ecosystems.

The meta-dashboard will automatically adapt to the needs of analysis and knowledge management in heterogeneous contexts such as the health sector, to improve these processes within the health system, with special impact on decision-making processes. The implementation of the meta-dashboard will make intensive use of user experience testing throughout its development, which will allow the incorporation of other actors in the ecosystem as new stakeholders (public administration, health managers, etc.).

These actors will be able to make use of the data for decision-making and improve the health provision. The exploitation of the data will make it possible to promote the ecosystem from its primary focus on primary users (caregivers and patients) to secondary and tertiary users (managers, public administration, research networks), significantly increasing the viability of the initial proposal. Thus, these tools that support decision-making will improve both the quality of the services provided and their economic efficiency.

Acknowledgments

This research work has been supported by the Spanish *Ministry of Education and Vocational Training* under an FPU fellowship (FPU17/03276). This research was partially funded by the Spanish Government Ministry of Science and Innovation through the AVisSA project grant number (PID2020-118345RB-I00).

References

1. Álvarez-Arana, A., Villamañe-Gironés, M., Larrañaga-Olagaray, M.: Improving Assessment Using Visual Learning Analytics. Education in the Knowledge Society 21, (2020)
2. García-Morales, V.J., Garrido-Moreno, A., Martín-Rojas, R.: The Transformation of Higher Education After the COVID Disruption: Emerging Challenges in an Online Learning Scenario. Frontiers in Psychology 12, (2021)
3. García-Peñalvo, F.J.: Avoiding the Dark Side of Digital Transformation in Teaching. An Institutional Reference Framework for eLearning in Higher Education. Sustainability 13, (2021)
4. García-Peñalvo, F.J.: Digital Transformation in the Universities: Implications of the COVID-19 Pandemic. Education in the Knowledge Society 22, (2021)

5. García-Peñalvo, F.J., Corell, A., Abella-García, V., Grande-de-Prado, M.: Recommendations for Mandatory Online Assessment in Higher Education During the COVID-19 Pandemic. In: Burgos, D., Tlili, A., Tabacco, A. (eds.) Radical Solutions for Education in a Crisis Context. COVID-19 as an Opportunity for Global Learning, pp. 85-98. Springer Nature, Singapore, Singapore (2021)
6. García-Peñalvo, F.J., Corell, A.: The COVID-19: the enzyme of the digital transformation of teaching or the reflection of a methodological and competence crisis in higher education? Campus Virtuales 9, 83-98 (2020)
7. Hai, T.N., Van, Q.N., Thi Tuyet, M.N., 5, 21-36.: Digital transformation: Opportunities and challenges for leaders in the emerging countries in response to Covid-19 pandemic. Emerging Science Journal 5, (2021)
8. Barrutia, J.M., Echebarria, C.: Effect of the COVID-19 pandemic on public managers' attitudes toward digital transformation. Technology in Society 67, (2021)
9. Almeida, F., Santos, J.D., Monteiro, J.A.: The challenges and opportunities in the digitalization of companies in a post-COVID-19 World. IEEE Engineering Management Review 48, 97-103 (2020)
10. Infante-Moro, A., Infante-Moro, J.C., Gallardo-Pérez, J.: The Employment Possibilities of the Internet of Things in the Hotel Sector and its Training Needs. Education in the Knowledge Society 21, (2020)
11. do Nascimento, M.G., Iorio, G., Thomé, T.G., Medeiros, A.A., Mendonça, F.M., Campos, F.A., David, J., M., Ströele, V., Dantas, M.A.: Covid-19: A digital transformation approach to a public primary healthcare environment. Proceedings of the 2020 IEEE Symposium on Computers and Communications (ISCC) (Rennes, France, 7-10 July 2020). IEEE, USA (2020)
12. Furtner, D., Shinde, S.P., Singh, M., Wong, C.H., Setia, S.: Digital Transformation in Medical Affairs Sparked by the Pandemic: Insights and Learnings from COVID-19 Era and Beyond. Pharmaceutical Medicine In Press, (2021)
13. Rajkomar, A., Dean, J., Kohane, I.: Machine Learning in Medicine. N Engl J Med 380, 1347-1358 (2019)
14. Litjens, G., Kooi, T., Bejnordi, B.E., Setio, A.A.A., Ciompi, F., Ghafoorian, M., van der Laak, J.A.W.M., van Ginneken, B., Sánchez, C.I.: A survey on deep learning in medical image analysis. Med Image Anal 42, 60-88 (2017)
15. González Izard, S., Sánchez Torres, R., Alonso Plaza, Ó., Juanes Méndez, J.A., García-Peñalvo, F.J.: Nextmed: Automatic Imaging Segmentation, 3D Reconstruction, and 3D Model Visualization Platform Using Augmented and Virtual Reality. Sensors (Basel) 20, 2962 (2020)
16. Izard, S.G., Juanes, J.A., García Peñalvo, F.J., Estella, J.M.G., Ledesma, M.J.S., Ruisoto, P.: Virtual Reality as an Educational and Training Tool for Medicine. Journal of Medical Systems 42, 50 (2018)
17. García-Peñalvo, F.J., Vázquez-Ingelmo, A., García-Holgado, A., Sampedro-Gómez, J., Sánchez-Puente, A., Vicente-Palacios, V., Dorado-Díaz, P.I., Sánchez, P.L.: Application of Artificial Intelligence Algorithms Within the Medical Context for Non-Specialized Users: the CARTIER-IA Platform. International Journal of Interactive Multimedia and Artificial Intelligence 6, 46-53 (2021)
18. García-Peñalvo, F.J., Rodríguez-Conde, M.J., Therón, R., García-Holgado, A., Martínez-Abad, F., Benito-Santos, A.: Grupo GRIAL. IE Comunicaciones. Revista Iberoamericana de Informática Educativa 33-48 (2019)

19.García-Holgado, A., García-Peñalvo, F.J.: Validation of the learning ecosystem metamodel using transformation rules. Future Generation Computer Systems 91, 300-310 (2019)
20.García-Peñalvo, F.J., Franco-Martín, M.: Sensor Technologies for Caring People with Disabilities. Sensors 19, (2019)
21.García-Holgado, A., Marcos-Pablos, S., Therón, R., García-Peñalvo, F.J.: Technological ecosystems in the health sector: A mapping study of European research projects. Journal of Medical Systems 43, (2019)
22.Marcos-Pablos, S., García-Peñalvo, F.J.: Technological Ecosystems in Care and Assistance: A Systematic Literature Review. Sensors 19, (2019)
23.Sarikaya, A., Correll, M., Bartram, L., Tory, M., Fisher, D.: What Do We Talk About When We Talk About Dashboards? IEEE Transactions on Visualization Computer Graphics 25, 682-692 (2018)
24.Few, S.: Information dashboard design. Sebastopol. O'Reilly Media, CA, USA (2006)
25.Berinato, S.: Good charts: The HBR guide to making smarter, more persuasive data visualizations. Harvard Business Review Press, Brighton, MA, US (2016)
26.Vartak, M., Huang, S., Siddiqui, T., Madden, S., Parameswaran, A.: Towards visualization recommendation systems. ACM SIGMOD Record 45, 34-39 (2017)
27.Hu, K., Bakker, M.A., Li, S., Kraska, T., Hidalgo, C.: VizML: A Machine Learning Approach to Visualization Recommendation. Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems (Glasgow, Scotland, UK, May, 2019). ACM, New York, NY, USA (2019)
28.Dibia, V., Demiralp, Ç.: Data2Vis: Automatic generation of data visualizations using sequence to sequence recurrent neural networks. IEEE computer graphics and applications 39, 33-46 (2019)
29.Vázquez-Ingelmo, A., García-Holgado, A., García-Peñalvo, F.J., Therón, R.: Proof-of-concept of an information visualization classification approach based on their fine-grained features. Expert Systems n/a, e12872 (In Press)
30.Vázquez-Ingelmo, A., García-Peñalvo, F.J., Therón, R., Amo-Filvà, D., Fonseca-Escudero, D.: Connecting domain-specific features to source code: Towards the automatization of dashboard generation. Cluster Computing. The Journal of Networks, Software Tools and Applications 23, 1803-1816 (2020)
31.Office of Government Commerce: An introduction to PRINCE2: Managing and directing successful projects. The Stationery Office, Belfast, Ireland (2009)
32.Kitchenham, B., Brereton, O.P., Budgen, D., Turner, M., Bailey, J., Linkman, S.: Systematic literature reviews in software engineering – A systematic literature review. Information and Software Technology 51, 7-15 (2009)
33.García-Holgado, A., Marcos-Pablos, S., García-Peñalvo, F.J.: Guidelines for performing Systematic Research Projects Reviews. International Journal of Interactive Multimedia and Artificial Intelligence 6, 136-144 (2020)
34.Baskerville, R.L.: Investigating information systems with action research. Commun. AIS 2, (1999)
35.Schwaber, K., Beedle, M.: Agile software development with Scrum. Prentice Hall PTR, Upper Saddle River, NJ, USA (2001)

36.Vázquez-Ingelmo, A., García-Peñalvo, F.J., Therón, R.: Taking advantage of the software product line paradigm to generate customized user interfaces for decision-making processes: a case study on university employability. PeerJ Computer Science 5, e203 (2019)
37.King, G., Keohane, R.O., Verba, S.: Designing social inquiry: Scientific inference in qualitative research. Princeton university press, USA (1994)
38.Buendía, L., Colás, P., Hernández, F.: Métodos de investigación en psicopedagogía. McGraw-Hill, Madrid (1998)
39.Marcos-Pablos, S., García-Holgado, A., García-Peñalvo, F.J.: Modelling the business structure of a digital health ecosystem. In: Conde-González, M.Á., Rodríguez-Sedano, F.J., Fernández-Llamas, C., García-Peñalvo, F.J. (eds.) TEEM'19 Proceedings of the Seventh International Conference on Technological Ecosystems for Enhancing Multiculturality (Leon, Spain, October 16th-18th, 2019), pp. 838-845. ACM, New York, NY, USA (2019)

7.33 Appendix AG. Content-validation questionnaire of a meta-model to ease the learning of data visualization concepts

Content-validation questionnaire of a meta-model to ease the learning of data visualization concepts

Andrea Vázquez-Ingelmo[1], Alicia García-Holgado[1], Francisco José García-Peñalvo[1], Roberto Therón[1,2] and Ricardo Colomo-Palacios[3]

[1] GRIAL Research Group, Computer Science Department, Universidad de Salamanca, (https://ror.org/02f40zc51), Salamanca, Spain
[2] VisUSAL, Computer Science Department, Universidad de Salamanca, (https://ror.org/02f40zc51), Salamanca, Spain
{andreavazquez,aliciagh,fgarcia,theron}@usal.es

[2] Department of Computer Sciences, Østfold University College, Norway
ricardo.colomo-palacios@hiof.no

Abstract. Data visualizations and dashboards are powerful means to convey information to large audiences. However, the design and understanding of these tools are not straightforward because several factors are involved. It is essential to rely on theoretical frameworks to design and implement data visualizations for these reasons. In this context, we propose a meta-model to identify and arrange the main characteristics and elements of data visualizations and dashboards. The proposed meta-model provides a powerful artifact to generate information visualizations and dashboards automatically, but also a learning resource to understand how data visualizations elements interact and influence each other. However, it is necessary to validate this artifact to ensure its quality and usefulness. In this paper, we present a work-in-progress or a quality assessment and content validation of the meta-model to seek weaknesses and tackle them in subsequent iterations.

Keywords: Information Dashboards, Data Visualization, Learning Resource, Content Validation.

1 Introduction

Visual explanations are everywhere: they convey complex information, raise attention over target topics, improve the understandability of particular domains, etc. [1]. They can take the form of infographics, simple graphs, or even elaborated information visualizations.

These tools are also powerful because they let users visually perceive information to generate knowledge. However, the complexity involved in this domain can hamper the understanding of the displayed data.

Data visualizations are composed of different visual elements, including shapes, visual encodings (i.e., visual characteristics like color, size, position, etc.), and visual aids (i.e., legends, axes, etc.), but also abstract concepts like scales, data domain, data operations, among others.

If these concepts are not fully understood, they can lead to improper designs (on the developer's side) and to wrong conclusions (on the audience's side).

It is essential to consider all the concepts and factors involved in the data visualization domain to provide adequately designed and reliable visualization methods.

These factors can be considered through domain experts (i.e., information visualization experts) who also know the visualization's data domain and can provide a well-designed product through its expertise.

However, it isn't easy to have this expertise or domain knowledge levels for every practitioner who uses information visualizations to convey information. For all these reasons, it is crucial to deeply understand all the elements that compose data visualizations and how they relate to and influence each other.

The abstraction of these elements and their relationships can provide a framework to improve the knowledge about the most primitive aspects of data visualizations, no matter if they are tangible (understood as the elements that are directly displayed through data visualizations, such as shapes, encodings, layouts, etc.) or conceptual (data domains, data transformations, user characteristics, etc.).

To achieve this, paradigms like the model-driven architecture [2] provides guidelines to develop a meta-model that captures the most relevant factors involved in designing data visualizations and information dashboards. We have presented a dashboard meta-model in previous works and improved it through domain engineering. The prior version of the meta-model will be described in the methodology section.

One of the main benefits of relying on a meta-model is that it can be used as a conceptual map and educational tool to guide the design of data visualizations, but also as an artifact to automatically generate analytical dashboards in different domains (health [3], education [4, 5], employment [6], etc.). However, it is crucial to validate the content of this resource to check if the represented entities are relevant, coherent, and understandable.

In this paper, we provide a proposal to validate the content of the dashboard. By validating the content of the meta-model, it is possible to identify potential limitations and drawbacks of the dashboard domain representation and address them before using this artifact to instantiate real-world dashboards.

The rest of this paper is organized as follows. Section 2 describes previous works related to meta-model validations. Section 3 describes the methods employed to carry out the meta-model, the automatic generation of dashboards, and the experts' validation questionnaire. Finally, section 4 discusses the results, and section 5 offers the conclusions derived from this work.

2 Background

A meta-model is considered of quality if it contains the required elements to instantiate a model that adequately represents the elements in the domain and is technically built using solid principles. The first concern is related to meta-model validation, which is necessary to ensure that we are building the right meta-model [7]. It is important to validate meta-models with respect to the specifications of the domain or with the help of domain experts who can provide meaningful examples of correct and incorrect uses of the Domain-Specific Modelling Languages (DSML) [8]. This validation should be combined with a set of quality attributes.

There are few quality models or frameworks to measure the quality of a meta-model [9, 10]. On the one hand, Basciani, et al. [11] identify four quality assessment approaches focused on meta-models. QM4MM proposes a refinement of the ISO/IEC 9126 quality model [9], it identifies a set of quality attributes organized in six dimensions – functionality, reliability, usability, efficiency, maintainability, and portability. Ma, et al. [10] describe a quality model that measures five quality attributes – syntactic quality, semantic quality, pragmatic quality, capability quality, evolvability quality – and nine metrics based on object-oriented metrics to measure each quality property. Basciani, et al. [11] designed a questionnaire to evaluate the perceived overall quality of the meta-model based on the quality attributes previously defined by [9]. López-Fernández et al. propose 30 common quality criteria for meta-models and an integrated tool for checking them during the meta-model construction [8, 12].

On the other hand, Marín, et al. [13] proposes a metamodel for defect detection in Model-Driven Development oriented conceptual models and a set of rules for the detection of defects in the model.

Regarding validation techniques, there is no specific literature focused on validation methods for meta-models. According to the systematic literature review of meta-models for software quality and its evaluation [14], the selected papers applied different validation methods: designing case studies, toy experiments, peer reviews by experts, and a pilot project application. Moreover, the authors highlight the importance of using a real-world case for the validation of any meta-model to demonstrate its usefulness. These methods can be combined, for example, in [15] authors apply empirical validation and involve experts in a peer-reviewed approach. Furthermore, in [16] authors use experts validation joined to a case study.

3 Design of the questionnaire

Although meta-models are prone to extensions and modifications, the main entities and relationships must be identified to ensure the proper development of dashboards, in this case.

The dashboard meta-model has been developed using a domain engineering [17, 18] and an example-driven approach [19], in which different dashboards and data visualizations were analyzed to extract main and standard features among their elements. This methodology allows the identification of abstract qualities that can be arranged into a meta-model to obtain fully functional dashboard instantiations.

We developed the first version of an automatic dashboard generator based on the meta-model to test if it supports the generation of real-world dashboard and visualization examples. The code generator inputs a set of parameters that account for the elements and attributes of the meta-model, and the result is the source code of a dashboard according to the provided configuration.

The approach taken to automatically generate the source code is based on the software product line (SPL) paradigm [20, 21] and we developed different HTML and JavaScript code templates [22] to materialize the variability points of the product line [23].

During the development and testing of the dashboard generator, the meta-model was subject to modifications because this process raised new dimensions and relationships that need to be accounted for during the design and development of these tools. Figure 1 presents the current version of the dashboard meta-model.

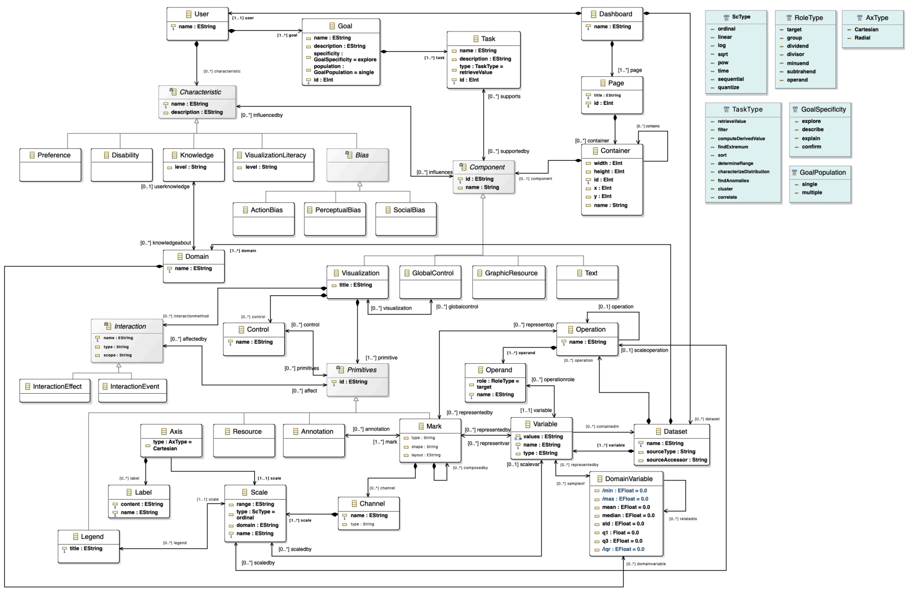


Fig. 1. The current version of the dashboard meta-model. A high-resolution version of the image can be consulted at https://doi.org/10.5281/zenodo.5788981.

We designed a questionnaire to carry out a validation of the dashboard meta-model's new version for these reasons. This validation aims to check if the dashboard meta-model's sections are clear, coherent, and relevant by applying expert judgment [24].

We created an online questionnaire in which six different sections of the meta-model (dashboard layout, user characteristics, goals and tasks, user and dashboard relationships, data visualization primitives, and data domain and operations), in addition to the whole meta-model, were scored in terms of the mentioned dimensions using a 1-4 scale, where one implies that the section does not meet the criterion, and four that it highly meets the criterion. Table 1 shows the rubric used to score the different dimensions based on previous works on content validation by experts [25].

Table 1. The employed rubric to score the meta-model's content.

Criterion	Does not meet the criterion	Low	Medium	High
Clarity	The meta-model section is not clear	The meta-model section requires modifications on the relationships or entities	The meta-model section requires specific modifications on some relationships or entities	The meta-model section is clear, and the relationships and entities capture the dashboards' domain characteristics
Coherence	The meta-model section has no logic relationship with the dashboards' domain	The meta-model section has a slight logic relationship with the dashboards' domain	The meta-model section has a moderate logic relationship with the dashboards' domain	The meta-model section is completely related with the dashboards' domain
Relevance	The meta-model section could be removed without affecting the meta-model as a whole	The meta-model section has some relevance, but it is not especially relevant to define dashboards	The meta-model section is relatively important to define dashboards	The meta-model section is crucial to define dashboards

We also added a brief explanation of the meta-model section and a "Yes/No" question to test if the representation meets the intended goal of the section. Finally, we included an open text field to collect any qualitative feedback of justification that experts might have. The questionnaire was configured in a customized installation of LimeSurvey and sent by e-mail to different domain experts from both the software engineering and data visualization fields.

4 Discussion and conclusions

This paper presents our work-in-progress for validating a meta-model framed in the data visualization and dashboard domain. The meta-model is set to be used as a foundational learning resource for users to understand the primitive elements of data visualizations, which is, in turn, important to reach proper insights.

The meta-model has been subject to continuous modifications following the domain engineering approach, including not only tangible and visual elements but also abstract concepts related to the user and data domain. The resulting outcome is a complex meta-model (Figure 1) that tries to enclose every primitive element involved in the data visualization domain.

Due to the meta-model's inherent complexity and our intention to use it as a learning resource, we proposed a content validation of this artifact before integrating it into real-world processes.

The validation questionnaire has been designed to measure the coherence, relevance, and clarity of the dashboard meta-model. It was necessary to divide the meta-model into different sections to ease the analysis. This questionnaire has been sent to experts in the data visualization and software engineering domain with the goal of obtaining different perspectives about the content of the meta-model.

The measured dimensions will provide crucial information to improve the meta-model, which will be translated into a better learning resource that provides the most relevant elements in data visualizations in an understandable, clear, and coherent manner.

Future research will involve the analysis of the questionnaire responses as well as the improvements related to the potential issues that might arise from the expert validation.

Once the validation has been carried out, we plan to use the meta-model as a learning resource by implementing an educational tool focused on improving the understandability of the elements involved in data visualizations and information dashboards.

References

[1] A. Álvarez-Arana, M. Villamañe-Gironés, and M. Larrañaga-Olagaray, "Improving Assessment Using Visual Learning Analytics," *Education in the Knowledge Society,* vol. 21, Article 9, 2020, doi: https://doi.org/10.14201/eks.21554.

[2] A. G. Kleppe, J. Warmer, and W. Bast, "MDA Explained. The Model Driven Architecture: Practice and Promise," ed: Addison-Wesley Longman Publishing Co., Inc., Boston, MA, 2003.

[3] A. Vázquez-Ingelmo, A. García-Holgado, F. J. García-Peñalvo, and R. Therón, "A Meta-Model Integration for Supporting Knowledge Discovery in Specific Domains: A Case Study in Healthcare," *Sensors,* vol. 20, no. 15, p. 4072, 2020. [Online]. Available: https://www.mdpi.com/1424-8220/20/15/4072.

[4] A. Vázquez-Ingelmo and R. Therón, "Beneficios de la aplicación del paradigma de líneas de productos software para generar dashboards en contextos educativos," *RIED. Revista Iberoamericana de Educación a Distancia,* vol. 23, no. 2, pp. 169-185, 07/01 2020, doi: 10.5944/ried.23.2.26389.

[5] A. Vázquez-Ingelmo, F. J. García-Peñalvo, and R. Therón, "Generating Dashboards Using Fine-Grained Components: A Case Study for a PhD Programme," in *Learning and Collaboration Technologies. Designing, Developing and Deploying Learning Experiences*, Cham, P. Zaphiris and A. Ioannou, Eds., 2020// 2020: Springer International Publishing, pp. 303-314, doi: 10.1007/978-3-030-50513-4_23.

[6] A. Vázquez-Ingelmo, F. J. García-Peñalvo, and R. Therón, "Taking advantage of the software product line paradigm to generate customized user interfaces for decision-making processes: a case study on university employability," *PeerJ Computer Science,* vol. 5, e203, 2019, doi: 10.7717/peerj-cs.203.

[7] B. W. Boehm, "Verifying and validating software requirements and design specifications," *IEEE Software,* vol. 1, no. 1, pp. 75-88, 1984.

[8] J. J. López-Fernández, E. Guerra, and J. de Lara, "Combining unit and specification-based testing for meta-model validation and verification," *Information Systems,* vol. 62, pp. 104-135, 2016/12/01/ 2016, doi: https://doi.org/10.1016/j.is.2016.06.008.

[9] M. F. Bertoa and A. Vallecillo, "Quality attributes for software metamodels," presented at the Quantitative Approaches in Object-Oriented Software Engineering (QAOOSE 2010), Málaga, Spain, 2010.

[10] Z. Ma, X. He, and C. Liu, "Assessing the quality of metamodels," *Frontiers of Computer Science,* vol. 7, no. 4, pp. 558-570, 2013/08/01 2013, doi: 10.1007/s11704-013-1151-5.

[11] F. Basciani, J. Di Rocco, D. Di Ruscio, L. Iovino, and A. Pierantonio, "A tool-supported approach for assessing the quality of modeling artifacts," *Journal of Computer Languages,* vol. 51, pp. 173-192, 2019/04/01/ 2019, doi: https://doi.org/10.1016/j.cola.2019.02.003.

[12] J. J. López-Fernández, E. Guerra, and J. de Lara, "Assessing the Quality of Meta-models," in *MoDeVVa*, vol. 1235, F. Boulanger, M. Famelis, and D. Ratiu Eds., no. CEUR Workshop Proceedings Series). Valencia, Spain: CEUR-WS.org, 2014, pp. 3-22.

[13] B. Marín, G. Giachetti, O. Pastor, and A. Abran, "A Quality Model for Conceptual Models of MDD Environments," *Advances in Software Engineering,* vol. 2010, p. 307391, 2010/07/27 2010, doi: 10.1155/2010/307391.

[14] N. Yılmaz and A. K. Tarhan, "Meta-models for Software Quality and Its Evaluation: A Systematic Literature Review," in *Joint Proceedings of the 30th International Workshop on Software Measurement and the 15th International Conference on Software Process and Product Measurement (IWSM Mensura 2020) (Mexico City, Mexico, October 29-30, 2020)*, vol. 2725, A. Abran and Ö. Özcan-Top Eds., no. CEUR Workshop Proceedings Series): CEUR-WS.org, 2020.

[15] S. Wagner *et al.*, "Operationalised product quality models and assessment: The Quamoco approach," *Information and Software Technology,* vol. 62, pp. 101-123, 2015/06/01/ 2015, doi: https://doi.org/10.1016/j.infsof.2015.02.009.

[16] P. Mohagheghi and V. Dehlen, "A metamodel for specifying quality models in model-driven engineering," in *Proceedings of the Nordic Workshop on Model Driven Engineering*, 2008, pp. 51-65.

[17] G. F. Arango, "Domain engineering for software reuse," 1988.

[18] K. C. Kang, S. G. Cohen, J. A. Hess, W. E. Novak, and A. S. Peterson, "Feature-oriented domain analysis (FODA) feasibility study," Carnegie-Mellon University, Software Engineering Institute, Pittsburgh, PA, USA, 1990.

[19] J. J. López-Fernández, J. S. Cuadrado, E. Guerra, and J. De Lara, "Example-driven meta-model development," *Software & Systems Modeling,* vol. 14, no. 4, pp. 1323-1347, 2015.

[20] P. Clements and L. Northrop, *Software product lines*. Addison-Wesley, 2002.

[21] K. Pohl, G. Böckle, and F. J. Van der Linden, *Software Product Line Engineering: Foundations, Principles and Techniques*. New York, NY, USA: Springer-Verlag New York, Inc., 2005.

[22] A. Vázquez-Ingelmo, F. J. García-Peñalvo, and R. Therón, "Addressing Fine-Grained Variability in User-Centered Software Product Lines: A Case Study on Dashboards," in *World Conference on Information Systems and Technologies*, 2019: Springer, pp. 855-864.

[23] A. Vázquez-Ingelmo, F. J. García-Peñalvo, and R. Therón, "Taking advantage of the software product line paradigm to generate customized user interfaces for decision-making processes: a case study on university employability," *PeerJ Computer Science,* vol. 5, p. e203, 2019/07/01 2019, doi: 10.7717/peerj-cs.203.

[24] R. Skjong and B. H. Wentworth, "Expert judgment and risk perception," in *Proceedings of the Eleventh (2001) International Offshore and Polar Engineering Conference (Stavanger, Norway, June 17-22, 2001)*: International Society of Offshore and Polar Engineers, 2001.

[25] J. Escobar-Pérez and A. Martínez, "Validez de contenido y juicio de expertos: Una aproximación a su utilización," *Avances en Medición,* vol. 6, pp. 27-36, 01/01 2008.

7.34 Appendix AH. A proposal to measure the understanding of data visualization elements in visual analytics applications

A proposal to measure the understanding of data visualization elements in visual analytics applications

Andrea Vázquez-Ingelmo[1], Francisco José García-Peñalvo[1], Roberto Therón[1,2], Vetria Byrd[3], and Jorge D. Camba[3]

[1] GRIAL Research Group, Computer Science Department, Universidad de Salamanca, (https://ror.org/02f40zc51), Salamanca, Spain
[2] VisUSAL, Computer Science Department, Universidad de Salamanca, (https://ror.org/02f40zc51), Salamanca, Spain
{andreavazquez,aliciagh,fgarcia,theron}@usal.es

[3] Computer Graphics Technology, Purdue University, West Lafayette, IN, USA
{vlbyrd,jdorribo}@purdue.edu

Abstract. Data visualizations and information dashboards are useful but complex tools. They must be fully understood to draw proper insights and to avoid misleading conclusions. However, several elements and factors are involved in this domain, which makes it difficult to learn. In previous works, we proposed a meta-model to capture the primitive elements that compose visualizations and dashboards. This meta-model has served as a framework for conducting data visualization research, but also to develop a graphical tool for generating data visualizations and dashboards. This tool (namely MetaViz) enables users to create data visualizations through fine-grained components based on the entities represented in the meta-model. The main goal of the system is to provide a learning experience in which users can freely add and configure elements to understand how they influence the final display. This work describes work in progress to validate the pedagogical value of MetaViz in terms of the understanding of data visualization concepts.

Keywords: Data visualization, Information Dashboards, Understandability, Instrument.

1 Introduction

Conveying information effectively has become a challenge over the years; the increase of data in terms of quantity and complexity has hampered their analysis and communication to different audiences. However, along with the increase in quantity and complexity, there is also an increase in relevance of data within informed decision-making processes [1]. Using a data-driven philosophy to make decisions allows better communication, measurability, accountability and more objective approaches to tackle new problems [2].

Visual tools such as data visualizations and information dashboards are being widely used to address the extraction of knowledge from raw data [3]. Although these mechanisms provide means to convey data easier [4-7], audiences must understand the employed visual metaphors [8]. Many factors are involved at this point: potential biases [9], beliefs [10-12], and even -purposely or not- misleading designs [13, 14].

Regarding the latter, several platforms have been developed to facilitate the process of designing and implementing data visualizations. Systems like Tableau (https://www.tableau.com/), Microsoft Excel (https://www.microsoft.com/microsoft-365/excel), Power BI (https://powerbi.microsoft.com/), etc., provide graphical interfaces that allow users with no experience in programming to create data visualizations and even assist them in the design process to choose the best encodings. However, it is crucial to understand and account for every element involved in understanding data visualizations to deliver effective and clear displays of information [5, 6, 15, 16].

In this context, we propose MetaViz, a new system designed and implemented through a model-driven approach. A dashboard meta-model was developed in previous works [17-21] to provide a decomposition of the dashboards and data visualizations domain and to fuel the model-driven development.

The meta-model represents the most primitive elements of data visualizations (tangible and conceptual entities) and how they influence one another. In this sense, MetaViz takes advantage of this artifact and provides a graphical interface to generate data visualizations and dashboards through the composition of basic shapes and configurations. The aims of this research are two-fold: 1) to generate data visualizations through a usable interface and 2) to provide a learning experience in which users can learn how different configurations and visual elements influence how we see data and our reached insights.

This paper describes a work-in-progress for a pilot study to measure the understanding of the elements involved in the design of data visualizations and information dashboards. To do so, we designed a questionnaire focused on determining the level of awareness of users of Tableau and MetaViz in terms of the visual metaphors and configurations they employed to create a data visualization.

The rest of this work is organized as follows. Section 2 outlines the dashboard meta-model and the MetaViz platform. Section 3 details the study procedure and the developed questionnaire. Finally, section 4 discusses the impressions of the pilot study.

2 Background

The foundation of this work is a meta-model which has been subject to several improvements and modifications through domain engineering [22, 23] to capture the most relevant elements and factors in the domain of data visualizations and information dashboards. The current version of the dashboard meta-model is shown in Fig 1. The user [21], the layout and visual components [20] are represented along with concepts related to data, such as its domain and context [17].

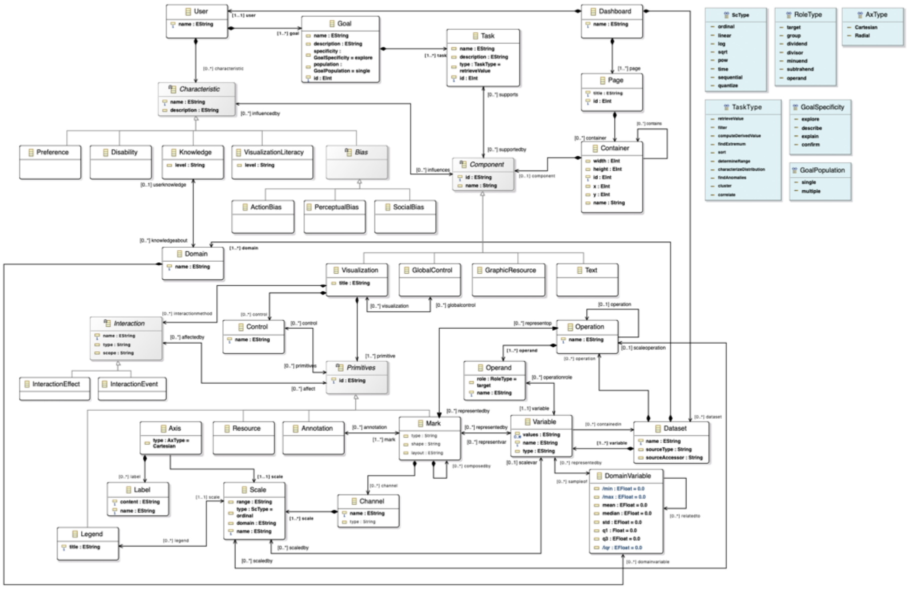


Fig. 1. The current version of the dashboard meta-model. A high-resolution version of the image can be viewed at https://doi.org/10.5281/zenodo.5788981.

Starting from this meta-model and following the model-driven development [24, 25] and software product lines [26, 27] approaches, we implemented a graphical meta-model instantiator, i.e., a platform that allows users to create models following the meta-model abstract entities. MetaViz also enables the generation of functional, real-world visualizations following the instantiated models' specifications. MetaViz's interface is shown in Fig 2. The platform is available at https://metaviz.grial.eu/.

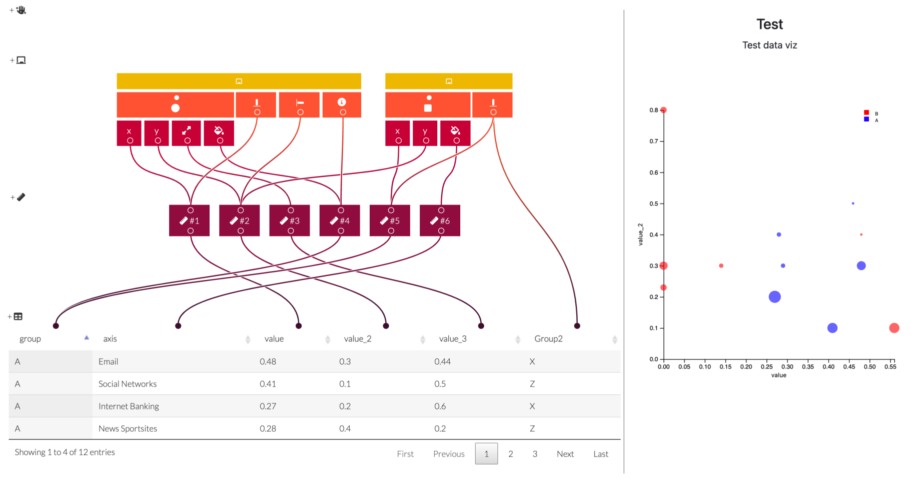


Fig. 2. MetaViz's interface. A high-resolution version of the image can be viewed at https://bit.ly/3G3JUPK.

3 Study procedure

MetaViz is focused on developing factual, conceptual, and procedural visualization knowledge [28]. The study focuses on the learning experience that the MetaViz system offers. MetaViz's strengths are flexibility and the fine-grained configuration of every element. In this sense, we want to test if the system's interface and architecture improve attention during the design process of data visualizations.

While powerful tools like Tableau offer assisted and straightforward implementation of data visualizations, MetaViz forces the user to be aware of the elements and underlying mappings they define when creating a visualization.

For this matter, we designed a procedure and a questionnaire that aims at measuring two cognitive dimensions following Bloom's taxonomy of educational objectives [28-30]:

- **Remember** - Identification of the elements that compose a data visualization at first sight, i.e., to test if the student can recognize and recall basic elements of data visualizations.
- **Understand** – Understandability of the data visualization design process, i.e., to test if the student is aware of the dimensions and elements involved in the display.

The study procedure consists of two similar parts. The first part involves a widely used data visualization tool, in this case, Tableau. Users are asked to download a test dataset and create a scatter plot that shows data values from two numerical variables and one categorical variable with Tableau. When finished, users close Tableau and answer the following questions about the scatter plot they just created:

1. Which variable was represented in the X-axis?
2. What was the maximum value of the Y-axis scale?
3. How many visual encodings were employed in your visualization?

4. How many scales were involved in your visualization?

This set of questions are focused on testing if users were aware of the design process of their own data visualizations and if they remembered basic features of their charts. Textual and graphical indications about the meaning of the data visualization terms involved in the questions are provided to avoid confusion. Explanations of the meaning of "scale" and "visual channel/encoding" are shown in Fig. 3 and Fig. 4, respectively.

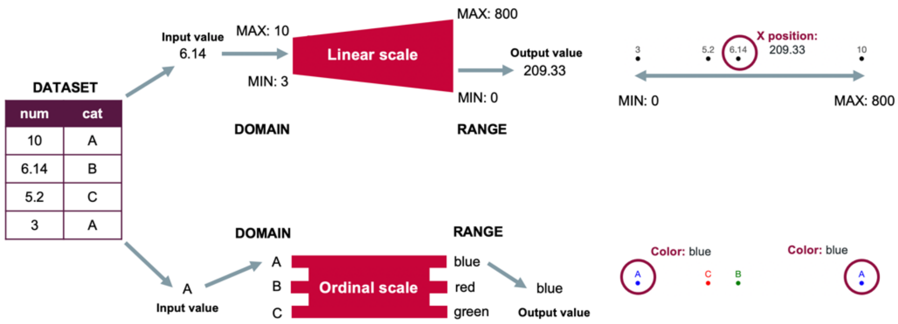


Fig. 3. Indications regarding the scale concept in the data visualization domain.

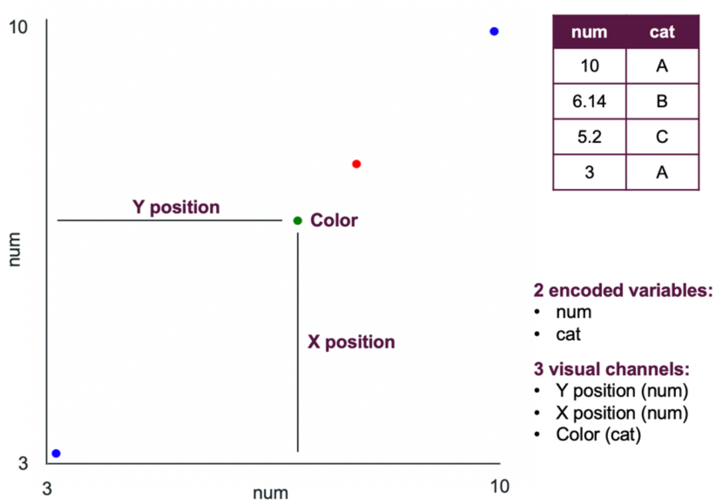


Fig. 4. Indications regarding the channels/encodings concept in the data visualization domain.

The remaining questions in this part test the ability of users to identify the elements that compose an already implemented data visualization. Different screenshots of data visualizations created in Tableau are displayed, following the next questions:

5. How many variables are involved in the (screenshot's) visualization?

6. How many scales are involved in the (screenshot's) visualization?
7. How many visual channels or encodings are involved in the (screenshot's) visualization?

Figure 5 (a) shows an example of a question from this block. Once finished, the user is asked to create an additional scatter plot with MetaViz with the same dataset. A tutorial is also provided due to the novelty of this system. When the task has been completed, the previous questionnaire is presented in the context of the MetaViz system, that is, questions 1 to 4 referring to the scatter plot created by the user in MetaViz and questions 5 to 7 with screenshots of data visualizations created in this same system, as shown in Fig. 5 (b).

Fig. 5. Questions regarding visualizations created in Tableau (a) and MetaViz (b). A high-resolution version of this image can be viewed at: https://bit.ly/3Mx4ryu.

4 Discussion and conclusions

This research presents a work-in-progress to measure the understanding of the elements involved in data visualizations and information dashboards. The study is set to validate the learning dimension of a novel system focused on the creation of data visualizations and dashboards through fine-grained components. This system (MetaViz) is based on a dashboard meta-model developed through domain engineering that captures different concepts and relationships from the data visualization domain.

We developed a study procedure and a questionnaire to determine whether users can understand and recall concepts related to data visualization with MetaViz. The questionnaire aims at measuring two cognitive dimensions –understanding and remembering– [28-30] in the context of education. In this case, MetaViz is compared to a commercial tool (Tableau), but other visual analytics tools can be employed to conduct this research.

The study will be carried out with a sample of students with data visualization skills. Future research will involve replicating this study with other samples, including lay users and people with knowledge from other domains, to compare the outcomes.

References

[1] A. Álvarez-Arana, M. Villamañe-Gironés, and M. Larrañaga-Olagaray, "Improving Assessment Using Visual Learning Analytics," *Education in the Knowledge Society,* vol. 21, no. 9, pp. 1-9, 2020, doi: 10.14201/eks.21554.

[2] D. Patil and H. Mason, *Data Driven*. Sebastopol, CA, USA: O'Reilly Media, Inc., 2015.

[3] T. H. Davenport and L. Prusak, *Working knowledge: How organizations manage what they know*. Brighton, Massachusetts, USA: Harvard Business Press, 1998.

[4] C. Ware, *Information Visualization: Perception for Design.* Burlington: Elsevier, 2004.

[5] A. Cairo, *The Functional Art: An introduction to information graphics and visualization*. San Francisco, CA, USA: New Riders, 2012.

[6] A. Cairo, *The truthful art: Data, charts, and maps for communication*. San Francisco, CA, USA: New Riders, 2016.

[7] A. Sarikaya, M. Correll, L. Bartram, M. Tory, and D. Fisher, "What Do We Talk About When We Talk About Dashboards?," *IEEE Transactions on Visualization Computer Graphics,* vol. 25, no. 1, pp. 682 - 692, 2018, doi: 10.1109/TVCG.2018.2864903.

[8] F. García-Sánchez, R. Therón, and J. Gómez-Isla, "Visual literacy in New Media: Systematic Review and Mapping of the Literature," *Education in the Knowledge Society,* vol. 20, p. 44, 2019, doi: https://doi.org/10.14201/eks2019_20_a6.

[9] E. Dimara, S. Franconeri, C. Plaisant, A. Bezerianos, and P. Dragicevic, "A task-based taxonomy of cognitive biases for information visualization," *IEEE transactions on visualization and computer graphics,* vol. 26, no. 2, pp. 1413-1432, 2018, doi: 10.1109/TVCG.2018.2872577.

[10] P. Mena, "Reducing misperceptions through news stories with data visualization: The role of readers' prior knowledge and prior beliefs," *Journalism,* vol. 0, no. 0, p. 14648849211028762, 2021, doi: 10.1177/14648849211028762.

[11] E. M. Peck, S. E. Ayuso, and O. El-Etr, "Data is Personal: Attitudes and Perceptions of Data Visualization in Rural Pennsylvania," presented at the Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems, Glasgow, Scotland Uk, 2019. [Online]. Available: https://doi.org/10.1145/3290605.3300474.

[12] Y.-S. Kim, K. Reinecke, and J. Hullman, "Data through others' eyes: The impact of visualizing others' expectations on visualization interpretation," *IEEE transactions on visualization and computer graphics,* vol. 24, no. 1, pp. 760-769, 2018, doi: 10.1109/TVCG.2017.2745240.

[13] A. Cairo, *How charts lie: Getting smarter about visual information*. New York, NY, USA: WW Norton & Company, 2019.

[14] D. A. Szafir, "The good, the bad, and the biased: Five ways visualizations can mislead (and how to fix them)," *Interactions,* vol. 25, no. 4, pp. 26-33, 2018, doi: 10.1145/3231772.

[15] S. Berinato, *Good charts: The HBR guide to making smarter, more persuasive data visualizations*. Brighton, MA, USA: Harvard Business Review Press, 2016.

[16] C. O. Wilke, *Fundamentals of data visualization: a primer on making informative and compelling figures*. Sebastopol, CA, USA: O'Reilly Media, 2019.

[17] A. Vázquez Ingelmo, A. García-Holgado, F. J. García-Peñalvo, and R. Therón Sánchez, "A Meta-modeling Approach to Take into Account Data Domain Characteristics and Relationships in Information Visualizations," in *9th World Conference on Information Systems and Technologies*, Azores, Portugal, Á. Rocha, H. Adeli, G. Dzemyda, F. Moreira, and A. M. R. Correia, Eds., 2021, vol. 2: Springer Nature, in Trends and Innovations in Information Systems and Technologies, WorldCIST 2021, pp. 570-580, doi: 10.1007/978-3-030-72651-5_54. [Online]. Available: http://hdl.handle.net/10366/145626

[18] A. Vázquez-Ingelmo, F. J. García-Peñalvo, R. Therón, and M. Á. Conde, "Representing Data Visualization Goals and Tasks through Meta-Modeling to Tailor Information Dashboards," *Applied Sciences,* vol. 10, no. 7, p. 2306, 2020. [Online]. Available: https://www.mdpi.com/2076-3417/10/7/2306.

[19] A. Vázquez-Ingelmo, F. J. García-Peñalvo, R. Therón, and A. García-Holgado, "Specifying information dashboards' interactive features through meta-model instantiation," in *LASI-SPAIN 2020. Learning Analytics Summer Institute Spain 2020: Learning Analytics. Time for Adoption?*, Valladolid, Spain, A. Martínez-Monés, A. Álvarez, M. Caeiro-Rodríguez, and Y.

Dimitriadis, Eds., June 15-16 2020, Aachen, Germany: CEUR-WS.org, in CEUR Workshop Proceedings Series, no. 2671, pp. 47-59.

[20] A. Vázquez-Ingelmo, F. J. García-Peñalvo, and R. Therón, "Capturing high-level requirements of information dashboards' components through meta-modeling," presented at the Proceedings of the Seventh International Conference on Technological Ecosystems for Enhancing Multiculturality, León, Spain, 2019. [Online]. Available: https://doi.org/10.1145/3362789.3362837.

[21] A. Vázquez Ingelmo, F. J. García-Peñalvo, R. Therón Sánchez, and M. Á. Conde González, "Extending a dashboard meta-model to account for users' characteristics and goals for enhancing personalization," (in eng), *Proceedings of LASI-SPAIN 2019. Learning Analytics Summer Institute Spain 2019: Learning Analytics in Higher Education (Vigo, Spain, June 27-28, 2019). CEUR Workshop Proceedings Series,* 2019. [Online]. Available: http://hdl.handle.net/10366/139803.

[22] I. Reinhartz-Berger, S. Cohen, J. Bettin, T. Clark, and A. Sturm, *Domain engineering*. Heidelberg, Germany: Springer, 2013.

[23] D. Bjørner, "Domain Engineering," in *Formal Methods: State of the Art and New Directions*, P. Boca, J. P. Bowen, and J. Siddiqi Eds. London: Springer London, 2010, pp. 1-41.

[24] A. Pleuss, S. Wollny, and G. Botterweck, "Model-driven development and evolution of customized user interfaces," in *Proceedings of the 5th ACM SIGCHI symposium on Engineering interactive computing systems*, 2013: ACM, pp. 13-22.

[25] A. G. Kleppe, J. Warmer, and W. Bast, "MDA Explained. The Model Driven Architecture: Practice and Promise," ed: Addison-Wesley Longman Publishing Co., Inc., Boston, MA, 2003.

[26] P. Clements and L. Northrop, *Software product lines*. Boston, MA, USA: Addison-Wesley, 2002.

[27] K. Pohl, G. Böckle, and F. J. Van der Linden, *Software Product Line Engineering: Foundations, Principles and Techniques*. New York, NY, USA: Springer-Verlag New York, Inc., 2005.

[28] M. Keck, E. Stoll, and D. Kammer, "A Didactic Framework for Analyzing Learning Activities to Design InfoVis Courses," *IEEE Computer Graphics and Applications,* vol. 41, no. 6, pp. 80-90, 2021, doi: 10.1109/MCG.2021.3115416.

[29] D. R. Krathwohl, "A Revision of Bloom's Taxonomy: An Overview," *Theory Into Practice,* vol. 41, no. 4, pp. 212-218, 2002/11/01 2002, doi: 10.1207/s15430421tip4104_2.

[30] B. Bloom, M. Engelhart, E. Furst, W. Hill, and D. Krathwohl, *Taxonomy of educational objectives: The classification of educational goals: Handbook I: Cognitive Domain*. New York, NY, USA: D. Mckay, 1956.

7.35 Appendix AI. KoopaML: A graphical platform for building machine learning pipelines adapted to health professionals

KoopaML: A graphical platform for building machine learning pipelines adapted to health professionals

Francisco José García-Peñalvo[1], Andrea Vázquez-Ingelmo[1], Alicia García-Holgado[1], Jesús Sampedro-Gómez[2], Antonio Sánchez-Puente[2], Víctor Vicente-Palacios[3], P. Ignacio Dorado-Díaz[2], Pedro L. Sánchez[2]

[1] GRIAL Research Group, University of Salamanca, Spain
andreavazquez@usal.es, aliciagh@usal.es, fgarcia@usal.es

[2] Cardiology Department, Hospital Universitario de Salamanca, SACyL. IBSAL, Facultad de Medicina, University of Salamanca, and CIBERCV (ISCiii), Spain
jmsampedro@saludcastillayleon.es, asanchezpu@saludcastillayleon.es, pidorado@saludcastillayleon.es, plsanchez@saludcastillayleon.es

[3] Philips Healthcare, Spain
victor.vicente.palacios@philips.com

Abstract

Machine Learning (ML) has extended its use in several domains to support complex analyses of data. The medical field, in which significant quantities of data are continuously generated, is one of the domains that can benefit from the application of ML pipelines to solve specific problems such as diagnosis, classification, disease detection, segmentation, assessment of organ functions, etc. However, while health professionals are experts in their domain, they can lack programming and theoretical skills regarding ML applications. Therefore, it is necessary to train health professionals in using these paradigms to get the most out of the application of ML algorithms to their data. In this work, we present a platform to assist non-expert users in defining ML pipelines in the health domain. The system's design focuses on providing an educational experience to understand how ML algorithms work and how to interpret their outcomes and on fostering a flexible architecture to allow the evolution of the available components, algorithms, and heuristics.

Keywords

Information system, Medical data management, Medical imaging management, Artificial Intelligence, Health platform, HCI

I. Introduction

Machine Learning (ML) has become a powerful approach to tackle complex tasks that involve analyzing significant amounts of data. Data-intensive contexts, such as the health domain, benefit directly from applying ML algorithms to their data, supporting tasks such as identifying patterns, clustering, classification, predictions, etc., that could become time- and resource-consuming if approached through manual paradigms. The application of ML to health data has proven its usefulness in specific challenges like diagnoses, disease detection, segmentation, assessment of organ functions, etc. [1-3].

However, applying ML approaches is not straightforward. More specifically, using them in sensitive domains (such as health) could be hazardous if practitioners do not fully understand the results derived from the models.

ML does not only consist of applying a set of pre-defined functions. It needs a deep understanding of the input data, the transformations that need to be performed to fit a model, the selection of a proper model, and its quality metrics before using trained models in production. Otherwise, the outputs could lead to wrong conclusions, losses, discrimination, and even negligence [4-7].

Therefore, it is necessary to balance data domain knowledge and ML expertise. While ML experts have a wealth of knowledge about ML algorithms, they can lack understanding regarding the input data. The same applies to health professionals; they have a profound knowledge of the data domain, but they would not obtain quality models without programming or ML skills.

In this scenario, it is necessary to provide practitioners with tools that alleviate this knowledge gap, enabling health professionals to implement ML pipelines and learn how, when, and why to apply specific models or functions to their data. This way, the introduction of ML in medical tasks could yield complementary support to automate and enhance decision-making processes without consuming an excessive quantity of resources and time.

In this context we pose the following research question:

RQ1. Which features can ease the application of ML algorithms in the medical context?

Driven by this research question, we present a graphical platform (KoopaML) to offer intuitive and educational interfaces to build and run ML pipelines to tackle these challenges. The primary target audience of this platform is non-expert users interested in learning and applying ML models to their domain data. We followed a user-centered design approach to capture relevant requirements and necessities from potential user profiles involved in this context.

In addition, we focused on providing a flexible architecture to allow expert users to extend the platform's functionality through new custom algorithms, components, or new heuristics to guide the definition of ML pipelines.

In this paper, we describe the design process, the platform's architecture, its underlying processes, and the feedback obtained from experts regarding the first development stages of the system.

The rest of the work is structured as follows. Section 2 provides an overview of similar tools for learning and building ML and data science pipelines. Section 3 describes the methodology followed for eliciting requirements and the technologies employed to implement the platform. Section 4 details the platform's architecture and modular decomposition, while section 5 describes the implemented functionalities. Finally, sections 7 and 8 discuss the results and conclude the work, respectively.

II. Related Work

Plenty of helper tools has been developed due to the increasing popularity of ML. Specifically, there are three main categories: programming frameworks and libraries, platforms for experts and non-experts, and platforms that support learning and understanding regarding how ML algorithms and pipelines work.

The first category encloses several well-known programming libraries: TensorFlow [8], Apache Mahout [9], and other Python frameworks like PyTorch (https://pytorch.org/), Scikit-learn (https://scikit-learn.org/), or Keras.io (https://keras.io/). These libraries provide an abstraction layer to implement ML models, but they require programming skills to employ them properly.

The second category focuses on visual environments that assist users through intuitive interfaces in creating and defining ML pipelines. Weka, for instance, provides a collection of algorithms for data mining tasks. One of its environments enables users to define data streams by connecting nodes representing data sources, preprocessing tasks, evaluation methodologies, visualizations, or algorithms, among other [10, 11].

On the other hand, Orange Data Mining Field [12] allows the definition of data mining workflows, with several methodologies, operations, and visualizations available through a user-friendly interface. The possibility of introducing customized dashboards [12-14] to present the outcomes of ML tasks an extremely valuable feature to ease the comprehension of the pipeline stages. Another tool with similar features is Rapid Miner [15], which follows a node-and-link philosophy to specify and define ML workflows. These applications provide robust and complex features through intuitive interfaces and interaction methods, adding abstraction to programming libraries.

The last category refers to platforms whose primary goal is to offer a didactic experience and learning resources to ease understanding ML algorithms and workflows. Tools within this category provide user-friendly and simple graphical interfaces avoiding technical details. Examples include Machine Learning for Kids (https://machinelearningforkids.co.uk/) or LearningML (https://web.learningml.org/).

Although several solutions are developed to assist non-expert users in the definition of ML pipelines, it is difficult to adapt them to specific contexts with particular necessities and requirements. For these reasons, we opted to develop a customized tool focused on the provenance of an educative experience for health professionals that want to start applying ML models to support their tasks.

III. Methodology

A. Requirements Elicitation

We identified the main features and specifications of the platform through a requirements elicitation process. Specifically, we interviewed potential users and domain experts, including physicians, computer scientists, and managers.

The output of this process was the description of the platform's basic features:

1. Definition of ML pipelines
2. Execution of ML pipelines
3. Interpretation of ML results
4. Visualization of ML results
5. Data validation
6. Heuristics management

The first two features are related to implementing ML pipelines by connecting different tasks, including data preprocessing and cleaning, ML algorithms, and evaluation functions. The platform allows users to choose different ML algorithms and configure their parameters. Users can personalize their pipelines by connecting nodes and analyzing each step's intermediate results.

Features 3 and 4 are related to the outcomes obtained during the execution of the pipelines. Each stage will output new results, and these results need to be understood to gain insights. For these reasons, the platform needs to provide methods to convey and assist the interpretation of the pipeline outputs through explanations, annotations, and data visualizations. This process is vital because a wrong interpretation of the results could lead to useless results and lose all its potential benefits.

On the other hand, the quality of the training process not only depends on the algorithm's configuration but also on the quality of input data. The platform needs to support validation processes and emphasize cleaning and preprocessing functions before training ML models. This feature focuses on providing information regarding the applicability of the available algorithms to the input data and potential issues (missing values, data imbalance, data samples, data types, etc.).

The last functionality refers to applying heuristics to assist non-expert users in the definition of ML pipelines. The management of heuristics and recommendation rules should be flexible to support the evolution of the suitability of algorithms depending on the context. Therefore, the platform will allow the modification and addition of new heuristics to provide more flexibility and build customized rule-based recommenders.

During the elicitation process, two user roles were identified. This categorization of users is essential to adapt the functionalities depending on the role, as well as their privileges:

- Non-expert users. The primary users of the platform. Non-expert users (mainly physicians) who know the data domain are interested in IA and ML but don't have enough skills to create ML pipelines programmatically.
- AI experts. Experts will have access to the ML pipelines workspace, but they will also have privileges to define and modify heuristics to configure the recommendations or preferred workflows of the platform.

B. Development

As introduced, one of the main goals of this work is to provide a flexible platform with the capability to evolve to include discoveries in ML. Therefore, it is crucial to rely on flexible technologies and paradigms that support the reusability of components.

ML pipelines share common features and can be represented through abstract elements to leverage their commonalities and foster the reusability of core assets. We followed the software product line (SPL) paradigm and domain engineering to capture ML pipelines and tasks' commonalities and variability points and arrange the software components accordingly [16-20].

With this approach, it is possible to reuse these "building blocks" and modify/add new ones without impacting the rest of them. On the other hand, building each pipeline task as an independent component with well-defined inputs and outputs also meets the requirement of inspecting intermediate results and even executing pipelines step by step.

We materialized the variability of pipelines through SciLuigi (https://github.com/pharmbio/sciluigi), a wrapper for Spotify's Luigi Python library (https://github.com/spotify/luigi), which supports the definition of dynamic workflows avoiding hard-coded dependencies [21, 22].

C. Validation

To validate the first version of KoopaML, we carried out an expert judgement [23] validation with experts from the medical and AI fields. Three experts were recruited to thoroughly test the platform and seek for issues regarding its contents and interaction mechanisms.

The three participants are AI developers in the medical domain, so they were able to test the platform from the two perspectives.

IV. Architecture

Providing flexible and extensible architecture is crucial in this field, as approaches constantly improve and evolve. This section outlines the platform's architecture and the mechanisms employed to support the evolution of its components.

A. Modules

The architecture of KoopaML is based on different modules connected by information flows. One of the primary purposes of this design is to provide flexible pipelines with reusable components.

In this regard, we followed a domain engineering approach through the previously described requirements elicitation process with potential users and literature reviews.

Following this approach, we propose four general functional blocks that will interact and collaborate among them to provide support for the implementation of flexible ML workflows:

- User management module
- Heuristics management module
- Pipelines management module
- Tasks management module

The user management module provides the services related to authentication, sessions, and roles. The heuristics management module allows IA experts to modify the heuristics through a graphic interface. The pipelines management module provides a workspace to create ML pipelines using visual elements. Finally, the tasks management module defines the operations related to each ML pipeline potential stage.

Figure 1 shows the schematic overview of the platform's architecture with the C4 model notation [24].

B. Pipelines' structure

While the previous functional blocks provide flexibility to evolve the system's features, they still need fine-grained flexibility regarding the implementation of ML pipelines.

Following the software product line architecture paradigm [16-20], we divided ML pipelines into fine-grained tasks with well-defined inputs and outputs. Through this approach, the tasks management module acts as a repository of loosely coupled ML-related tasks, in which algorithms and operations can be added and modified without impacting the features of the remaining modules/tasks.

As explained in the methodology section, this encapsulation of ML tasks is achieved through the SciLuigi library. Figure 2 outlines the structure of the pipelines following this approach.

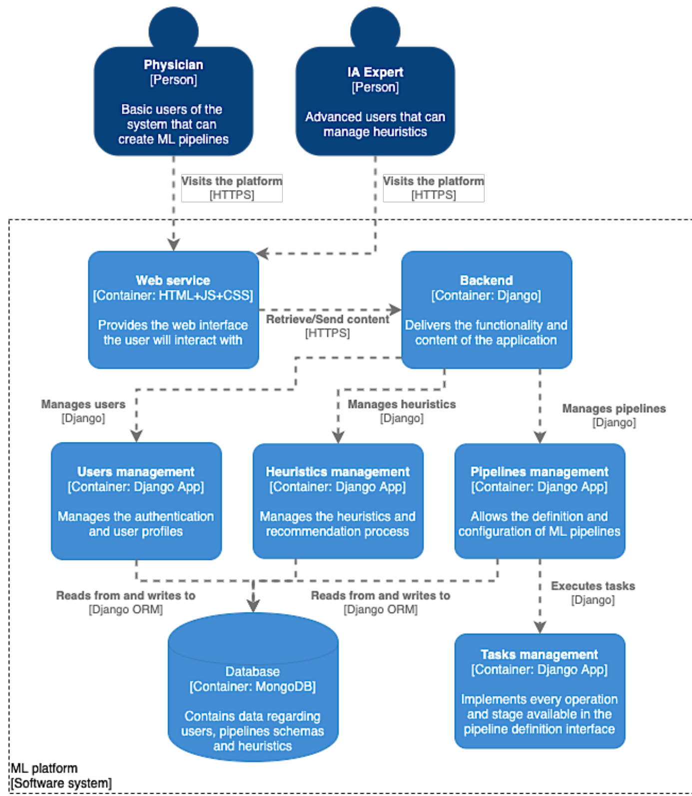


Fig. 1. Outline of the platform's architecture.

Tasks are categorized following their high-level functionality (tasks related to data upload, data preprocessing, ML algorithms, or evaluation metrics). Then, more specific tasks are implemented; for example, within the "ML algorithms" category, we can find particular algorithms such as Naïve Bayes, Random Forest, Linear Regression, etc.

Users can instantiate nodes from each category and connect them through their inputs and outputs. These inputs and outputs are also categorized to ensure that information flows are compatible among the instantiated nodes.

The connection restrictions between nodes are implemented in the interface to ensure that the SciLuigi pipeline is correctly instantiated. With this method, the construction of the final SciLuigi pipeline is straightforward.

The simplified code in Figure 3 outlines the implementation of a SciLuigi workflow through the pipeline specification defined by the user in the graphical interface. The main challenge was related to the dynamic connection of inputs and outputs. SciLuigi requires knowing the specific inputs/outputs names beforehand to connect them through explicit attribute value assignment. Lines 14-17 (Figure 3) show how this issue was solved using the *setattr* and *getattr* Python methods, allowing dynamic access to class attributes.

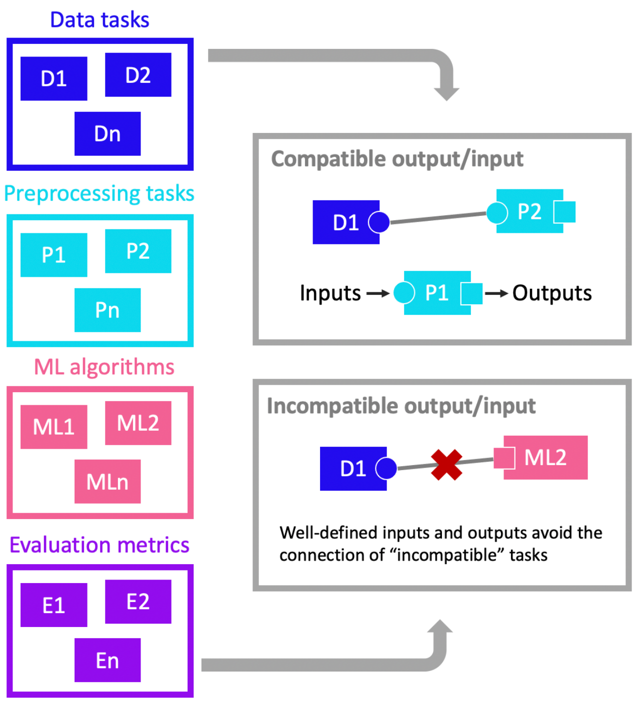


Fig. 2. Outline of programming approach. Each task contains its own logic and belongs to a specific category. Inputs and outputs compatibilities (in terms of information flows) are computed from each node's logic.

```
tasks = { }

for node in pipeline.config:
    tasks[node.id] = self.new_task(create_instance[node.type],
                                   pipeline_id=pipeline.id,
                                   node_id=node.id,
                                   node.params)

    end_nodes = [ ]
    for node in pipeline.config:
        if len(node.inputs) > 0:
            for input in node.inputs:
            # Connect the node inputs to the corresponding node outputs
            setattr(tasks[node.id],
                    "in_{0}".format(input.input_name),
                    getattr(tasks[input.connected_node.id],
                    "out_{0}".format(input.connected_node.output_name)))

        if len(node.outputs) == 0:
            end_nodes.append(tasks[node_id])

    return tuple(end_nodes)
```

Fig. 3. Algorithm to materialize a pipeline specification into a SciLuigi workflow (syntax simplified for readability).

This solution provides more flexibility and eases the addition and modification of new tasks, as the whole pipeline can be instantiated without hard-coding specific dependencies or class types.

V. KoopaML

A. Prototype

A prototype was developed and evaluated to complement the requirement elicitation process through a focus group. This methodology enabled us to capture more requirements and validate the platform's conceptual design before its implementation.

Figure 4 shows a screenshot of the interface for defining ML pipelines through node-link structures.

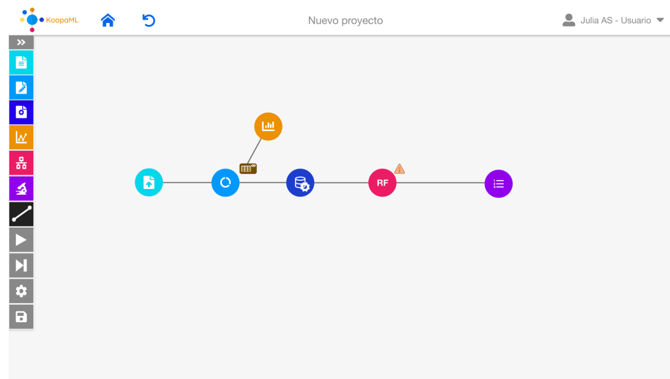


Fig. 4. Prototype of the workspace to define ML pipelines.

The focus group involved different user profiles, including physicians and AI experts related to the health domain. The outcomes of this study can be consulted in [25]. The feedback was positive and helpful for starting the implementation of the tool.

B. Functional system

As explained throughout this work, a crucial characteristic of the interface is that it should be simple to avoid overwhelming users with several complex concepts at once and robust to enable the definition of ML pipelines with enough detail. This section provides an overview of the interface proposal and the different features of the first version of KoopaML.

1. ML pipelines

When creating a new project or pipeline, the system displays an empty workspace with a toolbar containing the tasks included in the ML workflow. As introduced in previous sections, tasks are divided into high-level categories to ease users' search of specific nodes (Figure 5).

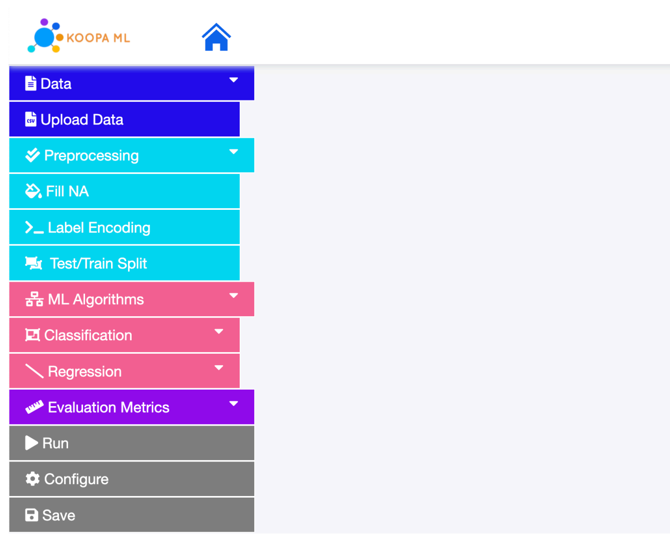


Fig. 5. New project workspace and available nodes.

Users can click on specific tasks or drag and drop them into the workspace to start configuring the pipeline. Figure 6 shows the "Upload CSV" node. This node is particularly complex because several circumstances need to be considered when uploading data:

1. CSV files can be separated by different characters, such as commas or semicolons. For this reason, the node allows the configuration of varying separator values through a text input.

2. Some nodes could take as an input a single column (or a subset of them). That is why each dataset's columns need to be considered single outputs and be accessible to create connections among nodes.
3. The whole dataset is also considered a single output ("Data" socket in Figure 6) to avoid multiple column connections and ease the data flows. This output includes the whole dataset (the set of all columns contained in the uploaded file).
4. Related to the previous point, some columns might be discarded from the dataset (i.e., columns that hold several missing values or aren't relevant to the problem). A checkbox beside each column allows users to select the columns that will be part of the dataset.
5. Finally, data related to the health domain can hold a significant quantity of variables. However, showing all variables as outputs in the "Upload CSV" node at once could impact the user experience. For this reason, a threshold has been configured to show only the first five columns of a dataset, allowing the user to add the remaining columns through a multiple selection input. This way, users can focus only on variables that need explicit connections through their ML pipeline.

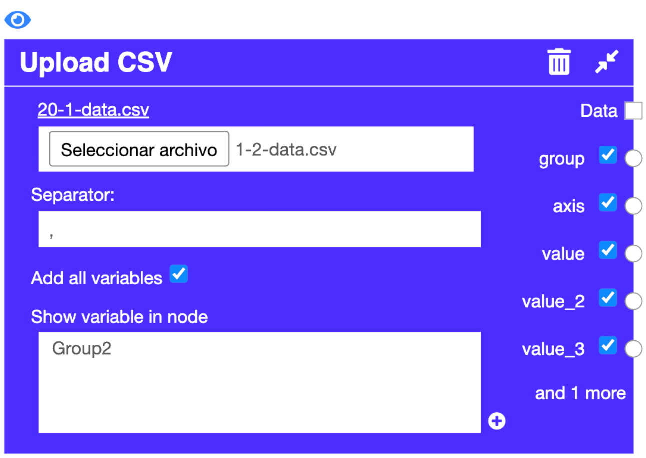


Fig. 6. A node for uploading CSV files.

Figure 7 shows a simple ML workflow in which:

1. Categorical data is encoded through a Label Encoder
2. The output from the label encoding process is then split into training and test sets. This node needs to know the variable to predict to perform the division of data. In this specific case, the variable "group" will be the one to be expected through this pipeline.
3. Training datasets are connected to a Random Forest classifier
4. Finally, the trained model and the test datasets are connected to an evaluation node to measure the model's accuracy.

Users can execute the pipeline whenever they want by clicking the "Run" button, triggering the backend to build the pipeline by connecting tasks using the algorithm presented in Figure 3. Once the pipeline has been executed, the workspace displays the results (or any error) individually in each node (bottom image of Figure 7). Storing intermediate results leverages one of the main benefits of using SciLuigi, which is the possibility of re-running failed tasks individually without triggering the whole pipeline again.

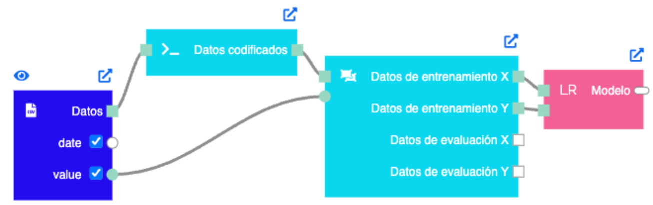
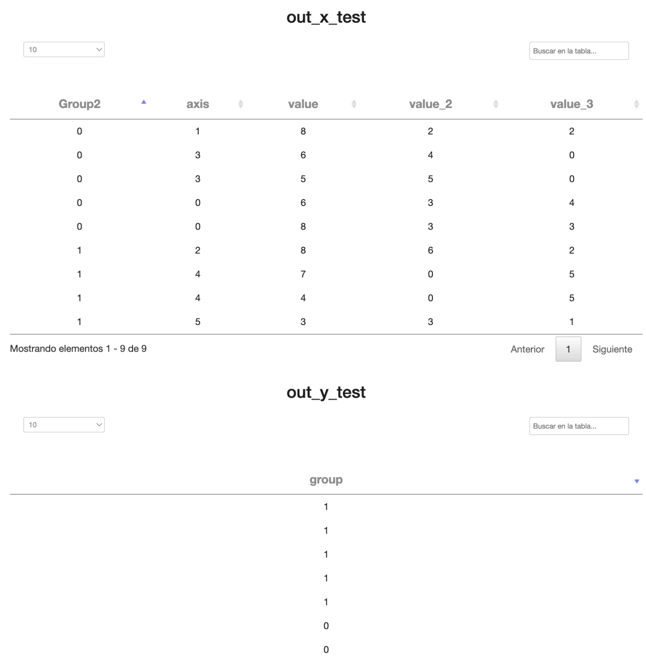


Fig. 7. Execution and results of an ML pipeline. Intermediate results of each node can be consulted by clicking on the top-right icon of each node.

Through this approach, intermediate results can also be inspected individually. On the one hand, Figure 8 displays the intermediate results from the test/train splitting node. This node yields four results: test and training datasets separated by the column to predict. The figure shows two of these intermediate results (the test datasets).



out_x_test

Group2	axis	value	value_2	value_3
0	1	8	2	2
0	3	6	4	0
0	3	5	5	0
0	0	6	3	4
0	0	8	3	3
1	2	8	6	2
1	4	7	0	5
1	4	4	0	5
1	5	3	3	1

Mostrando elementos 1 - 9 de 9 Anterior 1 Siguiente

out_y_test

group
1
1
1
1
1
0
0

Fig. 8. Results were derived from splitting the uploaded data into test and training datasets.

Evaluation metrics are also treated as intermediate results. In this case, the measurement of the accuracy of the trained model yielded 33% of correct predictions.

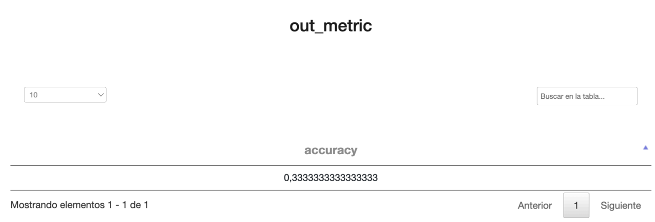
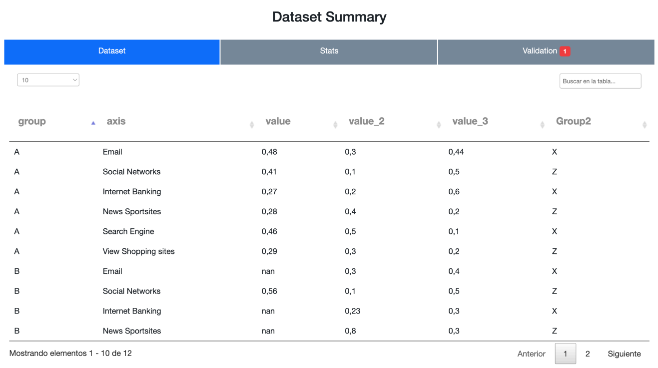


Fig. 9. Accuracy of the trained model. Note that low accuracy is related to the small dataset that illustrates the system's functionalities.

2. *Data validation*

Data validation and exploratory analysis are crucial steps when building successful ML pipelines. If data is not properly inspected and preprocessed, trained models could yield useless results. KoopaML provides a summary screen to assist users in the exploration process. This section is divided into three main blocks.

The first block provides a table view of the whole dataset. This view allows users to see all columns and rows of the uploaded data files and navigate through them in detail (Figure 9).


Dataset Summary

Dataset | Stats | Validation 1

10 — Buscar en la tabla...

group	axis	value	value_2	value_3	Group2
A	Email	0,48	0,3	0,44	X
A	Social Networks	0,41	0,1	0,5	Z
A	Internet Banking	0,27	0,2	0,6	X
A	News Sportsites	0,28	0,4	0,2	Z
A	Search Engine	0,46	0,5	0,1	X
A	View Shopping sites	0,29	0,3	0,2	Z
B	Email	nan	0,3	0,4	X
B	Social Networks	0,56	0,1	0,5	Z
B	Internet Banking	nan	0,23	0,3	X
B	News Sportsites	nan	0,8	0,3	Z

Mostrando elementos 1 - 10 de 12 — Anterior 1 2 Siguiente

Fig. 9. Results were derived from splitting the uploaded data into test and training datasets.

The second summary block is a data dashboard in which practical data details, such as the distribution of values, data types, number of missing values, or a correlation matrix, are presented visually to ease the analysis of the dataset characteristics (Figure 10). The dashboard is automatically generated and tailored according to the user needs [26-28].

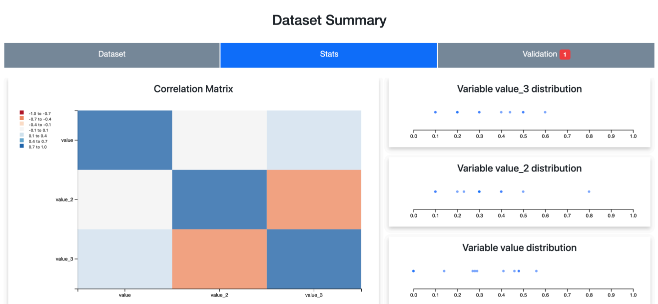
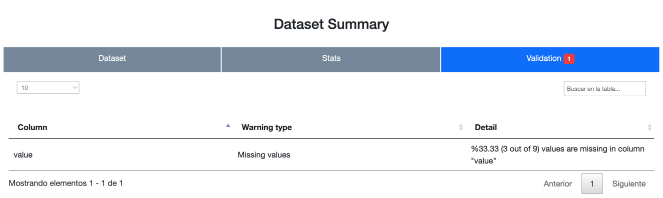


Fig. 10. Information dashboard of the input dataset characteristics.

Finally, the last block is focused on alerting users regarding potential issues of the dataset (Figure 11), such as columns with significant quantities missing values, mixed data types, unbalanced categories, etc. Users are encouraged to consider or solve these issues through this feature before using the dataset in a pipeline.


Dataset Summary

Dataset | Stats | Validation 1

10 — Buscar en la tabla...

Column	Warning type	Detail
value	Missing values	%33.33 (3 out of 9) values are missing in column "value"

Mostrando elementos 1 - 1 de 1 — Anterior 1 Siguiente

Fig. 11. Validation screen.

3. Heuristics management

As explained before, one of the goals of the platform users is to learn from the experience of developing pipelines and build skills related to the application of ML. However, this learning experience needs to be guided by expert knowledge.

We have tackled this challenge through the definition and management of custom heuristics. KoopaML allows expert users to design heuristics in graphical decision trees to yield recommendations and guide the implementation of pipelines.

Heuristics are represented through the DSL provided by the flowchart.js (https://github.com/adrai/flowchart.js) library. This library allows textual and graphical representation of flow charts, providing a fine-grain manipulation of heuristics and rule-based recommendations (Figure 12).

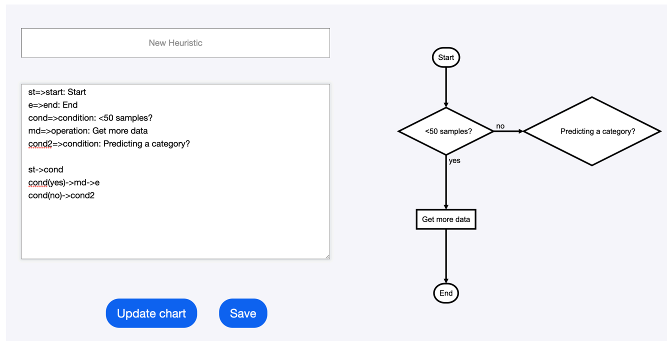


Fig. 12. Example of the definition of a heuristic.

VI. Expert Validation

The results of the expert validation were favorable. Overall, the platform was rated as useful to overcome the difficulties of creating ML pipelines in a medical context.

Regarding the issues encountered, apart from minor bugs that were fixed, the following can be highlighted:

- **Error reports.** The experts pointed out the possibility of having a variety of errors related to the execution of the pipeline. In the current version of KoopaML, these errors were displayed through tooltips associated to each node. However, experts indicated that it might be useful to have an unified report listing every error or warning raised during the execution of the pipeline.
- **Model metrics.** KoopaML allows the computation of different metrics to validate the trained models. For this matter, the user needs to select and connect every metric they want to calculate. This could be time-consuming if several metrics are to be analyzed. In this sense, the experts advised the possibility of unifying every metric on a node, and let the user select the metrics directly from there instead of carrying out the selection one by one.
- **Data visualizations.** The data summary presented in the previous section was highly valued by the experts. Following this idea, they recommended implementing a dashboard with visualizations related to model metrics as well.
- **Cross validation.** The experts pointed out that, in practice, they use cross validation [29], and thus, that the platform should support this approach.

Other comments were related to the addition of a wide set of algorithms and metrics, as well the possibility of configuring the hyperparameters of the algorithms through the interface.

VII. Discussion

This work presents the first version of KoopaML: a platform for automating and learning the definition of ML pipelines. We followed a user-centered approach for the design and development process, considering the primary goal of the system: to ease the application of ML for non-specialized users.

This version has been subject to iterative development with continuous feedback from experts. For instance, the "Upload CSV" node design shown in Figure 6 resulted from different evaluations in which domain experts exposed issues encountered or potential improvements when uploading their domain data.

Although there are commercial tools that tackle the automation of these processes, the specific requirements that arise from the medical context asked for a customized platform that aligns with the necessities of end-users (in this case, physicians with lack of data science skills but that are interested in applying ML).

On the other hand, another related benefit of the customized tool is implementing communication mechanisms among other already developed devices for the cardiology department at the University Hospital of Salamanca [30]. Connecting different platforms would foster the creation of a technological ecosystem [31] with powerful and transparent data management and data science features adapted to the health sector requirements.

The platform's architecture is designed to allow flexibility and evolution due to the changing nature of AI and ML methods. The abstraction of pipelines into tasks with well-defined inputs and outputs has facilitated the user interface design and the final implementation of the workflows through libraries such as SciLuigi, matching the same node-link structures.

In addition to the workspace for instantiating pipelines, the platform also provides an interface to support the exploratory analysis of data. This interface was included after the evaluation of the platform by expert users, who asked for more feedback related to the input data.

Finally, one novel feature of KoopaML is the heuristics management module. This module enables the definition of heuristics through a DSL and its graphical representation. Heuristics can be stored to rely on different knowledge bases depending on the data domain, for example. The dynamic heuristic definition fosters the flexibility of the recommendations and guided support provided within the workspace during the implementation of ML pipelines. Moreover, their structured format allows the inclusion of external heuristics from other knowledge bases stores [32, 33].

Regarding the expert validation, the results were highly valuable and useful to set the foundations of new improvements and features, as well as to identify minor bugs. Having experts from both AI and medical fields enabled the identification of issues and shortcomings of the current version of the platform. For these reasons, we will continue performing this kind of evaluations, as they provide insights related to theoretical concepts that will be difficult to reach with lay users.

Following the research question posed in the introduction and the results of the expert validation, the platform has been developed taking into account the necessities of the medical domain. The implementation of an interface with simple and visual mechanisms (such as drag and drop or visually connecting two nodes to instantiate a pipeline) set the foundations for a platform that can be used by non-expert users.

On the other hand, the development of the heuristics management module will also allow the definition of recommendations that could be adapted to any kind of user. These features will provide additional assistance while creating and interpreting ML pipelines.

VIII. Conclusions

This work describes the design process, architecture, and features of KoopaML: a graphical platform for building machine learning pipelines adapted to health professionals.

The platform has been designed to support the evolution and addition of new tasks related to ML pipelines through abstraction mechanisms. The abstraction of tasks has allowed simplifying the user interface and the automatic implementation of the graphically instantiated pipelines.

KoopaML assists users in the definition of ML pipelines, execution of ML pipelines, interpretation and visualization of ML results, data validation, and heuristics management.

Future research lines will involve further expert validations of the platform, as well as in-depth user tests to measure the usability, ease of use, and effectiveness of the tool.

Acknowledgment

This research work has been supported by the Spanish *Ministry of Education and Vocational Training* under an FPU fellowship (FPU17/03276). This research was partially funded by the Spanish Government Ministry of Economy and Competitiveness through the DEFINES project grant number (TIN2016-80172-R) and the Ministry of Science and Innovation through the AVisSA project grant number (PID2020-118345RB-I00). This work was also supported by national (PI14/00695, PIE14/00066, PI17/00145, DTS19/00098, PI19/00658, PI19/00656 Institute of Health Carlos III, Spanish Ministry of Economy and Competitiveness and co-funded by ERDF/ESF, "Investing in your future") and community (GRS 2033/A/19, GRS 2030/A/19, GRS 2031/A/19, GRS 2032/A/19, SACYL, Junta Castilla y León) competitive grants.

References

[1] G. Litjens *et al.*, "A survey on deep learning in medical image analysis," (in eng), *Med Image Anal,* vol. 42, pp. 60-88, Dec 2017, doi: 10.1016/j.media.2017.07.005.

[2] S. González Izard, R. Sánchez Torres, Ó. Alonso Plaza, J. A. Juanes Méndez, and F. J. García-Peñalvo, "Nextmed: Automatic Imaging Segmentation, 3D Reconstruction, and 3D Model Visualization Platform Using Augmented and Virtual Reality," (in eng), *Sensors (Basel),* vol. 20, no. 10, p. 2962, 2020, doi: 10.3390/s20102962.

[3] S. G. Izard, J. A. Juanes, F. J. García Peñalvo, J. M. G. Estella, M. J. S. Ledesma, and P. Ruisoto, "Virtual Reality as an Educational and Training Tool for Medicine," *Journal of Medical Systems,* vol. 42, no. 3, p. 50, 2018/02/01 2018, doi: 10.1007/s10916-018-0900-2.

[4] J. C. Weyerer and P. F. Langer, "Garbage in, garbage out: The vicious cycle of ai-based discrimination in the public sector," in *Proceedings of the 20th Annual International Conference on Digital Government Research*, Dubai, United Arab Emirates, 2019, pp. 509-511, doi: https://doi.org/10.1145/3325112.3328220.

[5] X. Ferrer, T. van Nuenen, J. M. Such, M. Coté, and N. Criado, "Bias and Discrimination in AI: a cross-disciplinary perspective," *IEEE Technology and Society Magazine,* vol. 40, no. 2, pp. 72-80, 2021, doi: 10.1109/MTS.2021.3056293.

[6] S. Hoffman, "The Emerging Hazard of AI-Related Health Care Discrimination," *Hastings Center Report,* vol. 51, no. 1, pp. 8-9, 2021, doi: 10.1002/hast.1203.

[7] S. Wachter, B. Mittelstadt, and C. Russell, "Why fairness cannot be automated: Bridging the gap between EU non-discrimination law and AI," *Computer Law & Security Review,* vol. 41, p. 105567, 2021, doi: 10.2139/ssrn.3547922.

[8] M. Abadi *et al.*, "TensorFlow: A System for Large-Scale Machine Learning," in *12th USENIX Symposium on Operating Systems Design and Implementation OSDI 16*. Savannah, GA: USENIX Association, 2016, pp. 265-283.

[9] R. Anil *et al.*, "Apache Mahout: Machine Learning on Distributed Dataflow Systems," *Journal of Machine Learning Research,* vol. 21, no. 127, pp. 1-6, 2020. [Online]. Available: https://jmlr.org/papers/v21/18-800.html.

[10] E. Frank, M. Hall, G. Holmes, R. Kirkby, B. Pfahringer, and I. H. Witten, "Weka-A Machine Learning Workbench for Data Mining," in *Data Mining and Knowledge Discovery Handbook*, O. Maimon and L. Rokach Eds. Boston, MA: Springer, 2009.

[11] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, "The WEKA data mining software: an update," *SIGKDD Explor. Newsl.,* vol. 11, no. 1, pp. 10–18, 2009, doi: 10.1145/1656274.1656278.

[12] A. Vázquez-Ingelmo, F. J. García-Peñalvo, and R. Therón, "Information Dashboards and Tailoring Capabilities - A Systematic Literature Review," *IEEE Access,* vol. 7, pp. 109673-109688, 2019, doi: 10.1109/ACCESS.2019.2933472.

[13] A. Sarikaya, M. Correll, L. Bartram, M. Tory, and D. Fisher, "What Do We Talk About When We Talk About Dashboards?," *IEEE Transactions on Visualization Computer Graphics,* vol. 25, no. 1, pp. 682 - 692, 2018, doi: 10.1109/TVCG.2018.2864903.

[14] S. Few, *Information dashboard design.* Sebastopol, CA, USA: O'Reilly Media, 2006.

[15] S. Land and S. Fischer, "Rapid miner 5," *Rapid-I GmbH,* 2012.

[16] J. Bosch, "From software product lines to software ecosystems," in *SPLC*, 2009, vol. 9, pp. 111-119.

[17] L. Chen, M. Ali Babar, and N. Ali, "Variability management in software product lines: a systematic review," 2009.

[18] P. Clements and L. Northrop, *Software product lines.* Addison-Wesley Boston, 2002.

[19] C. Kästner, S. Apel, and M. Kuhlemann, "Granularity in software product lines," in *2008 ACM/IEEE 30th International Conference on Software Engineering*, 2008: IEEE, pp. 311-320.

[20] J. Van Gurp, J. Bosch, and M. Svahnberg, "On the notion of variability in software product lines," in *Proceedings Working IEEE/IFIP Conference on Software Architecture*, 2001: IEEE, pp. 45-54.

[21] S. Lampa, J. Alvarsson, and O. Spjuth, "Towards agile large-scale predictive modelling in drug discovery with flow-based programming design principles," *Journal of Cheminformatics,* vol. 8, no. 1, p. 67, 2016/11/24 2016, doi: 10.1186/s13321-016-0179-6.

[22] S. Lampa, M. Dahlö, J. Alvarsson, and O. Spjuth, "SciPipe: A workflow library for agile development of complex and dynamic bioinformatics pipelines," *GigaScience,* vol. 8, no. 5, 2019, doi: 10.1093/gigascience/giz044.

[23] R. Skjong and B. H. Wentworth, "Expert judgment and risk perception," in *Proceedings of the Eleventh (2001) International Offshore and Polar Engineering Conference (Stavanger, Norway, June 17-22, 2001)*: International Society of Offshore and Polar Engineers, 2001.

[24] S. Brown. "The C4 Model for Software Architecture." https://c4model.com/ (accessed 16-05-2022).

[25] A. García-Holgado *et al.*, "User-Centered Design Approach for a Machine Learning Platform for Medical Purpose," Cham, 2021: Springer International Publishing, in Human-Computer Interaction, pp. 237-249.

[26] A. Vázquez Ingelmo, A. García-Holgado, F. J. García-Peñalvo, and R. Therón Sánchez, "A Meta-modeling Approach to Take into Account Data Domain Characteristics and Relationships in Information Visualizations," in *9th World Conference on Information Systems and Technologies*, Azores, Portugal, Á. Rocha, H. Adeli, G. Dzemyda, F. Moreira, and A. M. R. Correia, Eds., 2021, vol. 2: Springer Nature, in Trends and Innovations in Information Systems and Technologies, WorldCIST 2021, pp. 570-580, doi: 10.1007/978-3-030-72651-5_54. [Online]. Available: http://hdl.handle.net/10366/145626

[27] A. Vázquez-Ingelmo, A. García-Holgado, F. J. García-Peñalvo, and R. Therón, "Proof-of-concept of an information visualization classification approach based on their fine-grained features," *Expert Systems,* vol. n/a, no. n/a, p. e12872, In Press, doi: https://doi.org/10.1111/exsy.12872.

[28] A. Vázquez-Ingelmo, F. J. García-Peñalvo, and R. Therón, "Taking advantage of the software product line paradigm to generate customized user interfaces for decision-making processes: a case study on university employability," *PeerJ Computer Science,* vol. 5, p. e203, 2019/07/01 2019, doi: 10.7717/peerj-cs.203.

[29] C. Schaffer, "Selecting a classification method by cross-validation," *Machine Learning,* vol. 13, no. 1, pp. 135-143, 1993.

[30] F. García-Peñalvo *et al.*, "Application of Artificial Intelligence Algorithms Within the Medical Context for Non-Specialized Users: the CARTIER-IA Platform," *International Journal of Interactive Multimedia and Artificial Intelligence,* vol. 6, no. 6, 2021.

[31] A. García-Holgado and F. J. García-Peñalvo, "Validation of the learning ecosystem metamodel using transformation rules," *Future Generation Computer Systems,* vol. 91, pp. 300-310, 2019/02/01/ 2019, doi: https://doi.org/10.1016/j.future.2018.09.011.

[32] A. Martínez-Rojas, A. Jiménez-Ramírez, and J. Enríquez, "Towards a Unified Model Representation of Machine Learning Knowledge," presented at the Proceedings of the 15th International Conference on Web Information Systems and Technologies, Vienna, Austria, 2019. [Online]. Available: https://doi.org/10.5220/0008559204700476.

[33] C. Kumar, M. Käppel, N. Schützenmeier, P. Eisenhuth, and S. Jablonski, "A Comparative Study for the Selection of Machine Learning Algorithms based on Descriptive Parameters," in *Proceedings of the 8th International Conference on Data Science, Technology and Applications. Volume 1. DATA*, Prague, Czech Republic, 2019, pp. 408-415, doi: 10.5220/0008117404080415.

Francisco José García-Peñalvo

He received the degrees in computing from the University of Salamanca and the University of Valladolid, and a Ph.D. from the University of Salamanca (USAL). He is Full Professor of the Computer Science Department at the University of Salamanca. In addition, he is a Distinguished Professor of the School of Humanities and Education of the Tecnológico de Monterrey, Mexico. Since 2006 he is the head of the GRIAL Research Group GRIAL. He is head of the Consolidated Research Unit of the Junta de Castilla y León (UIC 81). He was Vice-dean of Innovation and New Technologies of the Faculty of Sciences of the USAL between 2004 and 2007 and Vice-Chancellor of Technological Innovation of this University between 2007 and 2009. He is currently the Coordinator of the PhD Programme in Education in the Knowledge Society at USAL. He is a member of IEEE (Education Society and Computer Society) and ACM.

Andrea Vázquez-Ingelmo

Andrea Vázquez-Ingelmo received the bachelor's degree in computer science from the University of Salamanca, Salamanca, in 2016 and the master's degree in computer science from the same university in 2018. She is a member of the Research Group of Interaction and eLearning (GRIAL), where she is pursuing her PhD degree in computer science. Her area of research is related to human-computer interaction, software engineering, data visualization and machine learning applications.

Alicia García-Holgado

She received the degree in Computer Sciences (2011), a M.Sc. in Intelligent Systems (2013) and a Ph.D. (2018) from the University of Salamanca, Spain. She is member of the GRIAL Research Group of the University of Salamanca since 2009. Her main lines of research are related to the development of technological ecosystems for knowledge and learning processes management in heterogeneous contexts, and the gender gap in the technological field. She has participated in many national and international R&D projects. She is a member of IEEE (Women in Engineering, Education Society and Computer Society), ACM (and ACM-W) and AMIT (Spanish Association for Women in Science and Technology).

Jesús Sampedro-Gómez

Jesús Sampedro-Gómez is industrial engineer by the Universidad Politécnica of Madrid. He is also a PhD student by the University of Salamanca and works as data scientist in the Cardiology Department of the University Hospital of Salamanca.

Antonio Sánchez-Puente

Antonio Sánchez Puente, PhD, is a junior researcher from CIBER working at the cardiology department of the University Hospital of Salamanca as a data scientist. He was awarded his doctorate in physics by the University of Valencia for his study of gravity theories before turning his career around the application of artificial intelligence in medicine.

Víctor Vicente-Palacios

Víctor Vicente-Palacios holds a PhD from the University of Salamanca in Statistics and works as a Data Scientist at Philips Healthcare in the area of AI applied to Medicine. He is also an alumnus of the Data Science for Social Good program (University of Chicago) and organizer of PyData Salamanca.

P. Ignacio Dorado-Díaz

P. Ignacio Dorado-Díaz holds a PhD from the University of Salamanca in Statistics. He is currently working as a research coordinator in the Cardiology Department of the Hospital de Salamanca in addition to being a professor at the Universidad Pontificia de Salamanca.

Pedro L. Sánchez

Pedro Luis Sánchez holds a doctorate in medicine from the University of Salamanca. He is currently the head of the Cardiology Department at the University Hospital of Salamanca, in addition to being a professor at the University of Salamanca.

7.36 Appendix AJ. MetaViz - A graphical meta-model instantiator for generating information dashboards and visualizations

ARTICLE IN PRESS

Journal of King Saud University – Computer and Information Sciences xxx (xxxx) xxx

HOSTED BY

ELSEVIER

Contents lists available at ScienceDirect

Journal of King Saud University – Computer and Information Sciences

journal homepage: www.sciencedirect.com

MetaViz – A graphical meta-model instantiator for generating information dashboards and visualizations

Andrea Vázquez-Ingelmo [a,*], Francisco José García-Peñalvo [a], Roberto Therón [b]

[a] GRIAL Research Group, Computer Science Department, University of Salamanca, Spain
[b] VisUSAL, GRIAL Research Group, Computer Science Department, University of Salamanca, Spain

ARTICLE INFO

Article history:
Received 18 February 2022
Revised 20 September 2022
Accepted 22 September 2022
Available online xxxx

Keywords:
Data visualization
Information dashboards
Meta-modeling
Software product line
Model-driven development

ABSTRACT

Objectives: This work presents the application of meta-modeling to the data visualization and dashboards' domain to obtain two main products: a system that allows the instantiation of the meta-model through a graphical interface and a code generator that takes the instantiation of the meta-model as an input to generate visualizations and dashboards.
Methods: A domain engineering approach complemented with an example-driven methodology was iteratively employed to develop the dashboard meta-model. This meta-model was subsequently used as an input to implement a code generator of information dashboards. These two artifacts were finally combined to design and develop the architecture of MetaViz.
Results: Through this process, it was possible to generate visualizations and dashboards using visual elements and basic interactions. MetaViz allows the generation of basic charts (line charts, scatter plots, pie charts, etc.) as well as more complex displays with interactive behavior along different views, layouts, and operations.
Conclusions: The development of MetaViz has served as proof of the viability and benefits of applying these methodologies to a complex domain, but also to set the foundations of a system that allows users to trace every single element from a data visualization to its most primitive values.

1. Introduction

Data visualizations and dashboards are everywhere in (Sarikaya et al., 2018). Unconsciously or not, we are facing these tools constantly in plenty of contexts; watching the news, reading newspapers Field (Perin et al., 2018), and even seeing our favorite sports (Perin et al., 2018). Visualizations help us understand data and grasp knowledge from it, with the ultimate goal of making better decisions –even if they seem trivial, such as if we should go out earlier to catch the bus on time (Kay et al., 2016).

However, as well as visualizations can be truly useful (Álvarez-Arana et al., 2020; Benito-Santos and Sánchez, 2019; García-Sánchez et al., 2019), they can also be misleading and deceptive (Cairo, 2019; O'Brien and Lauer, 2018; Pandey et al., 2015; Correll et al., 2020). Tricky visualizations are not always designed to persuade the audience regarding some topic; sometimes, deceptive visualizations are the product of a lack of understanding of how visual elements (encodings, scales, transformations, etc.) work.

Studying every influential factor and primitive element that make up these powerful communication means is crucial because these research studies not only support the design of more effective visualizations and dashboards but also shed light on the elements that make them effective (Franconeri et al., 2021; Tufte and Graves-Morris, 2014).

When designing visualizations and dashboards, the audience only sees and make use of the finished product, that is, the visualization or dashboard itself. But design decisions and data transformations are also essential to understand the finished product and avoid potential biases (Bedek et al., 2018; Dimara et al., 2020). For example, the intentions of a visualization can be wholly misunderstood if the audience does not know precisely how the data is encoded or the methods employed to obtain the variables displayed (Millet et al., 2020; Ruginski, 2016).

* Corresponding author.
E-mail address: andreavazquez@usal.es (A. Vázquez-Ingelmo).
Peer review under responsibility of King Saud University.

ELSEVIER Production and hosting by Elsevier

https://doi.org/10.1016/j.jksuci.2022.09.015
1319-1578/© 2022 The Author(s). Published by Elsevier B.V. on behalf of King Saud University.

For all these reasons, it is also necessary to give attention to the preliminary steps before developing these tools and communicating them to the audience. This can be accomplished by complementing the final visualization with an explanation of the followed methodology to establish the delivered visualizations, for example, explanations related to how the different variables of the raw dataset are represented in the final product.

Through this work, we aim to bring the methodology as a product and the final visualization or dashboard through a platform that allows users to graphically instantiate information dashboards through visual models that hold the design methodology of the generated tools.

While commercial tools provide compelling assistance during the implementation of information dashboards, they could lack didactic uses due to the automatization of the design processes and thus, a potential decrease in situation awareness (Agnisarman et al., 2019).

The present work explores the following research question: **RQ1.** Which functional and non-functional benefits are derived from the application of domain engineering in the field of data visualization and information dashboards?

This research question is set to provide insights into the benefits of applying the software product line (SPL) and model-driven development (MDD) methodologies, which have been preliminarily researched in previous studies by the authors (Vázquez-Ingelmo et al., 2019; Vázquez-Ingelmo et al., 2018; Vázquez-Ingelmo et al., 2018).

In this sense, this work has two main goals. First, to provide a meta-model of dashboards (and visualizations) containing the most primitive and theoretical elements of these tools and how they influence each other. meta-models are critical artifacts of the model-driven development (MDD) paradigm (Pleuss et al., 2013; Kleppe et al., 2003). The MDD paradigm is focused on the abstraction of information systems' requirements, shifting details from low-level to high-level specifications. In this context, we aim at testing the functional benefits derived from the application of MDD and SPL, specifically, the customization of information dashboards and data visualizations at a low level of detail, but also the traceability of the design decisions made by the users.

And second, to build an instantiator (a dashboard generator tool, namely MetaViz) based on the mentioned meta-model to support the creation of information dashboards through a visual interface, also capturing the methodology by visually modeling these tools before generating them.

This graphical meta-model instantiator can, in turn, enable different usages, as it can be used as a visual tool for generating information dashboards without the necessity of having programming skills, but also as a didactic tool to unravel the elements of these tools and to experiment how different configurations influence the generated product in real-time. To sum up, the main contributions of this work are:

1. The materialization of the dashboard meta-model into a system architecture for a graphical instantiator by making use of two paradigms: model-driven architecture and software product lines.
2. An approach to instantiate information dashboards visually using the identified meta-elements.
3. A web application to automatically design and generate information dashboards and visualizations using fine-grained elements.

The rest of this paper is organized as follows. Section 2 provides the background for this research. Section 3 describes the methodology followed to create both the meta-model and meta-model instantiator. Section 4 outlines the architecture and interface of the meta-model instantiator and integrates the model-driven architecture paradigm's concepts into a functional system. Section 5 provides some usage examples of the meta-model instantiator and a set of generated information dashboards. Finally, sections 6, 7, and 8 discuss the research outcomes and its limitations and present the conclusions derived from this work, respectively.

2. Background

Selecting and designing compelling information visualizations is a challenging step when conveying information. Not only because it is necessary to make crucial design decisions but also because coding these tools is not a trivial task.

There are several solutions to these two issues, specifically those related to visualization recommendations and visual programming tools.

2.1. Visualization recommendation

One of the main challenges that a non-expert user will face when developing a visualization is the lack of expert knowledge. Expert knowledge of this field is necessary to make proper design decisions and produce useful visualizations. But can this expert knowledge be "automated"? Some authors have tried to solve this issue through different methodologies.

One of these methodologies is visual mapping and rules to recommend a visualization based on the target data to be displayed in (Kaur and Owonibi, 2017). This methodology analyzes a dataset and applies different directions based on expert knowledge and the properties of the data variables. Some tools take advantage of this solution to yield recommendations, like Tableau's Show Me (Mackinlay et al., 2007); Manyeyes (Viegas et al., 2007), or Voyager (Wongsuphasawat et al., 2015).

Other tools, like VizDeck (Key et al., 2012), collect feedback from users to learn which visualizations are preferred, subsequently providing the user with similar visualizations.

Context also needs to be considered, including the data domain and the users' experiences with it. For example, visualization ontologies like VISO Field (Viegas et al., 2007) can be used "to represent context and factual knowledge." This way, a ranking process can be executed to yield recommendations regarding suitable visual encodings for a specific context.

User behavior can also be a key factor in selecting the best visualization configuration. User interactions can be logged when exploring data and then analyzed to recommend new data views that align with the users' interests as in (Gotz and Zhou, 2009; Gotz and Wen, 2009).

Finally, artificial intelligence can also be applied to this field by using neural networks (Hu et al., 2019), where a model is trained to predict design choices, or sequence-to-sequence models (Dibia and Demiralp, 2019), in which the generation of data visualizations is seen as a translation problem between data specifications and visualization specifications.

2.2. Data visualization and dashboard tools

Although there are several solutions for automatically recommending data visualizations' configurations, it is also possible to rely on tools that assist their development process. Using these tools, users without coding skills can devote more time to the design decisions instead of the implementation process.

Some commercial tools include Tableau (https://www.tableau.com/), Grafana (https://grafana.com/), Microsoft PowerBI (https://powerbi.microsoft.com/), Scimago Graphica (https://www.graphica.app/), among others. These systems allow users to craft new visualizations through simple visual interfaces; however, they can lack flexibility and fine-grained configuration. Other programming libraries such as D3.js (Bostock et al., 2011) or ggplot (Wickham and Wickham, 2007) or syntaxes like Vega (Satyanarayan et al., 2016) provide more powerful means to develop visualizations.

To overcome these challenges, some authors have developed solutions that fuse the versatility of programming languages with the usability of graphical interfaces. VisComposer (Mei et al., 2018), for example, allows the definition of data visualizations through the connection of different nodes that, in turn, can even hold custom code. Another solution that makes use of nodes and links to generate data visualizations is presented in (Ivanov et al., 2020), and it relies on the grammar of graphics (Wilkinson, 2012), like Vega and ggplot.

Charticulator (Ren et al., 2018) also provides a visual interface to define every element of data visualizations and, in addition, automatically computes the layout of the chart based on the user's selected features.

Our contribution is focused on providing a tool with similar expressiveness by using a model-driven approach, which will allow the fine-grained composition of the elements that are part of the system, as explained in the following sections. In this sense, we aim to assist the design process of information dashboards and data visualization while raising awareness and understanding of the elements being displayed on the screen.

While existing tools focus on efficiency and performance when generating information dashboards and data visualizations, this work focus on how to provide better means to understand how data visualizations are developed.

3. Methodology

Different methodologies have been employed to develop the meta-model and the generative pipeline that drive the architecture and interface of MetaViz. These are the steps followed in carrying out this research:

1. A domain engineering approach complemented with an example-driven methodology was iteratively employed to develop the dashboard meta-model (Vázquez-Ingelmo et al., 2019; Vázquez Ingelmo et al., 2021; Vázquez-Ingelmo et al., 2306; Vázquez-Ingelmo et al., 2020; Vázquez-Ingelmo et al., 2019; Vázquez Ingelmo et al., 2019)
2. The meta-model was subsequently used as an input to implement a code generator of information dashboards (Vázquez-Ingelmo et al., 2019; Vázquez-Ingelmo et al., 2020; Vázquez-Ingelmo and Therón, 2020)
3. The meta-model and the dashboard generator were finally combined to design and develop the architecture of MetaViz

3.1. Meta-modeling

Meta-models are the backbone of the model-driven development (MDD) paradigm (Pleuss et al., 2013; Kleppe et al., 2003), which focuses on abstracting low-level details of information systems to obtain higher-level specifications.

The abstraction of specific characteristics provides the means to design generic schemas of information systems, focusing on generic commonalities instead of low-level implementation details.

One of the main benefits of this methodology is the increase in reusability, which often translates to faster development of systems belonging to the same domain. Reusability is not only focused on software components; meta-modeling also supports the reusability of knowledge because the features identified within the systems' domain can evolve to obtain better specifications.

The MDD approach can be implemented through the model-driven architecture (MDA). MDA is a guideline proposed by the Object Management Group (OMG) and provides an architecture for model-driven software development by using descriptions of the target system (Álvarez et al., 2001). The OMG proposal also determines a set of standards to develop the approach, such as meta-object facility (MOF), unified modeling language (UML), XML (Extensible Markup Language), and metadata interchange (XMI), and query/view/transformation (QVT).

The MDA framework is composed of four layers, in which each layer represents different levels of abstraction. The most abstract layer (M3 level) is the meta–meta-model level. This layer defines basic structures and concepts to represent more specific layers as well as itself, and it can be implemented with the mentioned MOF standard or Ecore. The M2 level, namely the meta-model level, complies with the meta–meta-model and represents abstract and technological-independent entities and relationships. The M1 level defines models that instantiate and configure the abstract features contained in the meta-model. The syntax of M1 models must comply with the M2 level. Finally, the M0 level represents real-world applications based on a previously defined M1 model, which, in this case, are the implemented information dashboards. Fig. 1 summarizes these levels in the dashboards domain.

We propose a dashboard meta-model framed within the MDA paradigm (Álvarez et al., 2001). The first version of the dashboard meta-model was an instance of MOF. However, it was finally transformed into an instance of Ecore using Graphical Modelling for Ecore included in Eclipse Modeling Framework (EMF).

Fig. 2 shows an excerpt of the current version of the dashboard meta-model, specifically the section related to the specification of the primitive elements of data visualizations, which is the focus of this work. For example, different essential elements of data visualizations are contained in the displayed section of the dashboard meta-model. Visualizations are composed of visual marks, which are abstracted through the *Mark* entity in the meta-model. These marks can be of different types (individual points, series, geographical, etc.) and shapes (rectangles, circles, lines, areas, etc.). The visual properties in charge of encoding the data variables' values are the *Channels; each* visual mark can have different encoding channels (position, size, color, etc.) depending on the visualization built. These channels represent values thanks to the *Scales*, which map the raw values from the datasets into proper encoding values associated with the channels. Other common elements identified through meta-modeling are visual aids such as axes, legends, labels, and interactivity.

This dashboard meta-model has been subject to several modifications and improvements by analyzing the domain. All the changes and contributions can be consulted in (Vázquez Ingelmo et al., 2021; Vázquez-Ingelmo et al., 2306; Vázquez-Ingelmo et al., 2020; Vázquez-Ingelmo et al., 2019; Vázquez Ingelmo et al., 2019). Although other dashboard meta-models exist in the literature (Morgan et al., 2018; I. Logre et al., 2007), they do not account for fine-grained features, which are crucial in this domain, as a slight modification in the design of a dashboard or data visualization could lead to significantly different insights (Correll et al., 2020; Pandey et al., 2014; Franconeri et al., 2021).

The features of the meta-model have been identified through domain engineering, the review of visualization grammars (Satyanarayan et al., 2016; Wilkinson, 2012; Harsu, 2002), and a literature review (Vázquez-Ingelmo et al., 2019). The review (Vázquez-Ingelmo et al., 2019) enabled the identification of gaps in the field of tailored information dashboards. Specifically, these kind of solutions lacked powerful and fine-grained specifications to account for the different features that dashboards and data visualizations could have.

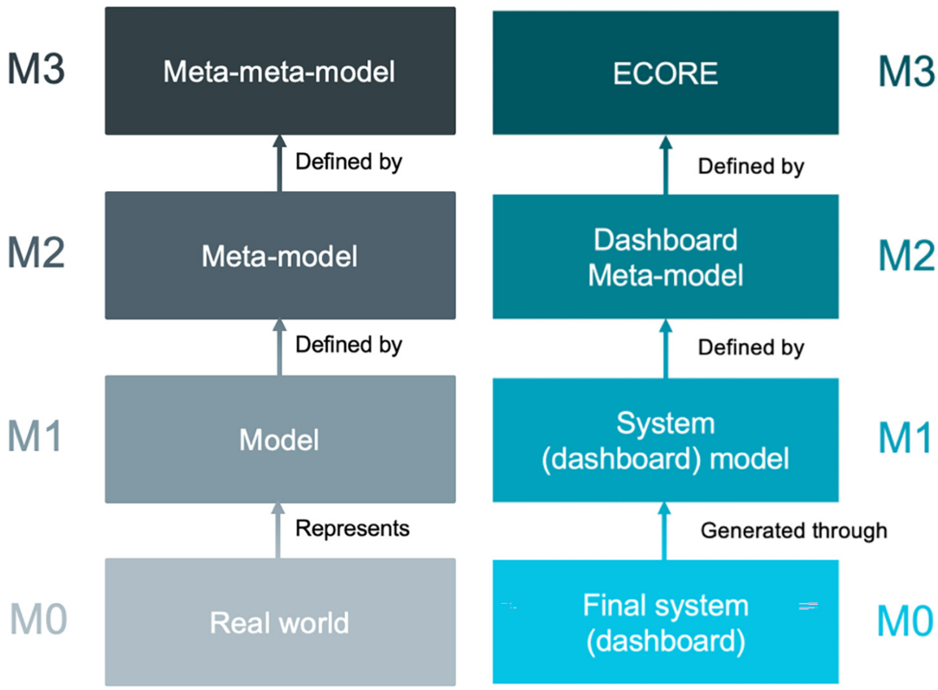


Fig. 1. The four-layer architecture proposed by the OMG contextualized in the dashboards' domain.

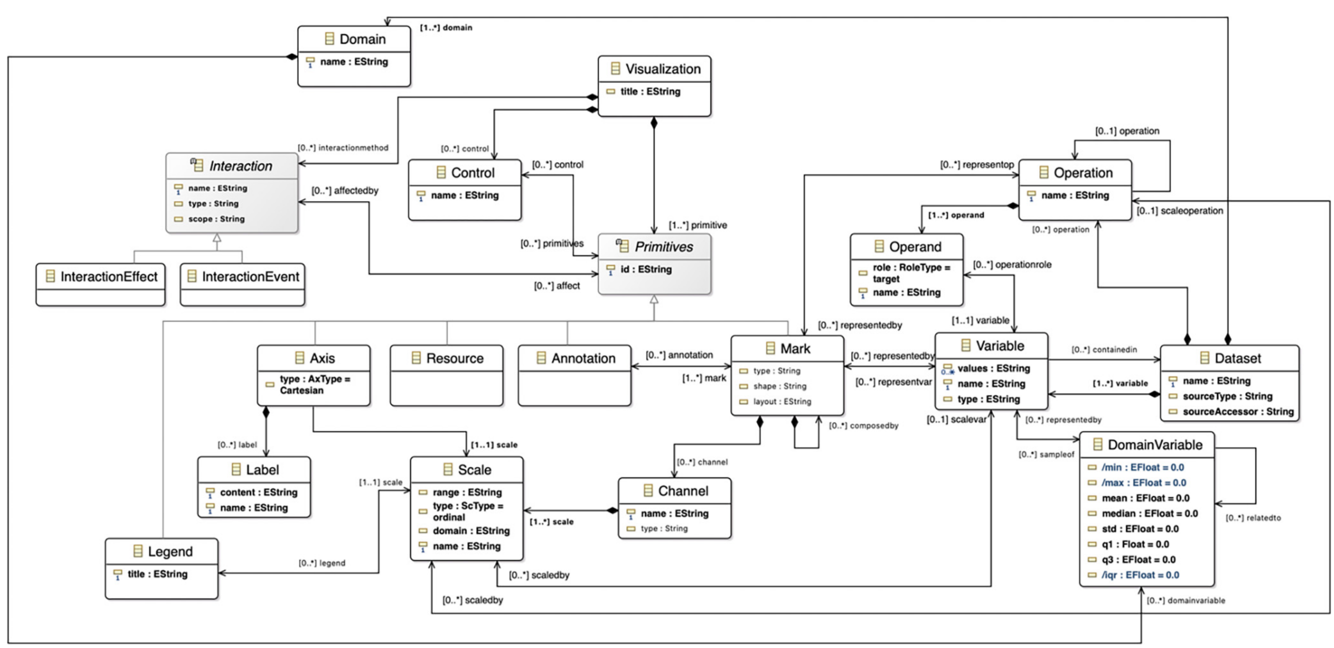


Fig. 2. Visualization specification section of the dashboard meta-model. Source (Vázquez Ingelmo et al., 2021). (Available at: https://doi.org/10.5281/zenodo.5788981).

In addition, the literature review (Vázquez-Ingelmo et al., 2019) drove the decision to use model-driven development over other methodologies, as meta-models can tackle the fine-grained features found in the dashboards domain. The complete version of the meta-model can be consulted at: https://doi.org/10.5281/zenodo.5788981.

Other sections of the dashboard meta-model contain the specification of user characteristics, layouts, goals, supported tasks, or data domain. The complete discussion on these sections can be found in previous works by the authors' (Vázquez-Ingelmo et al., 2022).

The meta-model factorizes the elements of dashboards into low-level features and the relationships among them. Each meta-class references a primitive piece, tangible or theoretical, because not only real elements (visual marks, axes, legends, etc.) are crucial for the dashboard design, but also non-tangible concepts such as interactions, supported tasks, or the profile of the users that will use the dashboard.

This meta-model has been validated using a content validation by experts (Escobar-Pérez and Martínez, 2008), in which the coherence, relevance, and clarity of the meta-model entities were scored in different sections of the meta-model, obtaining promising results (Vázquez-Ingelmo et al., 2022).

3.2. Software product lines

As the previous section explains, dashboards can be divided into abstract features to obtain a generic structure for building these tools. However, it is necessary to rely on other paradigms to develop specific and functional systems from meta-models. A suitable approach to tackle the materialization of the identified features is the software product line (SPL) paradigm because it allows planning of how different core assets and components will be built to ease the development processes of products belonging to the same domain or family (Clements and Northrop, 2002; Pohl et al., 2005).

By reutilizing, configuring, and composing these software assets, the time-to-market of new derived products decreases, in addition to an increase in requirements traceability, customization levels, flexibility, maintainability, and productivity.

However, implementing this paradigm is not straightforward because it is necessary to develop the core assets to support the variability of features identified during the domain engineering phase. Planning the development of highly configurable software components allows the delay of design decisions, enhancing flexibility regarding the materialization of dynamic or new requirements. These delayed design decisions are named variability points (Van Gurp et al., 2001), and in this work, they are implemented through code templates (Vázquez-Ingelmo et al., 2019).

3.3. Combining meta-modeling and software product lines

The approach taken to design and implement the dashboard generator is based on a combination of the latest methodologies. The meta-model provides the conceptual representation of information dashboards and data visualizations, while the SPL tackles the transformation of these abstract entities into core assets that can be composed. In this sense, the combination of meta-modeling to obtain a domain abstraction with the SPL philosophy of systematically reusing assets provides a robust framework for building families of products.

Specifically, this approach also drove the decision to use code templates as the method to materialize the variability points of the SPL. Code templates were selected given their suitability in this context and resemblance to meta-modeling in the philosophy of encapsulating and decomposing complex entities into primitive elements. The meta-model entities were matched with code fragments, allowing the development of the dashboard generator.

3.4. Code generation

The generation process leverages the SPL paradigm with the meta-modeling approach to map abstract entities into code fragments. A Python-based generator is used to read input configuration files and inject the concrete dashboard features into Jinja2 code templates (https://jinja.palletsprojects.com/en/3.0.x/). This process results from a set of JavaScript and HTML files that render a specific dashboard and visualization configuration.

The approach to creating the generative pipeline is based on the idea of matching the meta-model entities and relationships with source code fragments. These fragments are, in turn, encapsulated into macros that can be conditionally included to render specific dashboard features. Code fragments are glued together through the Python generator that generates the final code considering the input model (which is, in turn, an instance of the meta-model).

4. The MetaViz system

4.1. Architecture

The architecture of MetaViz relies on the previously described MDA paradigm's concepts. The MDA layers (specifically the M2, M1, and M0 layers) are materialized through different software modules. The reason why the M3 layer is excluded from the final system's architecture is that the meta-model was built as a standalone artifact using Ecore (that is, it is an instance of the M3 layer as explained in the methodology section) but directly integrated as a set of rules and constraints into the codebase of MetaViz's interface.

Fig. 3 outlines the architecture of MetaViz. The backend is developed using the Django framework (https://www.djangoproject.com/), and it is divided into four main modules:

1. **Workspace module.** This module acts as an Application Programming Interface (API), and it oversees receiving the current dashboard configuration crafted through the user interface to send the information to other modules.
2. **Recommendation module.** This module is still under development, but it will implement a rule engine based on the meta-model rules and heuristics to yield recommended dashboard configurations using the current and previous ones.
3. **Dashboard management module.** The dashboard management module has two main goals: to store and retrieve the last dashboard configuration persistently and to format it to feed the dashboard generator module.
4. **Dashboard generator module.** The last module implements a dashboard generator based on Jinja2 code templates (Vázquez-Ingelmo et al., 2019).

4.2. Graphical user interface

Not only the software architecture is based on the MDA layers, but also on the user interface. Fig. 4 shows the correspondence between the different layers of MDA and the interface components of MetaViz.

The toolbar (M2) enables users to add new primitive elements to the workspace (M1). The configuration parameters of each element also rely on the meta-model attributes and relationships, as seen in the dialogue box in Fig. 4.

The workspace (M1) allows the instantiation of new information dashboards by connecting and configuring the different "meta-elements." Finally, the canvas (M0) shows the generated dashboard or visualizations in real time.

The workspace elements that can be included are listed and detailed below.

1. **Dataset.** It matches the *Dataset* and *Variable* elements from the meta-model. The interface provides a table to visualize the raw dataset.
   - **Filters.** A filter is an operation that can be applied to the dataset to yield a subset of rows that match a specific condition. Although this element matches the *Operation* class of the meta-model, it was decided to separate it as an element in the interface to make the active filters easier to spot and manipulate.
2. Operations. Operations are transformations that can be applied to the input dataset. In this case, it is possible to aggregate data (to obtain the mean, the count, the standard deviation, etc. of a

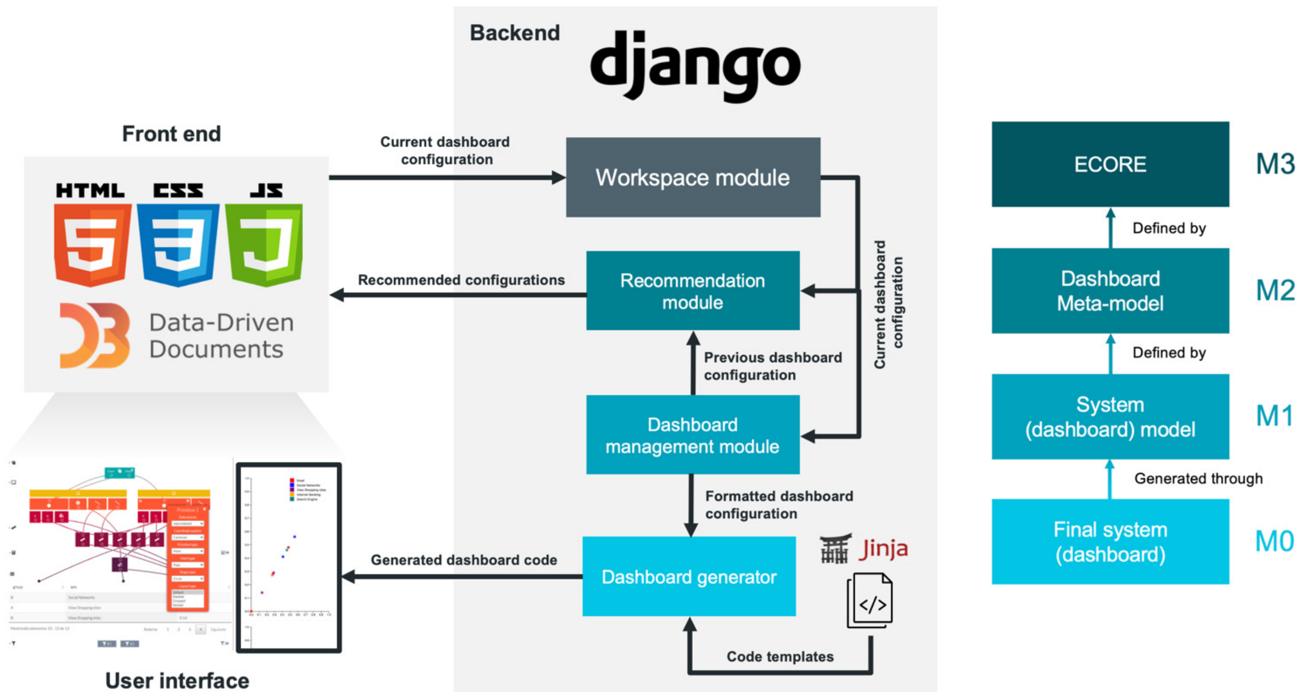


Fig. 3. MetaViz architecture and connections among its different modules. The right section of the figure shows the MDA architecture layers adapted to this context. The color of each module matches the MDA layer that it addresses.

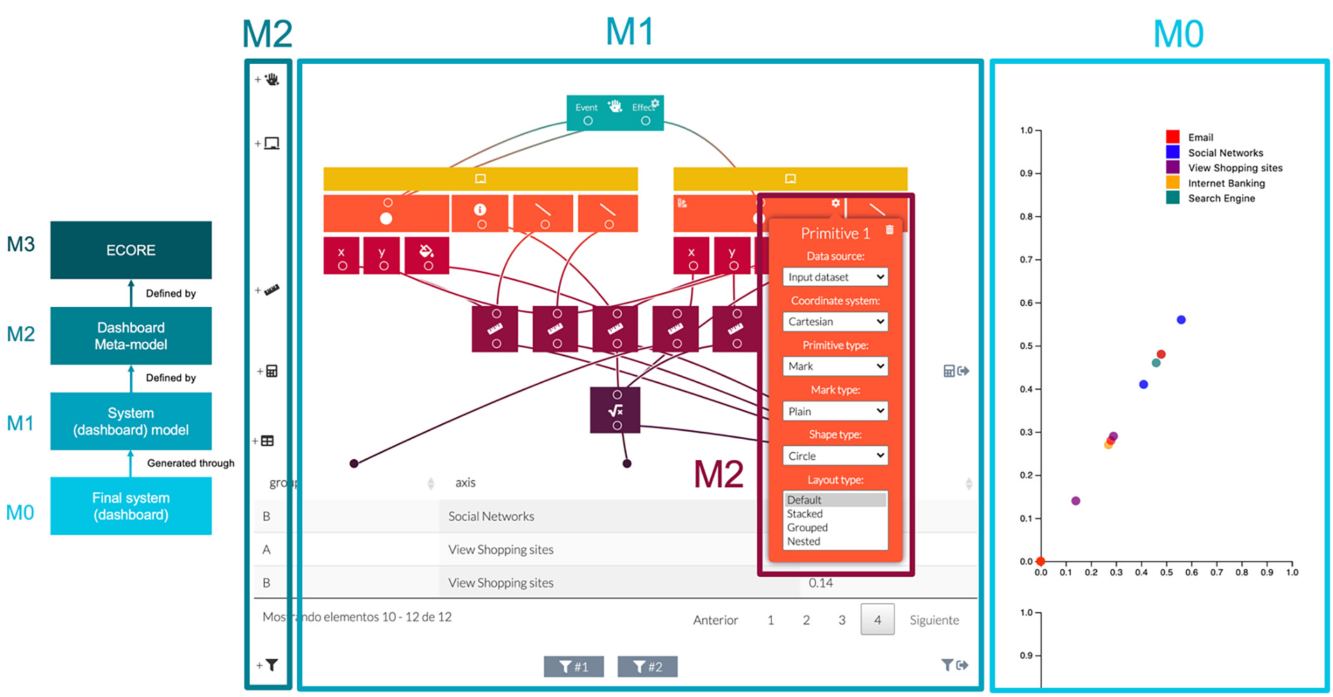


Fig. 4. MetaViz interface. The left section of the figure shows the MDA architecture layers adapted to this context. The color of each container matches the MDA layer that it addresses.

column) or perform operations among columns or scalars (subtractions, additions, multiplications, divisions, etc.). As its name implies, this interface element matches the meta-model's Operation class.

3. Scales. Scales are the "transition" between the data and visual spaces. They allow the materialization of data into visual elements. They match the Scale element in the meta-model.
4. Visualizations. Visualizations are "containers" that can hold different primitives (as seen in the meta-model's Visualization meta-class).

- Primitives. This concept is employed to identify the elements that can be included in a visualization. The meta-model elements are Marks, Legends, Axes, Resources, and Annotations. The system now enables users to have the first three elements.

- (Visual) Channels or Encodings. Suppose a primitive is identified as a Mark. In that case, it is possible to add different visual channels or encodings (Channel class in the meta-model), which are the elements that ultimately encode (Wilke, 2019) the input data into visual marks' characteristics (position, size, color, stroke color, text, etc.).

5. Interactive behavior. The dashboard meta-model also captures interactive behavior. The support of user interactions can be crucial to making dashboards and visualization more effective and engaging. This interface element represents the meta-class Interaction. It is possible to configure the trigger event (such as hovering over a primitive) and the interaction's effect (such as highlighting the target primitive or showing a tooltip). Interactions, as specified in the meta-model, affect the primitives of visualizations (and not the visualization as a whole), supporting a fine-grained specification of interactive behaviors (Vázquez-Ingelmo et al., 2020). At this moment, the available events that trigger interactions are "hovering" and "clicking," while the potential available effects are "highlight" and "show tooltip."

5. Results

This section provides examples of the instantiation process, including the final generated visualizations and their model. Fig. 5 provides the instantiation workflow, in which the "tangible" elements (the visual marks, axes, or legends) are inspected first to see if they need to be rendered. If not, the generation process stops and waits for the user to add new elements to the model or to change any other configuration. The platform is accessible at https://metaviz.grial.eu, and its tutorial can be consulted at https://bit.ly/3NnSGdb.

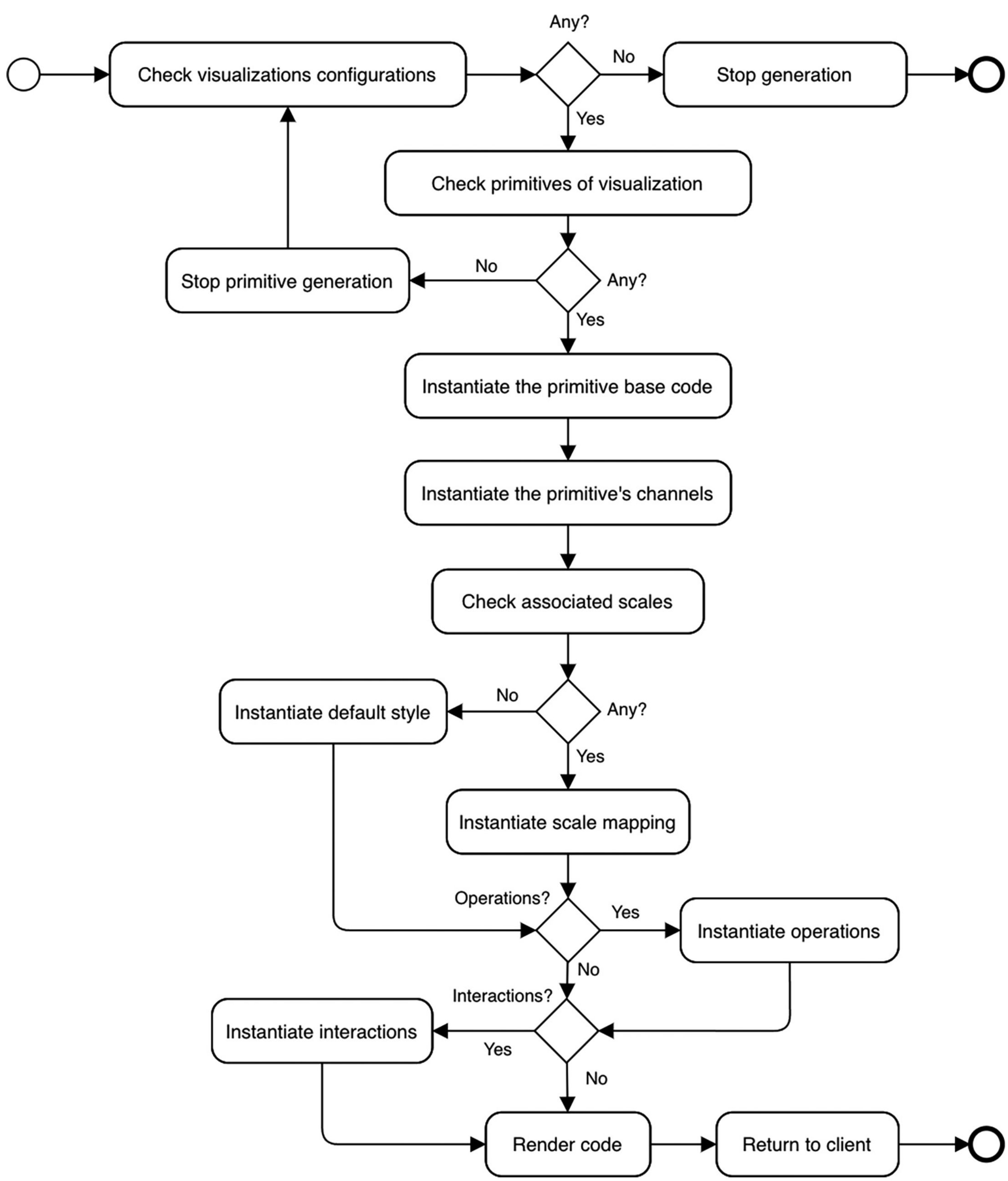


Fig. 5. Business Process Model of the instantiation workflow.

5.1. Basic visualizations

MetaViz allows the instantiation of different data visualizations. First, it is necessary to upload a dataset (for example, a CSV file). The system will show the uploaded dataset as a data table, as seen in Fig. 5. Once the dataset is uploaded, the user can start the instantiation process, which is manual, and the system will automatically generate the visualization that matches the user's configuration. In this case, a simple dot plot has been created using a visual mark with a circle shape, which is positioned through the X and Y channels.

These channels encode the "Date" and "High" columns of the dataset through a time and linear scale, respectively. The model also includes two axes to ease the interpretation of the visualization. The result of this model is shown at the rightmost part of Fig. 6.

The mark type can be easily changed (into a series mark) and the shape type (to a line) by configuring the primitive. The result of this slight change is shown in Fig. 7. Different primitives can also be combined within the same visualization. In Fig. 8, it is possible to see the combination of the last two primitives and their results (at the right of the image).

It is also possible to create additional visualizations and to add new encodings to represent the values of a variable through the color channel (in this case, by using a linear scale that goes from red to black), as is shown in Fig. 9.

5.2. Interactive behavior

MetaViz also allows the instantiation of interactive behavior. To add an interactive behavior, it is necessary to select the trigger event and the desired effect and to connect the primitives involved to the "Event" (the primitive to which the event will be implemented) and the "Effect" (the primitives that will be affected by the event).

Fig. 10 shows the event triggered when hovering over the circle marks in the first visualization and its effect on the connected primitives. In this case, a tooltip is shown beside the circle marks on the first visualization and the circle marks on the second visualization, which matches the model. The following video is available for a more detailed demonstration of the interactive behavior: https://bit.ly/3sowVTV.

5.3. Filters & operations

Filters can also be specified at the primitive level; some primitives could represent data from filtered datasets with different conditions. A filter string can be specified with logical operators to different chain conditions. Fig. 11 highlights the results of applying various operations:

1. On the bottom, a filter that affects the "High" column of the data set has been applied to the circle mark of the first visualization. The result of using the filter can be seen at the top right of Fig. 10: marks that encode values less or equal to 7300 are not rendered.
2. On the other hand, operations can also be included in the workspace. In this case, the mean value of the "High" variable was obtained from the dataset and encoded the value through a strip mark on the Y-axis.

5.4. Layouts

The last examples show how to instantiate basic visualizations through the meta-model. However, users cannot always rely on basic layouts to convey data; sometimes, it is necessary to arrange the visual marks following a specific layout (for example, stacks,

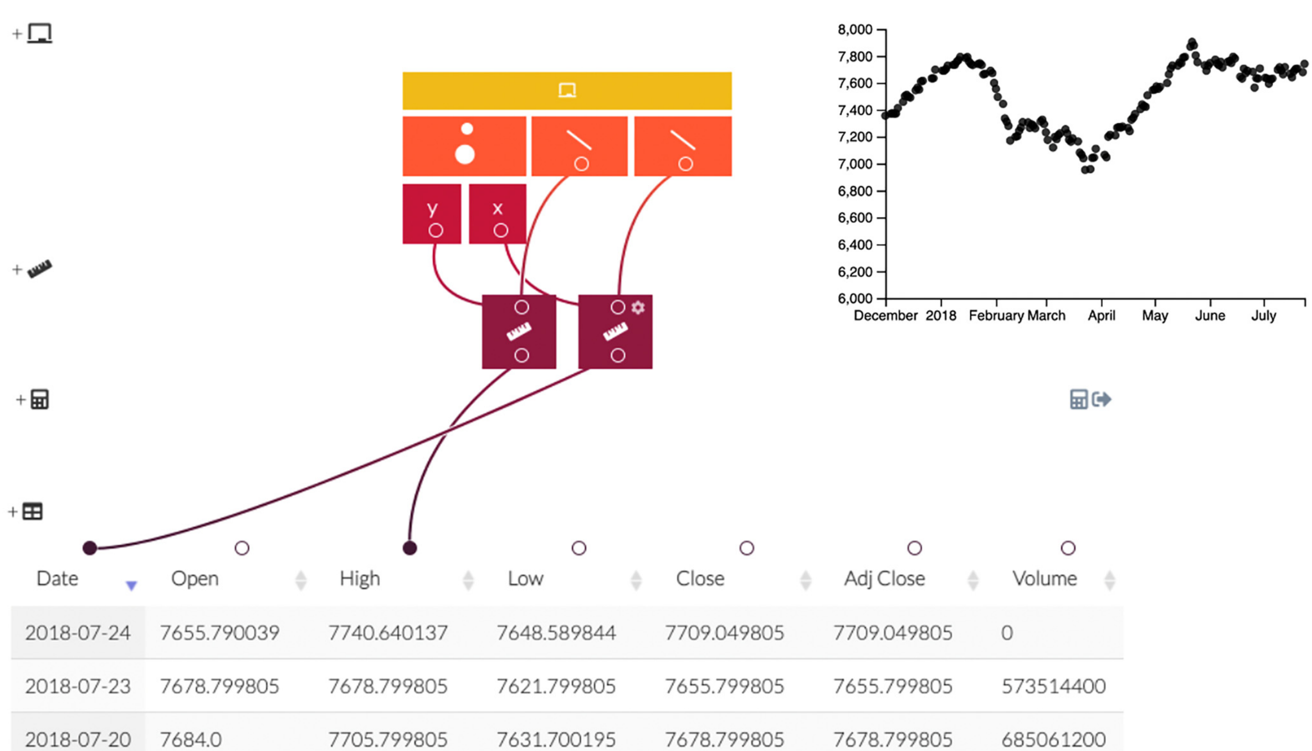


Fig. 6. A generated dot plot and its model.

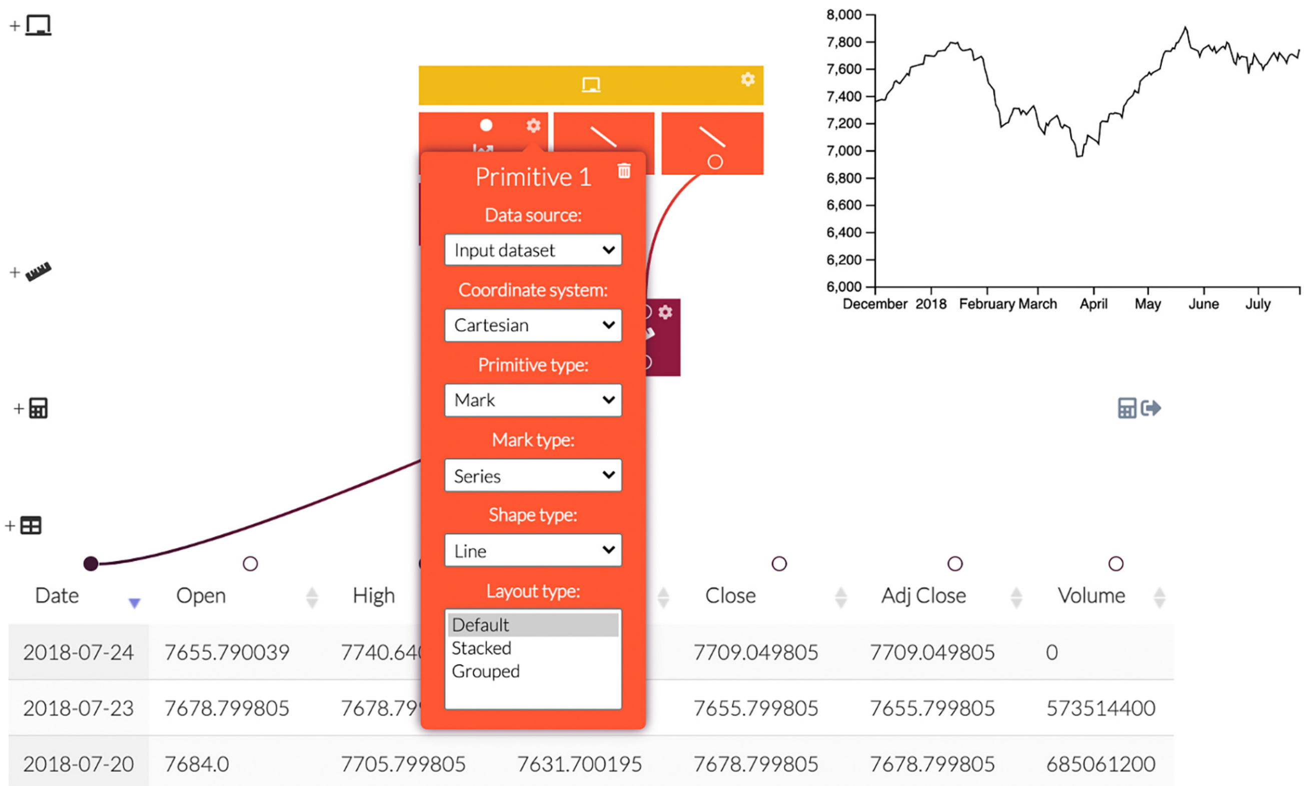


Fig. 7. A generated line chart and its configuration.

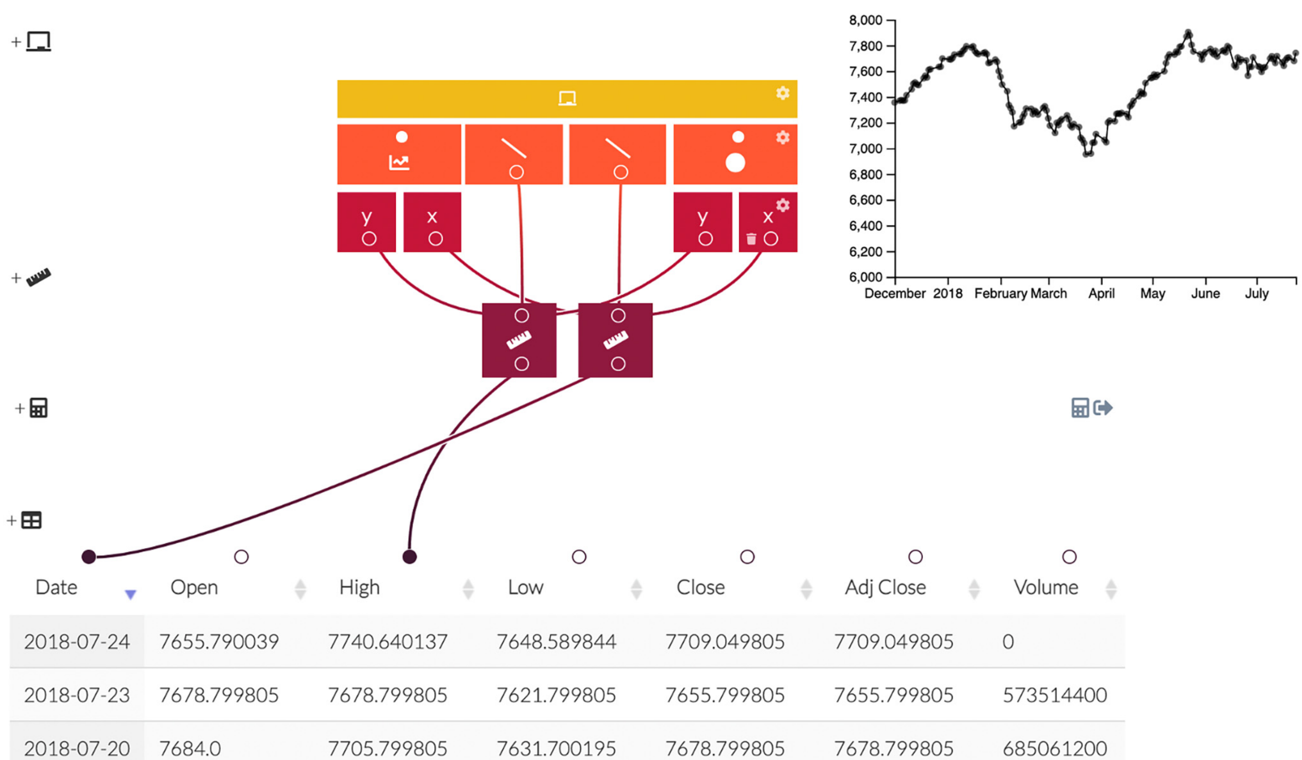


Fig. 8. Combination of primitives on the same visualization.

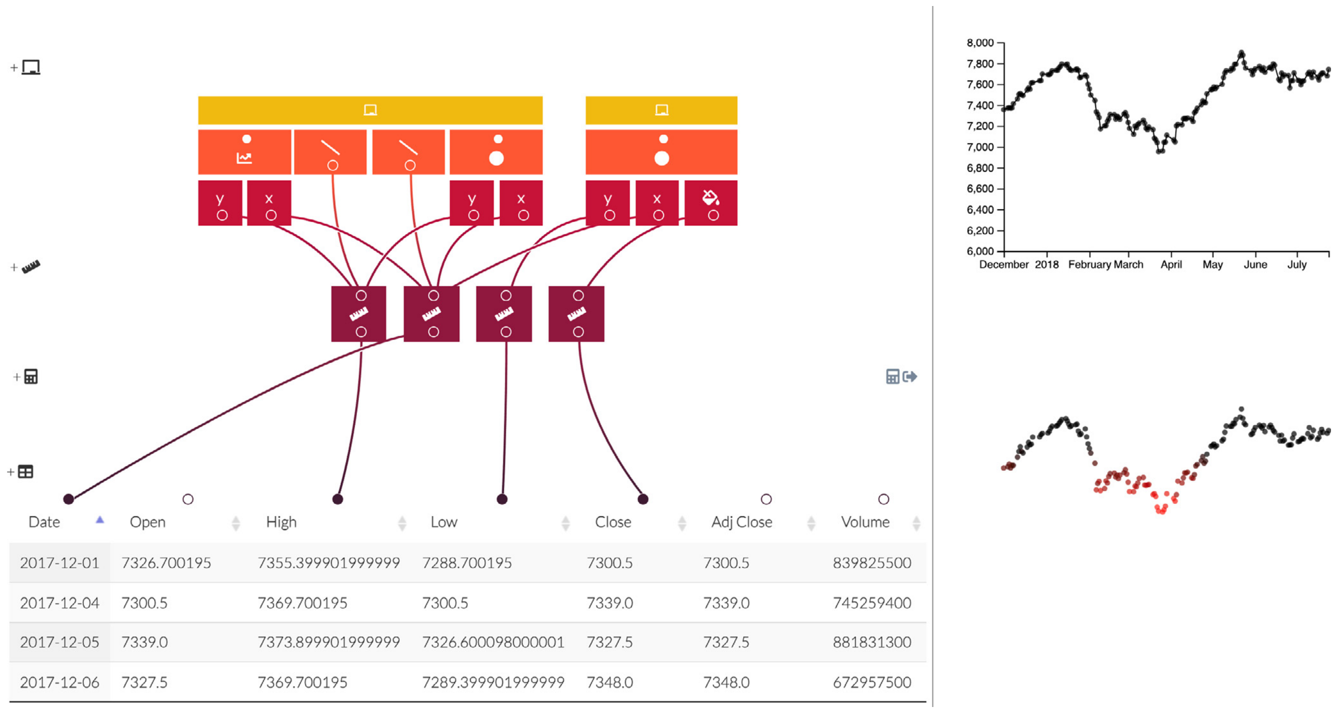


Fig. 9. A generated dashboard with two visualizations.

groupings, or hierarchies). MetaViz also supports the definition of these layouts, as shown in Fig. 12 and Fig. 13. It is necessary to specify additional information to instantiate these kinds of visualizations, for example, which variable will represent the size or "levels" of a hierarchy (Fig. 12), or which variables will compute the stacks in a stacked bar chart (Fig. 13).

6. Discussion

This research has proved the viability of meta-modeling not only to obtain an artifact that can serve as a framework to design data visualizations but also to design a whole functional dashboard

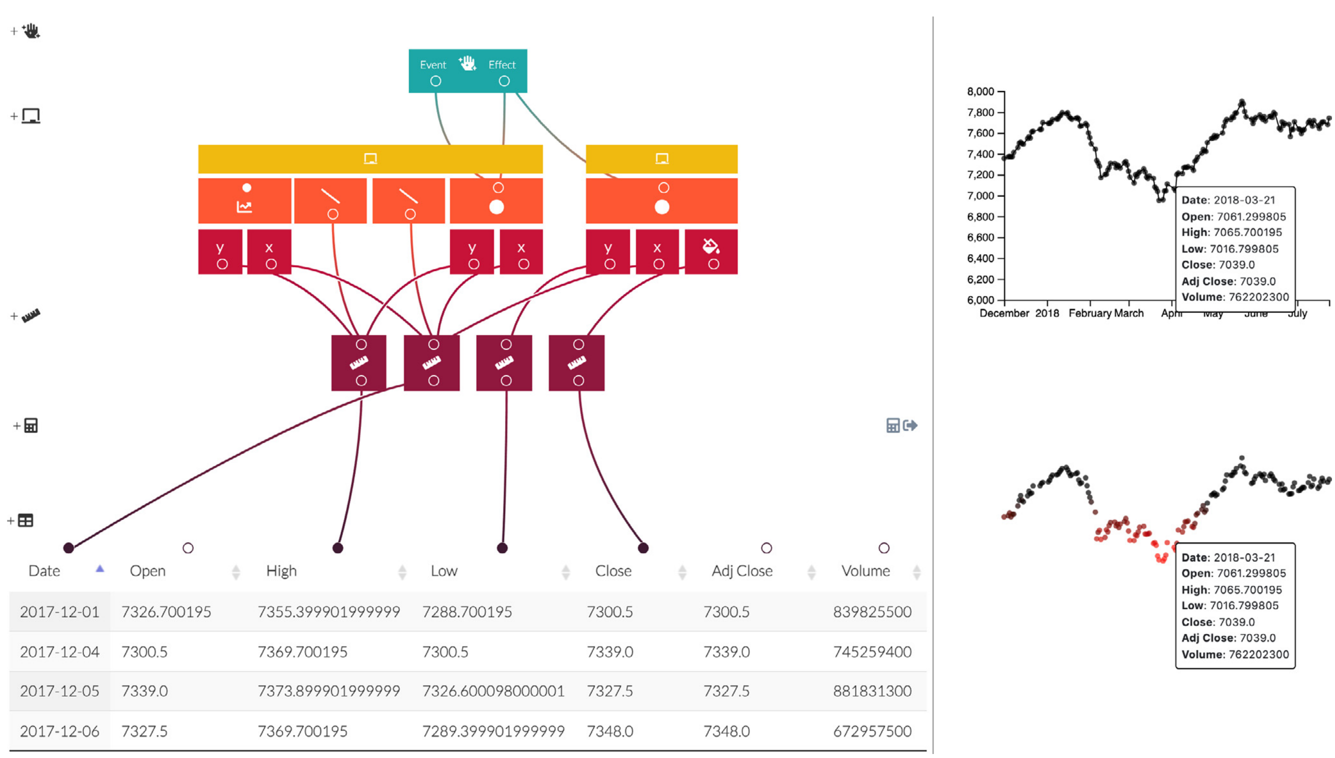


Fig. 10. Interactive behavior between two generated data visualizations.

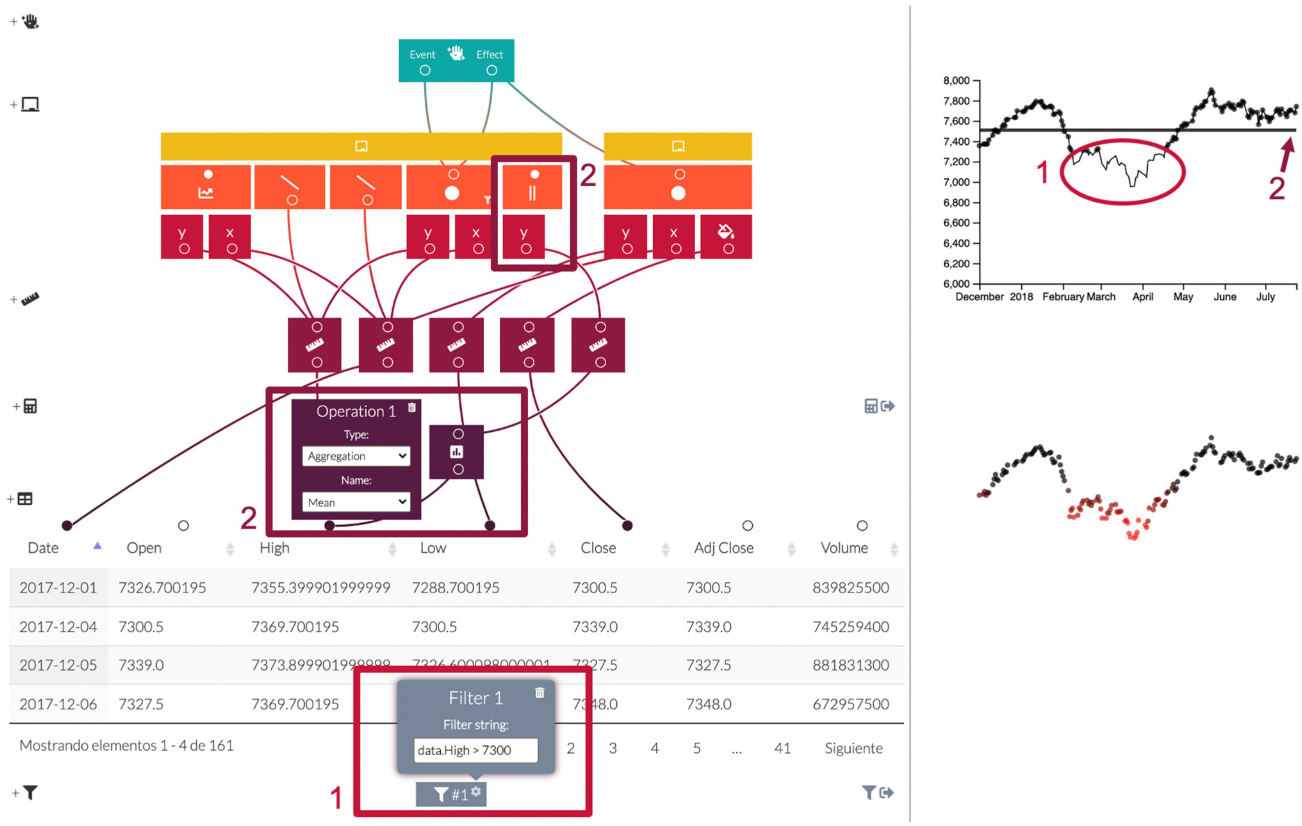


Fig. 11. Result of the combination of filters and operations on the first generated visualization.

generation system and user interface entirely driven by the proposed meta-model.

Following the presented methodology, it has been possible to identify essential elements of data visualizations and dashboards and arrange them into a graphical interface.

Applying the model-driven approach has yielded two main benefits. First, because this paradigm supports the evolution of the meta-model, it also eases the evolution of the platform's architecture and the generative process, which can be easily modified thanks to the organization of the different components. Second is the possibility of tracing every step taken to obtain the final product, from the conceptual model to the final dashboard.

This second benefit is beneficial in a domain as complex as conveying data visually. Due to the rise of fake news and its misleading communication of data and information, knowing precisely the transformations that the source data have suffered to obtain the visualization, it's finally displayed to the audience makes these tools more transparent. For example, sharing the whole workspace and letting users explore the final visualization and the model could lead to better comprehending the data being displayed. On the other hand, sharing the model could ease spotting underlying design issues in data visualizations.

Also, understanding the lower-level elements of data visualizations and instantiating them explicitly could minimize the risks of introducing biases during the design of data visualizations. Still, this notion will be explored and analyzed in future works.

The fact that the meta-model not only captures the primitive elements of visualizations but also the methodology behind their composition (as it includes the user, the data domain, data transformations, goals, and tasks) enables the materialization of both theoretical and tangible elements from this domain into a system model.

Meta-modeling also enables the possibility of including a visualization recommendation layer by adding expert knowledge and heuristics through Object Constraint Language (OCL) rules (Richters and Gogolla, 2002). These rules can be implemented directly into the meta-model itself, providing good practices to design visualizations and dashboards directly at the M2 layer of the MDA paradigm. In addition, the user entity, the dashboard goals, supported tasks, and the data domain (the other sections of the dashboard meta-model, as explained in the methodology) will be added to the system to create a more powerful tool in which these relevant factors are also considered to assist the design and modeling phase.

One interesting aspect that can be derived from this research is the correlation between the connections and elements of the data visualization model and the complexity of the generated visualization. In the end, the system model (located in the M1 layer of the MDA paradigm) is represented through nodes and links with different characteristics. Complex visualizations, understood as those that obscure the visualization goals by adding unnecessary elements, could be identified at the modeling phase by measuring the number of nodes, their type, and the transformations that data has suffered before the visualization is generated.

With this information, users could vary the complexity of the visualization to adapt it to the target audience and the tasks that the visualization is set to support. MetaViz can be used as a didactic tool to teach the components of data visualizations and dashboards and to understand how different configurations affect the final product.

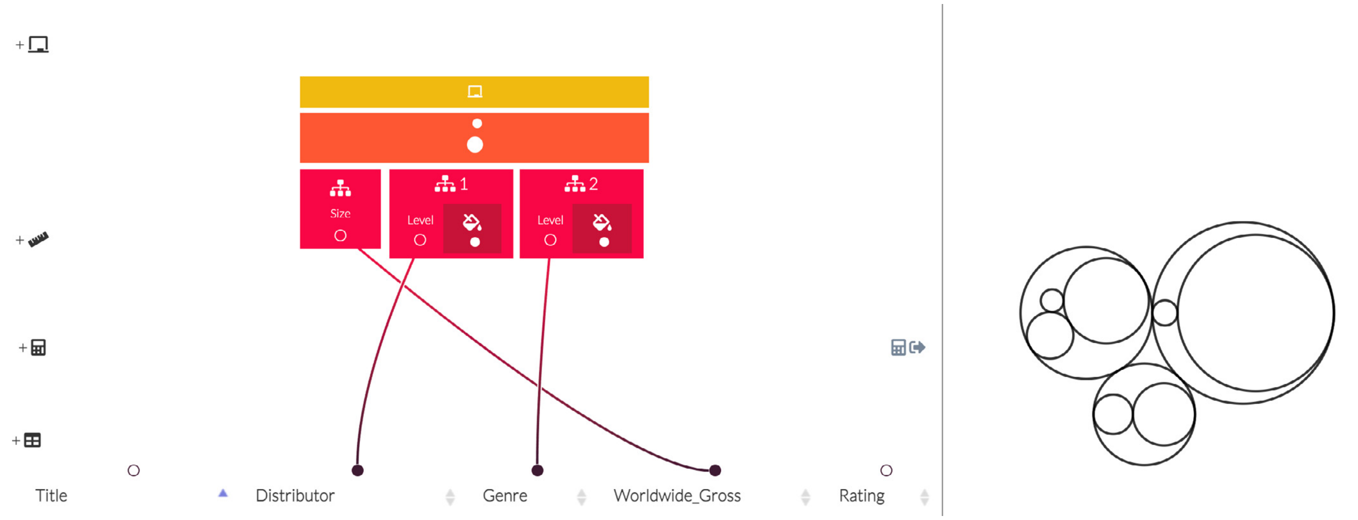


Fig. 12. Hierarchy layout with two nesting levels codified through circle visual marks.

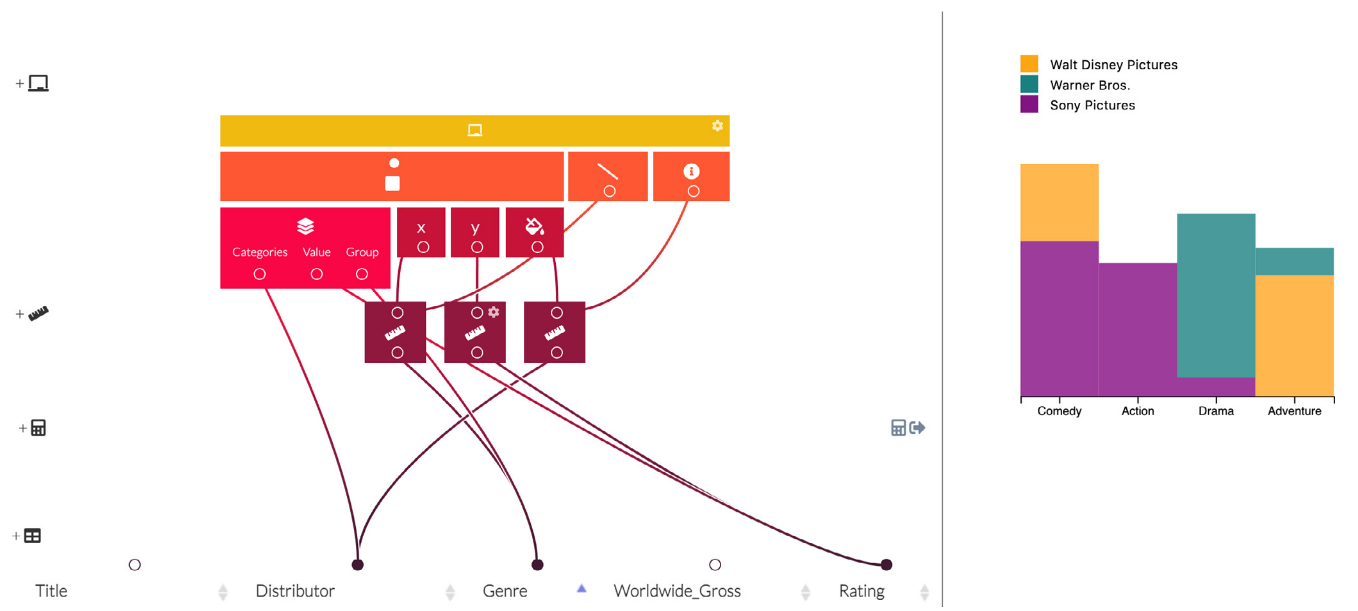


Fig. 13. Stack layout applied to a bar chart.

Following these results and the research question posed at the beginning of this work, it is possible to affirm that the use of meta-modeling and the SPL paradigm has been helpful to increment the benefits related to some functional and non-functional features of information dashboards. Specifically, the reduction of development times (as dashboards are automatically generated), the traceability of the requirements (as they are stored as models that can be visually consulted using MetaViz), as well as the potential to define and configure functional features at a fine-grained level. In this sense, this study provides proof of the viability of applying these two methodologies in a domain with fine-grained and low-level features.

Finally, this research has provided a platform to graphically instantiate the OMG's M1 and M0 levels. This opens the possibility of generating dashboards automatically (like other commercial tools like Tableau or Microsoft Excel) and tracing and storing the models of the developed products, fostering more transparent design and development processes.

To sum up, this platform can be employed to teach data visualizations and improve the understanding of these useful tools by visualizing and tracing every design decision (and its implications in the final product) in real-time.

7. Limitations

One of the main limitations is that automation comes with a cost detriment with some design features or more complex possibilities that a declarative language like D3.js could achieve by itself. However, MetaViz proposes a set of fine-grained elements that can generate pretty expressive visualizations without the necessity of programming skills.

Another limitation of this first version of MetaViz is that the meta-model is tightly coupled to the system's codebase. This means that any change to the meta-model requires manual modifications to the codebase. Although it is not a critical issue, connecting the meta-model (in Ecore format) and the codebase would make the code much more scalable and maintainable.

Regarding the interface, the connections between the nodes could be overwhelming when more than three visualizations are instantiated simultaneously. The current version of the platform allows scrolling the different sections of the interface. Still, some mechanisms such as highlighting certain links or zooming in to see relationships in detail could be an improvement for the usability of the system. However, this will be further explored in future usability tests.

8. Conclusions

This work presents a web application developed using model-driven development (MDD) and software product line paradigms (SPL) to generate information visualizations and dashboards.

A meta-model has been developed to capture the primary elements that make up these tools. It has served as a framework to implement generation workflow and a graphical instantiator. Both the generation process of these tools and the whole system architecture is guided by the mentioned paradigms. An instantiation workflow to use the meta-model template to create data visualizations and dashboards was also provided.

This has served as proof of the viability and benefits of applying these methodologies to a complex domain, but also to set the foundations of a system that allows users to trace every single element from data visualizations to its most primitive values (that is, raw data).

The MetaViz system provides support to assist and capture the methodology of the whole instantiation process, from the meta-model to the final generated product. In addition, the visual programming interface enables the user to obtain data visualizations without requiring coding skills.

The benefits derived from the application of this approach include the support for the extensibility of the system (which is a benefit related to MDD and SPL paradigms), the integration of heuristics through formal languages like OCL, and the traceability of each primitive element through the different OMG's MDA levels (fostering transparency in the visualizations and dashboards design process).

Future research lines will be focused on testing the proposed platform, extending its functionalities, including heuristics support through OCL, and researching its potential uses as a didactic tool.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

This research work has been supported by the Spanish Ministry of Education and Vocational Training under an FPU fellowship (FPU17/03276). This research was partially funded by the Ministry of Science and Innovation through the AVisSA project grant number (PID2020-118345RB-I00).

References

Agnisarman, S., Lopes, S., Madathil, K.C., Piratla, K., Gramopadhye, A., 2019. A survey of automation-enabled human-in-the-loop systems for infrastructure visual inspection. Autom. Constr. 97, 52–76.

J. M. Álvarez, A. Evans, and P. Sammut, "Mapping between Levels in the Metamodel Architecture," in *≪UML≫ 2001 — The Unified Modeling Language. Modeling Languages, Concepts, and Tools. UML 2001. Lecture Notes in Computer Science*, vol. 2185, M. Gogolla and C. Kobryn Eds. Berlin, Heidelberg: Springer, 2001, pp. 34-46.

A. Álvarez-Arana, M. Villamañe-Gironés, and M. Larrañaga-Olagaray, "Improving Assessment Using Visual Learning Analytics," *Education in the Knowledge Society,* vol. 21, Article 9, 2020, doi: https://doi.org/10.14201/eks.21554.

Bedek, M.A., Nussbaumer, A., Huszar, L., Albert, D., 2018. Methods for Discovering Cognitive Biases in a Visual Analytics Environment. In: *Cognitive Biases in Visualizations*. Springer, pp. 61–73.

Benito-Santos, A., Sánchez, R.T., 2019. Cross-domain visual exploration of academic corpora via the latent meaning of user-authored keywords. IEEE Access 7, 98144–98160.

Bostock, M., Ogievetsky, V., Heer, J., 2011. D^3 data-driven documents. IEEE Trans. Visual Comput. Graphics 17 (12), 2301–2309.

Cairo, A., 2019. How charts lie: Getting smarter about visual information. WW Norton & Company, New York, NY, USA.

Clements, P., Northrop, L., 2002. Software product lines. Addison-Wesley, Boston, MA, USA.

M. Correll, E. Bertini, and S. Franconeri, "Truncating the y-axis: Threat or menace?," in *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, Honolulu, HI, USA, 2020: ACM, pp. 1-12, doi: 10.1145/3313831.3376222.

V. Dibia and Ç. Demiralp, "Data2Vis: Automatic generation of data visualizations using sequence to sequence recurrent neural networks," *IEEE computer graphics and applications,* 2019.

Dimara, E., Franconeri, S., Plaisant, C., Bezerianos, A., Dragicevic, P., 2020. A Task-Based Taxonomy of Cognitive Biases for Information Visualization. IEEE Trans. Visual Comput. Graphics 26 (2), 1413–1432. https://doi.org/10.1109/TVCG.2018.2872577.

Escobar-Pérez, J., Martínez, A., 2008. Validez de contenido y juicio de expertos: Una aproximación a su utilización. Avances en Medición 6.

S. Franconeri, L. Padilla, P. Shah, J. Zacks, and J. Hullman, "The science of visual data communication: What works," *Psychological Science in the Public Interest,* 2021.

Franconeri, S.L., Padilla, L.M., Shah, P., Zacks, J.M., Hullman, J., 2021. The Science of Visual Data Communication: What Works. Psychological Science in the Public Interest 22 (3), 110–161. https://doi.org/10.1177/15291006211051956.

García-Sánchez, F., Therón, R., Gómez-Isla, J., 2019. Visual literacy in New Media: Systematic review and mapping of the literature. Education in the Knowledge Society 20, 1–44.

D. Gotz and Z. Wen, "Behavior-driven visualization recommendation," in *Proceedings of the 14th international conference on Intelligent user interfaces*, 2009: ACM, pp. 315-324.

Gotz, D., Zhou, M.X., 2009. Characterizing users' visual analytic activity for insight provenance. Information Visualization 8 (1), 42–55.

Harsu, M., 2002. A survey on domain engineering. Citeseer.

Hu, K., Bakker, M.A., Li, S., Kraska, T., Hidalgo, C., 2019. In: ACM, p. 128.

I. Logre, S. Mosser, P. Collet, and M. Riveill, "Sensor data visualisation: a composition-based approach to support domain variability," in *European Conference on Modelling Foundations and Applications*, York, United Kingdom, R. J. Cabot J., Ed., 2014, vol. 8569: Springer, in Modelling Foundations and Applications. ECMFA 2014, pp. 101-116, doi: https://doi.org/10.1007/978-3-319-09195-2_7.

Ivanov, E., Khoroshavin, A., Karsakov, A., 2020. Visual programming environment based on data visualization grammar specification. Procedia Comput. Sci. 178, 434–439.

P. Kaur and M. Owonibi, "A Review on Visualization Recommendation Strategies," in *12th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP 2017)*, 2017, vol. 3: SCITEPRESS, pp. 266-273.

Kay, M., Kola, T., Hullman, J.R., Munson, S.A., 2016. When (ish) is my bus? user-centered visualizations of uncertainty in everyday, mobile predictive systems. In: In *ProceedIngs of the 2016 CHI Conference on Human Factors In ComputIng Systems*, pp. 5092–5103.

Key, A., Howe, B., Perry, D., Aragon, C., 2012. In: ACM, pp. 681–684.

A. G. Kleppe, J. Warmer, and W. Bast, "MDA Explained. The Model Driven Architecture: Practice and Promise," ed: Addison-Wesley Longman Publishing Co., Inc., Boston, MA, 2003.

J. Mackinlay, P. Hanrahan, and C. Stolte, "Show me: Automatic presentation for visual analysis," *IEEE transactions on visualization computer graphics,* vol. 13, no. 6, 2007.

Mei, H., Chen, W., Ma, Y., Guan, H., Hu, W., 2018. Viscomposer: A visual programmable composition environment for information visualization. Visual Informatics 2 (1), 71–81.

Millet, B., Carter, A.P., Broad, K., Cairo, A., Evans, S.D., Majumdar, S.J., 2020. Hurricane risk communication: visualization and behavioral science concepts. Weather Clim. Soc. 12 (2), 193–211.

R. Morgan G. Grossmann M. Schrefl M. Stumptner T. Payne VizDSL: A Visual DSL for Interactive Information Visualization Springer International Publishing, in Advanced Information Systems Engineering 2018 Cham 440 455.

S. O'Brien C. Lauer Testing the susceptibility of users to deceptive data visualizations when paired with explanatory text ACM 2018 USA 1 8 10.1145/3233756.3233961.

Palpanas, T., Chowdhary, P., Mihaila, G., Pinel, F., 2007. Integrated model-driven dashboard development. Information Systems Frontiers 9 (2–3), 195–208. https://doi.org/10.1007/s10796-007-9032-9.

A. V. Pandey, K. Rall, M. L. Satterthwaite, O. Nov, and E. Bertini, "How deceptive are deceptive visualizations? An empirical analysis of common distortion techniques," in *Proceedings of the 33rd annual acm conference on human factors in computing systems*, Seoul, Republic of Korea, 2015: ACM, pp. 1469-1478, doi: 10.1145/2702123.2702608.

Pandey, A.V., Manivannan, A., Nov, O., Satterthwaite, M., Bertini, E., 2014. The persuasive power of data visualization. IEEE Trans. Visual Comput. Graphics 20 (12), 2211–2220.

C. Perin, R. Vuillemot, C. D. Stolper, J. T. Stasko, J. Wood, and S. Carpendale, "State of the art of sports data visualization," in *Computer Graphics Forum*, 2018, vol. 37, no. 3: Wiley Online Library, pp. 663-686.

Pleuss, A., Wollny, S., Botterweck, G., 2013. In: ACM, pp. 13–22.

Pohl, K., Böckle, G., Van der Linden, F.J., 2005. Software Product Line Engineering: Foundations, Principles and Techniques. Springer-Verlag, New York Inc, New York, NY, USA.

Ren, D., Lee, B., Brehmer, M., 2018. Charticulator: Interactive construction of bespoke chart layouts. IEEE Trans. Visual Comput. Graphics 25 (1), 789–799.

Richters, M., Gogolla, M., 2002. "OCL: Syntax, semantics, and tools," in *Object Modeling with the OCL*. Springer, 42–68.

Ruginski, I.T. et al., 2016. Non-expert interpretations of hurricane forecast uncertainty visualizations. Spatial Cognition & Computation 16 (2), 154–172.

Sarikaya, A., Correll, M., Bartram, L., Tory, M., Fisher, D., 2018. What Do We Talk About When We Talk About Dashboards? IEEE Trans. Visualiz. Comput. Graphics 25 (1), 682–692. https://doi.org/10.1109/TVCG.2018.2864903.

Satyanarayan, A., Moritz, D., Wongsuphasawat, K., Heer, J., 2016. Vega-lite: A grammar of interactive graphics. IEEE Trans. Visual Comput. Graphics 23 (1), 341–350.

E. Tufte and P. Graves-Morris, "The visual display of quantitative information.; 1983," ed. Cheshire, CT, USA: Graphics Press, 2014.

J. Van Gurp, J. Bosch, and M. Svahnberg, "On the notion of variability in software product lines," in *Software Architecture, 2001. Proceedings. Working IEEE/IFIP Conference on*, 2001: IEEE, pp. 45-54.

A. Vázquez Ingelmo, F. J. García-Peñalvo, R. Therón Sánchez, and M. Á. Conde González, "Extending a dashboard meta-model to account for users' characteristics and goals for enhancing personalization," *Proceedings of LASI-SPAIN 2019. Learning Analytics Summer Institute Spain 2019: Learning Analytics in Higher Education (Vigo, Spain, June 27-28, 2019). CEUR Workshop Proceedings Series,* 2019. [Online]. Available: http://hdl.handle.net/10366/139803.

A. Vázquez Ingelmo, A. García-Holgado, F. J. García-Peñalvo, and R. Therón Sánchez, "A Meta-modeling Approach to Take into Account Data Domain Characteristics and Relationships in Information Visualizations," in *Trends and Innovations in Information Systems and Technologies, WorldCIST 2021*, Azores, Portugal, Á. Rocha, H. Adeli, G. Dzemyda, F. Moreira, and A. M. R. Correia, Eds., 2021, vol. 2, Cham, Switzerland: Springer Nature, in Advances in Intelligent Systems and Computing Series Series, no. 1366, 2021, pp. 570-580, doi: 10.1007/978-3-030-72651-5_54. [Online]. Available: http://hdl.handle.net/10366/145626.

A. Vázquez-Ingelmo and R. Therón, "Beneficios de la aplicación del paradigma de líneas de productos software para generar dashboards en contextos educativos," *RIED. Revista Iberoamericana de Educación a Distancia,* vol. 23, no. 2, pp. 169-185, 07/01 2020, doi: 10.5944/ried.23.2.26389.

Vázquez-Ingelmo, A., Cruz-Benito, J., García-Peñalvo, F.J., Martín-González, M., 2018. Scaffolding the OEEU's Data-Driven Ecosystem to Analyze the Employability of Spanish Graduates. In: García-Peñalvo, F.J. (Ed.), Global Implications of Emerging Technology Trends. IGI Global, Hershey, PA, USA, pp. 236–255.

A. Vázquez-Ingelmo F.J. García-Peñalvo R. Therón Domain engineering for generating dashboards to analyze employment and employability in the academic context 2018 Association for Computing Machinery New York, NY, USA 896 901 10.1145/3284179.3284329. [Online]. Available: doi: 10.1145/3284179.3284329.

A. Vázquez-Ingelmo, F. J. García-Peñalvo, and R. Therón, "Capturing high-level requirements of information dashboards' components through meta-modeling," in *TEEM'19 Proceedings of the Seventh International Conference on Technological Ecosystems for Enhancing Multiculturality*, León, Spain, M. Á. Conde-González, F. J. Rodríguez-Sedano, C. Fernández-Llamas, and F. J. García-Peñalvo, Eds., October 16th-18th 2019, New York, NY, USA: Association for Computing Machinery, 2019, pp. 815–821, doi: 10.1145/3362789.3362837. [Online]. Available: https://doi.org/10.1145/3362789.3362837.

A. Vázquez-Ingelmo, F. J. García-Peñalvo, R. Therón, and A. García-Holgado, "Specifying information dashboards' interactive features through meta-model instantiation," in *LASI-SPAIN 2020. Learning Analytics Summer Institute Spain 2020: Learning Analytics. Time for Adoption?*, Valladolid, Spain, A. Martínez-Monés, A. Álvarez, M. Caeiro-Rodríguez, and Y. Dimitriadis, Eds., June 15-16 2020, Aachen, Germany: CEUR-WS.org, in CEUR Workshop Proceedings Series, no. 2671, pp. 47-59.

A. Vázquez-Ingelmo, F. J. García-Peñalvo, and R. Therón, "Generating Dashboards Using Fine-Grained Components: A Case Study for a PhD Programme," in *Learning and Collaboration Technologies. Design, Experiences. 7th International Conference, LCT 2020, Held as Part of the 22nd HCI International Conference, HCII 2020*, Copenhagen, Denmark, P. Zaphiris and A. Ioannou, Eds., July 19–24 2020, no. 12205, Cham, Switzerland: Springer Nature, in Lecture Notes in Computer Science, 2020, pp. 303-314, doi: 10.1007/978-3-030-50513-4_23.

A. Vázquez-Ingelmo F. García-Peñalvo R. Therón A. García-Holgado Specifying information dashboards' interactive features through meta-model instantiation (Proceedings of LASI-SPAIN 2020. Learning Analytics Summer Institute Spain 2020: Learning Analytics. Time for Adoption? Valladolid, Spain, June 15–16, Aachen 2020 CEUR-WS.org Germany 2020.

A. Vázquez-Ingelmo, A. García-Holgado, F. J. García-Peñalvo, R. Therón, and R. Colomo-Palacios, "Content-validation questionnaire of a meta-model to ease the learning of data visualization concepts," presented at the Learning Analytics Summer Institute Spain 2022 (LASI Spain 22), Salamanca, Spain, 20-21 June, 2022.

A. Vázquez-Ingelmo, A. García-Holgado, F. J. García-Peñalvo, and R. Therón, "Proof-of-concept of an information visualization classification approach based on their fine-grained features," *Expert Systems, e12872,* 2022, doi: 10.1111/exsy.12872.

A. Vázquez-Ingelmo, F. J. García-Peñalvo, R. Therón, and M. Á. Conde, "Representing Data Visualization Goals and Tasks through Meta-Modeling to Tailor Information Dashboards," *Applied Sciences,* vol. 10, no. 7, p. 2306, 2020. [Online]. Available: https://www.mdpi.com/2076-3417/10/7/2306.

Vázquez-Ingelmo, A., García-Peñalvo, F.J., Therón, R., 2019. Information Dashboards and Tailoring Capabilities - A Systematic Literature Review. IEEE Access 7, 109673–109688. https://doi.org/10.1109/ACCESS.2019.2933472.

Vázquez-Ingelmo, A., García-Peñalvo, F.J., Therón, R., 2019. Taking advantage of the software product line paradigm to generate customized user interfaces for decision-making processes: a case study on university employability. PeerJ Computer Science 5, e203.

Vázquez-Ingelmo, A., García-Peñalvo, F.J., Therón, R., 2019. Addressing Fine-Grained Variability in User-Centered Software Product Lines: A Case Study on Dashboards. In: World Conference on Information Systems and Technologies. Springer, pp. 855–864.

Viegas, F.B., Wattenberg, M., Van Ham, F., Kriss, J., McKeon, M., 2007. Manyeyes: a site for visualization at internet scale. IEEE Trans. Visual Comput. Graphics 13 (6), 1121–1128.

H. Wickham and M. H. Wickham, "The ggplot package," ed: Google Scholar, 2007.

Wilke, C.O., 2019. Fundamentals of data visualization: a primer on making informative and compelling figures. O'Reilly Media, Sebastopol, CA, USA.

Wilkinson, L., 2012. The grammar of graphics. In: Handbook of computational statistics. Springer, pp. 375–414.

Wongsuphasawat, K., Moritz, D., Anand, A., Mackinlay, J., Howe, B., Heer, J., 2015. Voyager: Exploratory analysis via faceted browsing of visualization recommendations. IEEE Trans. Visual Comput. Graphics 22 (1), 649–658.

7.37 Appendix AK. Resumen extendido: Generación automática de interfaces software para el soporte a la toma de decisiones. Aplicación de ingeniería de dominio y machine learning

Generación automática de interfaces software para el soporte a la toma de decisiones. Aplicación de ingeniería de dominio y machine learning.

Tesis doctoral

Programa de doctorado en Ingeniería Informática
Departamento de Informática y Automática
Universidad de Salamanca (https://ror.org/02f40zc51)
Salamanca, España

Doctoranda

Andrea Vázquez Ingelmo

Directores

Dr. Francisco José García-Peñalvo
Dr. Roberto Therón Sánchez

Resumen extendido

Palabras clave: *Visualización de datos, Visualización de información, Dashboards, Desarrollo dirigido por modelos, Arquitectura dirigida por modelos, Líneas de productos de software, Metamodelado, Generación de conocimiento, Interfaces gráficas de usuario, Interacción persona-ordenador.*

El análisis de datos es un proceso clave para fomentar la generación de conocimiento en determinados dominios o campos de estudio. Con una sólida base informativa derivada del análisis de datos, los responsables de la toma de decisiones pueden diseñar estrategias con el objetivo de obtener beneficios en sus ámbitos de actuación específicos. Sin embargo, dado el constante crecimiento de los volúmenes de datos, el análisis de éstos debe contar con herramientas potentes que permitan la extracción de conocimientos.

Pese a que conceptos técnicos como "dirigido por datos" (en inglés, *data-driven*) y "toma de decisiones" pueden considerarse jerga empresarial, es necesario ser conscientes de que estos conceptos influyen en todos los aspectos de nuestro día a día. Llevamos a cabo procesos de toma de decisiones cuando organizamos nuestro tiempo para coger el autobús puntualmente o cuando miramos las previsiones meteorológicas para decidir si debemos llevar un paraguas en un día nublado. Estamos, conscientemente o no, tomando decisiones basadas en datos constantemente.

Pero, aunque nos enfrentemos a estas decisiones cada día, la comprensión de los datos no es una tarea simple ni un proceso sencillo que pueda darse por sentado. Hay varios mecanismos que se activan [1] para ayudarnos a generar nuevos conocimientos a partir de pequeñas porciones de datos [2-4]. Sin embargo, estos mecanismos no son infalibles. De hecho, vivir en una sociedad altamente conectada, donde se generan continuamente flujos de datos, puede dificultar estos procesos debido a la sobrecarga de información [5, 6].

En este contexto, el apoyo tecnológico es crucial para facilitar la generación de conocimiento en entornos complejos y con importantes cantidades de datos. Los *dashboards* o cuadros de mando y las visualizaciones de datos ofrecen una solución de software para analizar grandes volúmenes de datos de forma visual con el fin de identificar patrones y relaciones, así como tomar decisiones en función de la información presentada.

Las visualizaciones de datos y los *dashboards* están compuestos de un conjunto de elementos visuales colocados y configurados en función de los datos de entrada. Pero estos elementos se influyen mutuamente, y hay conceptos que ni siquiera aparecen en la pantalla pero que son cruciales, como las características del usuario. Debido a esta complejidad, es importante apoyarse en conocimiento experto a la hora de desarrollar visualizaciones de datos y *dashboards*.

Sin embargo, los usuarios quieran crear una visualización no siempre tendrán a su disposición a un experto en la materia para guiar el proceso y aplicar los principios de diseño adecuados.

Varias herramientas han intentado abordar esta cuestión asistiendo al usuario y automatizando la generación de visualizaciones de datos y *dashboards* mediante la implementación de procesos generativos que capturan y aplican el conocimiento de los expertos adaptando los elementos visuales en función de los datos y el contexto. Este es el caso de herramientas comerciales como Tableau, Microsoft Excel, Google Charts, etc.

Aunque estas plataformas son muy potentes, sigue existiendo un problema relacionado con la transferencia del conocimiento experto a los profesionales, y con la expresividad de las visualizaciones obtenidas.

Por otro lado, las librerías de programación declarativa e imperativa pueden mejorar la expresividad y funcionalidad de las visualizaciones de datos y *dashboards* desarrollados, pero, en este caso, suelen venir acompañadas de una pronunciada curva de aprendizaje que dificulta el proceso de implementación.

Teniendo en cuenta lo anterior, esta Tesis Doctoral aborda el reto de mejorar el proceso de desarrollo de *dashboards* y visualizaciones de datos, así como su calidad y prestaciones en términos de personalización, usabilidad y flexibilidad, entre otros.

La hipótesis principal de este trabajo se plantea como sigue: "*La automatización del desarrollo de interfaces de usuario a medida para apoyar los procesos de toma de decisiones incrementa sus beneficios en términos de características funcionales y no funcionales*."

En otras palabras, el objetivo principal de la investigación es diseñar e implementar un marco generativo para el desarrollo automático, sistemático y personalizado de *dashboards*, además de discutir los conocimientos alcanzados al automatizar la generación de estas herramientas. Mediante este enfoque, se trata de fomentar la individualización, la usabilidad y la flexibilidad para maximizar los beneficios derivados de las herramientas generadas.

Debido a la naturaleza mixta de los artefactos y escenarios propuestos, esta tesis se ha llevado a cabo siguiendo un proceso iterativo en el que el conocimiento adquirido a través de las experiencias pasadas y los resultados de los diferentes ciclos es crucial para las siguientes etapas. Para llevar a cabo este proceso se ha seguido el marco metodológico de la Investigación-Acción [7].

Kemmis planteó la Investigación-Acción [8] como un método de indagación llevado a cabo por los participantes en situaciones sociales con el objetivo de mejorar y comprender sus propias prácticas sociales y sus contextos.

Posteriormente, McTaggart y Kemmis describieron las características de esta metodología. La metodología de Investigación-Acción se basa en una espiral cíclica de investigación y acciones compuesta por una serie de fases y secuencias [9]. Por lo tanto, la Investigación-Acción es un proceso iterativo en el que cada ciclo proporciona un resultado que será la entrada para el siguiente ciclo.

El marco elegido para el desarrollo de *software* es un enfoque ágil basado en SCRUM [10]. Este marco proporciona los procesos, reglas, prácticas, roles y artefactos necesarios para aumentar la productividad de los equipos de desarrollo mediante un ciclo de desarrollo de software iterativo e incremental.

Para evaluar los artefactos derivados se ha utilizado un enfoque de investigación. La investigación se ha llevado a cabo utilizando métodos cuantitativos y cualitativos [11], aprovechando las dos perspectivas para obtener una visión más amplia de los resultados para afrontar los siguientes ciclos de investigación-acción.

Sin embargo, es necesario formalizar el problema a tratar antes de poder iniciar dichos ciclos. Para ello, se ha realizado una revisión sistemática de la literatura (SLR, por sus siglas en inglés) bajo las directrices propuestas por Kitchenham [12, 13] y García-Peñalvo [14], con el fin de estudiar problemas similares y soluciones desarrolladas previamente para comprender el contexto y el estado actual del campo.

Además de obtener un panorama de soluciones, el principal resultado de la revisión bibliográfica es un análisis crítico de las metodologías y arquitecturas encontradas en los trabajos seleccionados. Este tipo de análisis ofrece un buen punto de partida para diseñar e implementar la primera propuesta de un sistema para la generación automática de *dashboards*.

Las preguntas de investigación abarcaron aspectos relevantes para tener en cuenta a la hora de abordar los flujos generativos de *dashboards*. Con la información recopilada, fue posible seleccionar la mejor estrategia para implementar enfoques que abordan la generación automática de estas herramientas.

En virtud de los resultados del SLR, la decisión fue seguir un enfoque de metamodelado para conceptualizar el marco generativo y el paradigma de la línea de productos de software (SPL, por sus siglas en inglés) para materializar y transformar las características abstractas en código fuente. El análisis de los artículos recuperados ha demostrado que estos dos enfoques son viables en este ámbito; casi un tercio de los trabajos seleccionados (8 de 30) emplean uno de estos dos paradigmas.

Sin embargo, la viabilidad de las soluciones no fue el único objeto de estudio. Otros atributos, como la flexibilidad y las capacidades evolutivas, la posibilidad de transferir las soluciones a cualquier dominio de datos, la trazabilidad de los requisitos y el potencial de integrar algoritmos de inteligencia artificial (IA) para adaptar las características del *dashboard* a los cambios del entorno también fueron objeto de esta revisión.

Por ejemplo, algunos trabajos mencionan el uso de modelos de aprendizaje automático (ML, por sus siglas en inglés) para liberar a los usuarios de tareas complejas como la configuración del diseño del *dashboard*. Sin embargo, estas aplicaciones no están detalladas o se encuentran en sus primeras fases de desarrollo.

Esta línea de investigación es muy prometedora, ya que los enfoques de IA podrían aportar varios beneficios para ayudar a los usuarios y proporcionarles pautas útiles para aprender y comprender cómo diseñar *dashboards* y visualizaciones eficaces. En este sentido, la elección de un enfoque de metamodelado también es adecuada para aplicar métodos de ML, ya que estos modelos requieren datos estructurados de los que aprender. Las instancias del metamodelo pueden proporcionarse como entradas para identificar los patrones que hacen que determinadas configuraciones sean útiles, eficientes, eficaces, utilizables, etc.

En definitiva, la realización de este análisis ha proporcionado evidencias claras de que las soluciones que siguen los paradigmas de meta-modelado y/o SPL cumplen con estas propiedades, concluyendo que la versatilidad de estas metodologías proporciona buenos cimentos para implementar un sistema generativo basado en un metamodelo.

Concretamente, los metamodelos son artefactos cruciales en los paradigmas de desarrollo dirigido por modelos (MDD, por sus siglas en inglés) [15-17], ya que permiten mapear entidades de niveles muy abstractos a entidades más detalladas e incluso al código fuente mediante transformaciones.

El *Object Management Group* (OMG) propone la arquitectura dirigida por modelos (MDA, por sus siglas en inglés) como guía para implementar el desarrollo dirigido por modelos. Esta arquitectura proporciona un marco para el desarrollo de software que emplea modelos para describir y definir el sistema objetivo [18]. La principal diferencia entre MDD y MDA es que la propuesta del OMG utiliza un conjunto de estándares: meta-objeto (MOF), lenguaje de modelado unificado (UML), intercambio de metadatos XML (*Extensible Markup Language*), y consulta/vista/transformación (QVT).

El marco MDA está compuesto por cuatro capas arquitectónicas. Cada capa representa un nivel de abstracción de las entidades representadas. La capa más abstracta (nivel M3) se conoce como el nivel meta-metamodelo. Esta capa define estructuras y conceptos básicos para representar capas menos abstractas, así como a sí misma, y puede implementarse con el mencionado estándar MOF.

El nivel M2, es decir, el nivel de metamodelo, cumple con el metamodelo y representa entidades y relaciones abstractas. Los metamodelos pueden verse como lenguajes específicos de dominio (DSL, por sus siglas en inglés) que expresan características comunes y genéricas del dominio objetivo.

El nivel M1, define modelos que instancian y especifican las características abstractas contenidas en el metamodelo, y su sintaxis debe cumplir con el nivel M2.

Por último, el nivel M0 representa aplicaciones del mundo real basadas en un modelo M1 previamente definido.

Siguiendo este paradigma y la metodología de investigación-acción, el metamodelo desarrollado ha sido sometido a diferentes iteraciones antes de obtener la versión final. Estos incrementos no sólo han permitido centrarse en entidades de dominio específicas en cada iteración, sino también revisar y resolver posibles problemas que pudieran surgir en iteraciones posteriores.

Las primeras iteraciones se centraron en los elementos tangibles de los *dashboards* (diseño, recursos, componentes visuales, etc.), seguidas de un par de iteraciones para abordar la caracterización de la audiencia (el usuario).

Los dos últimos incrementos abordaron conceptos de más alto nivel, como los patrones de interacción (comportamientos dinámicos que no son explícitamente tangibles) y las características del dominio de los datos (el conocimiento implícito sobre el contexto en el que se enmarcan los conjuntos de datos del mundo real).

El desarrollo del metamodelo no solo proporcionó un artefacto para construir un marco generativo para los *dashboards*, sino que también incrementó el conocimiento del autor respecto a este dominio, que es otro beneficio bien conocido del uso de la ingeniería de dominio: capturar, generar y reutilizar el conocimiento.

Este enfoque también impulsó la decisión de utilizar plantillas de código como método para materializar los puntos de variabilidad de la SPL. Las plantillas de código se seleccionaron dada su idoneidad en este contexto y su parecido con el metamodelado en la filosofía de encapsular y descomponer entidades complejas en elementos primitivos.

El emparejamiento de las entidades del metamodelo con sus respectivos fragmentos de código permitió el desarrollo de un generador de *dashboards*. Este hito marcó el inicio de las fases de validación y aplicación de los artefactos desarrollados.

En este sentido, se realizó un estudio de validación del contenido del metamodelo para comprobar la coherencia, la relevancia y la claridad de las diferentes secciones de este artefacto. Al validar el metamodelo, es posible identificar las posibles limitaciones e inconvenientes de la representación del dominio de los *dashboards* y abordarlos antes de utilizar este artefacto para instanciar estas herramientas en el mundo real.

En primer lugar, se aplicó el marco de calidad del metamodelo propuesto en [19] para comprobar la calidad de la versión Ecore del metamodelo de *dashboards* antes y después de introducir las modificaciones. El metamodelo cumplía con las treinta características del marco relacionadas con el diseño, las mejores prácticas, las convenciones de nomenclatura y las métricas, lo que demuestra su calidad.

En segundo lugar, se llevó a cabo una validación por expertos de la versión final del metamodelo de *dashboards*. El objetivo de esta validación es comprobar si las secciones del metamodelo son claras, coherentes y pertinentes. En particular, se aplicó el juicio de expertos [20].

Para esta tarea, se creó un cuestionario en línea con seis secciones diferentes del metamodelo (diseño, características del usuario, objetivos y tareas, relaciones entre el usuario y el *dashboard*, componentes primitivos de las visualizaciones de datos, y dominio y operaciones de datos), además del metamodelo completo. Cada sección se valoró en función de las dimensiones mencionadas utilizando puntuaciones de 1 a 4, donde 1 implica que la sección no cumple el criterio, y 4 que lo cumple en gran medida.

Los resultados arrojaron buenas puntuaciones (puntuaciones altas -4- y medias -3-) y sólo cinco puntuaciones bajas -2- relacionadas con la relevancia y coherencia de la sección de diseño del *dashboard* (debido a su simplicidad) y la relevancia de las primitivas de visualización y el dominio de los datos (debido al alto nivel de detalle que podría ser "a demasiado bajo nivel para los usuarios finales").

En lo que se refiere a aplicaciones, se han llevado a cabo varias investigaciones para probar el funcionamiento del metamodelo y del generador de *dashboards* en diferentes escenarios, tanto teóricos como prácticos.

En cuanto a la dimensión teórica de las investigaciones, el metamodelo se ha integrado con éxito con otros metamodelos para apoyar la generación de conocimiento en ecosistemas de aprendizaje. Aunque la existencia de un metamodelo de ecosistemas de aprendizaje resuelve la mayor parte de los problemas asociados a la definición y desarrollo de estas soluciones tecnológicas, existen algunas cuestiones relacionadas con el análisis de los flujos de información y el apoyo a los procesos de toma de decisiones que deberían mejorarse.

La conexión de ambos metamodelos dio lugar a un metamodelo holístico que se puso a prueba mediante la instanciación de un modelo centrado en la provisión de un ecosistema de salud para apoyar a cuidadores [21]. Se diseñó un *dashboard* para lograr diferentes objetivos de información relacionados con la gestión de los cuidadores.

El metamodelo también se aplicó como marco para conceptualizar e instanciar *dashboards* en diferentes dominios. Los escenarios de esta categoría incluyen la instanciación manual del metamodelo en modelos que representan visualizaciones de datos del mundo real, la identificación de sesgos, la automatización del proceso de instanciación a través de herramientas externas y el concepto de proporcionar *dashboards* como servicio a través de interfaces de programación de aplicaciones (API, por sus siglas en inglés).

En cuanto a las aplicaciones prácticas, se ha hecho hincapié en cómo transformar el metamodelo en una instancia adaptada a un contexto específico, y cómo transformar finalmente este modelo posterior en código, es decir, en el producto final y funcional. En este contexto, se probó la generación automática de *dashboards* en un programa de doctorado.

En primer lugar, se llevó a cabo un proceso de elicitación de requisitos para modelar los *dashboards* que se generarían. El proceso de obtención de requisitos incluyó una entrevista con un miembro del comité de calidad del Programa de Doctorado para entender qué datos podrían mostrarse en un posible *dashboard*.

Estas propuestas fueron finalmente implementadas en el Programa de Doctorado de Educación en la Sociedad del Conocimiento de la Universidad de Salamanca (España) [22]. Se realizó un estudio de usabilidad para puntuar la integración de la visualización de datos en el portal de doctorado utilizando el cuestionario *System Usability Scale* (SUS) [23]. 35 participantes (entre los que se encontraban estudiantes de doctorado, asesores y gestores) respondieron al cuestionario SUS. La media de usabilidad percibida de las visualizaciones del portal de doctorado fue de 75,36, lo que puede considerarse una buena puntuación en el contexto del cuestionario SUS.

Esta aplicación sirvió para comprobar la viabilidad de la generación automática de código en entornos reales. Sin embargo, la transformación del metamodelo en modelos efectivos de *dashboards* y visualizaciones fue llevada a cabo manualmente. En este sentido, se exploró una combinación del pipeline generativo con la IA. La IA se encargaría de aprender las buenas prácticas y directrices en el diseño de la visualización de datos y de seleccionar la mejor configuración dadas las necesidades específicas del contexto.

En este sentido, se realizó un estudio para probar la idea de utilizar el generador de *dashboards* para entrenar modelos de ML e identificar configuraciones potencialmente perjudiciales en las visualizaciones de datos [24]. Para ello, se generó un conjunto de datos de entrenamiento de diferentes visualizaciones de información mediante el generador desarrollado. Además, al utilizar este enfoque, las configuraciones de las visualizaciones generadas ya estaban estructuradas y preparadas para su procesamiento y uso como entrada para algoritmos de ML, lo que también ahorraba tiempo y permitía centrarse más a fondo en el proceso de etiquetado. Este proceso de etiquetado consistió en evaluar las visualizaciones generadas como "útiles" o "no útiles", teniendo en cuenta si los datos mostrados con distintas configuraciones podrían llevar a engaño o a conclusiones erróneas.

Una vez generado el conjunto de datos, se entrenaron diferentes modelos de ML: Naïve Bayes, Support Vector Machine (SVM), Ada Boost y Random Forest (RF). El clasificador RF obtuvo los mejores resultados en términos de exactitud, precisión, recuperación y puntuación F1. Dados estos resultados, se eligió el clasificador RF para evaluar los resultados de las predicciones individuales introduciendo manualmente los valores de otras visualizaciones.

Una de las razones de la alta precisión es que el modelo imita los criterios definidos previamente para clasificar cada visualización. Sin embargo, esto también indica que el modelo aprendió satisfactoriamente las características más importantes del proceso de clasificación. Las características más importantes fueron los valores máximos y mínimos del dominio de las escalas del eje X, del eje Y, y del tamaño de los elementos visuales. Este resultado se alinea con investigaciones anteriores encontradas en la literatura, en las que la definición de los rangos de las escalas de una visualización es determinante para su correcta comprensión [25].

Por último, se desarrolló una plataforma que permite a los usuarios instanciar modelos a partir del metamodelo de *dashboards*. Esta plataforma, denominada MetaViz, permitió convertir el metamodelo en un recurso más accesible para las aplicaciones prácticas, así como implementar los mecanismos para abordar transformaciones complejas del modelo de forma visual.

La arquitectura de MetaViz se basa en los conceptos del paradigma MDA descritos anteriormente, y las capas MDA se materializan en el sistema a través de diferentes módulos de software.

No sólo la arquitectura del software se basa en las capas MDA, sino también la interfaz de usuario. Una barra de herramientas permite a los usuarios añadir nuevos elementos básicos a su espacio de trabajo. Los parámetros de configuración de cada elemento también se basan en los atributos y relaciones del metamodelo. El espacio de trabajo permite la instanciación de nuevos *dashboards* conectando y configurando los diferentes "meta-elementos". Por último, un lienzo muestra el *dashboard* o las visualizaciones generadas en tiempo real.

Finalmente, se ha llevado a cabo un conjunto de casos de estudios en diferentes dominios. El metamodelo y el sistema generativo se probaron en el ámbito del empleo y la empleabilidad. Después de esta integración, el metamodelo se mejoró y la nueva versión, junto con el enfoque de ecosistema conceptualizado en [26], se integraron en diferentes plataformas de salud basadas en datos. La dimensión de aprendizaje del metamodelo y el instanciador gráfico -MetaViz- también se ha evaluado en el contexto educativo.

La versión preliminar del metamodelo y la línea de productos de software se utilizaron para abordar la personalización del análisis visual [27, 28] en el dominio del empleo y la empleabilidad. En concreto, este enfoque se aplicó en el Observatorio Español de Empleo y Empleabilidad Universitaria (OEEU) [29, 30]. Esta red de investigadores y técnicos realiza estudios sobre estos campos en el contexto académico [30-32], a través de un enfoque basado en datos para recoger, analizar, visualizar y difundir datos de empleo y empleabilidad de los egresados de las universidades españolas.

Para los *dashboards* del Observatorio, se definieron tres componentes visuales configurables: un diagrama de dispersión, un diagrama de cuerdas y un mapa de calor. Estas visualizaciones responden a los requisitos de los datos del Observatorio, pero pueden reutilizarse para otros dominios de datos. Se diseñó un DSL para lograr la conexión entre el metamodelo y el código fuente, y se basó en las características del dominio identificado del SPL.

Los resultados muestran el éxito de la implementación de la SPL en este contexto. La versión anterior del sistema del OEEU proporcionaba *dashboards* estáticos con los mismos indicadores para cada usuario. Con este enfoque, los *dashboards* pueden personalizarse mediante la instanciación de los componentes y funcionalidades deseados.

En lo que respecta al ámbito de la salud, se han llevado a cabo diferentes trabajos de investigación relacionados con plataformas dirigidas por datos y centrados en proporcionar interfaces que ayuden tanto a los procesos de toma de decisiones como a las aplicaciones de IA. En primer lugar, el desarrollo de una plataforma (CARTIER-IA) que da soporte a todas las tareas relacionadas con la gestión de datos (incluida la recopilación de datos estructurados e imágenes médicas) y que también permite tanto a los profesionales sanitarios como a los científicos de datos aplicar modelos de IA a las imágenes almacenadas.

El diseño de la plataforma ha hecho posible la aplicación de algoritmos de IA en los datos de imagen sin requerir conocimientos de programación. Los algoritmos de IA están disponibles para que cualquier usuario los ejecute y recupere sus resultados. Esta arquitectura también permite la integración de fuentes de datos externas (como REDCap [33]) y otros módulos, como el generador de *dashboards* desarrollado en esta tesis.

En segundo lugar, se ha implementado otra plataforma gráfica (KoopaML) para ofrecer interfaces intuitivas y educativas con el fin de construir y ejecutar enfoques de ML en el contexto médico. La arquitectura de KoopaML se basa en diferentes módulos conectados por flujos de información. Uno de los principales propósitos de este diseño es proporcionar flujos flexibles con componentes y algoritmos reutilizables.

Para dar soporte al análisis exploratorio de datos [34], que es una parte crucial de las aplicaciones de ML, se integró en KoopaML el generador de *dashboards*. Esta integración permitió la generación dinámica de resúmenes de los datos de entrada, así como informes de los resultados derivados del proceso de entrenamiento adaptados a las métricas requeridas por el usuario.

En tercer lugar, el metamodelo y el generador también se integraron en un servicio web para visualizar los resultados del estudio SALMANTICOR [35], un estudio descriptivo transversal de base poblacional sobre la prevalencia de la cardiopatía estructural y sus factores de riesgo con un total de 2.400 individuos. La plataforma no sólo se centró en proporcionar visualizaciones básicas para resumir los resultados, sino también en ofrecer una buena experiencia de usuario para alcanzar conocimientos sobre el estudio [36]. Por estas razones, la arquitectura se compone de módulos que supervisan la recuperación, la visualización y el análisis de cada sección del estudio. La división de la arquitectura en módulos individuales pero relacionados entre sí permite la flexibilidad para ampliar la plataforma con más funcionalidades, análisis y visualizaciones.

El *front-end* proporciona una interfaz utilizable y visualizaciones de datos para navegar por los resultados del estudio, y el *back-end* ofrece funcionalidades de cálculo, almacenamiento y recuperación de datos a través de llamadas a la API. Además, las visualizaciones de datos también pueden generarse a demanda mediante interacciones simples utilizando el generador de *dashboards*.

Por último, en lo que respecta al contexto educativo, se diseñó un estudio para probar el rol del metamodelo y el instanciador gráfico (MetaViz) como recursos de aprendizaje.

Existen varias plataformas para facilitar el proceso de diseño e implementación de visualizaciones de datos. Sistemas como Tableau (https://www.tableau.com/), Microsoft Excel (https://www.microsoft.com/microsoft-365/excel), Power BI (https://powerbi.microsoft.com/), etc., proporcionan interfaces gráficas que permiten a los usuarios sin experiencia en programación crear visualizaciones de datos e incluso les ayudan en el proceso de diseño para elegir las mejores configuraciones. Sin embargo, es fundamental comprender y tener en cuenta todos los elementos que intervienen en las visualizaciones de datos para ofrecer una visualización de la información eficaz y que no sea confusa o engañosa [37-40].

Para explorar el potencial papel educativo del metamodelo, se desarrolló un estudio piloto para medir la comprensibilidad de los elementos que intervienen en el diseño de visualizaciones de datos y *dashboards*. El objetivo de este estudio es comprobar si MetaViz ofrece una experiencia más educativa y proporciona a los usuarios una mayor comprensión de los conceptos del campo de la visualización de datos que otras herramientas comerciales.

Siguiendo los resultados y aplicaciones descritas, se ha podido observar que el desarrollo de un metamodelo de *dashboards* ha abierto varias vías de investigación, tanto a nivel teórico como práctico.

La versatilidad de este artefacto y del sistema generativo ha sido validada a través de los casos de estudio descritos previamente. En cuanto a la validación de expertos, los resultados demostraron que las entidades y relaciones identificadas eran pertinentes, coherentes y comprensibles. Estas características son cruciales, especialmente en este contexto. Debido a la complejidad inherente al dominio de la visualización de datos y los cuadros de mando, es necesario transmitir el conocimiento y las entidades relacionadas con este contexto de forma comprensible.

La realización de estudios de casos en diferentes dominios también ha mejorado el metamodelo y el sistema generativo, ya que de las integraciones surgieron nuevas entidades y relaciones. La evolución de estos recursos puede observarse a través de los estudios presentados, partiendo de una versión muy básica y de grano grueso [27, 28] y terminando en un marco completo que ha servido para entrenar modelos de ML [24], conceptualizar *dashboards* a alto nivel [41-45], apoyar los procesos de toma de decisiones en plataformas de salud [36, 46-49], e incluso conducir la arquitectura e interfaz de una plataforma completa (MetaViz).

Todos estos estudios han formado parte de los ciclos de investigación-acción definidos, los cuales han permitido identificar nuevas relaciones, conceptos y matices en el ámbito de la visualización de datos, dando lugar a un metamodelo potente y versátil para definir *dashboards* y visualizaciones de datos.

Aunque existen otros metamodelos de *dashboards* en la literatura [50-52], no tienen en cuenta las características de grano fino, que son cruciales en este dominio, ya que una ligera modificación en el diseño de un *dashboard* o de una visualización de datos podría llevar a conclusiones significativamente diferentes [25, 53, 54].

El desarrollo del metamodelo no ha sido un proceso lineal. Se han realizado varias mejoras y modificaciones antes de obtener la versión actual. Estas modificaciones son el resultado de los estudios que se han derivado de la aplicación de la ingeniería de dominio en este contexto.

La primera versión del metamodelo carecía de detalles tanto en lo que respecta a los componentes visuales como al usuario final. La inclusión del usuario como elemento extremadamente significativo dentro del ámbito de los cuadros de mando y la visualización de datos es crucial. Los procesos de desarrollo de visualizaciones y *dashboards* comienzan con el usuario (obtención de requisitos) y terminan con el usuario (refinamiento del producto) [37, 55, 56], por lo que no sólo se deben tener en cuenta las características técnicas de un cuadro de mando a la hora de modelar estas herramientas, ya que estas características surgen de los requisitos de los usuarios y están influenciadas por ellos [57].

De hecho, los resultados de la instanciación en [41] y [43] han demostrado que los objetivos de información de los usuarios son necesarios para diseñar *dashboards* que soporten diferentes roles y necesidades.

Lo mismo ocurre con los trabajos derivados de la integración del sistema generativo en las plataformas sanitarias descritas en [36, 46-49]. La heterogeneidad de las fuentes de datos y la variedad de roles involucrados en el ámbito de la salud exigen interfaces altamente personalizables. En este sentido, la provisión de *dashboards* como servicio [26] permitió la posibilidad de generar estas herramientas de forma transparente y construir bajo demanda en diferentes etapas del análisis de datos estructurados, imágenes DICOM y resultados de ML.

Además, ha sido necesario incluir en el metamodelo entidades y relaciones relacionadas con el dominio datos. El experimento realizado en [24] permitió etiquetar las visualizaciones de datos utilizando el conocimiento de los expertos sobre el contexto de los datos. Sin embargo, en el segundo experimento [58], en el que los datos se generaron de forma aleatoria (y, por tanto, no pertenecían a ningún contexto específico), fue casi imposible decidir si una visualización era potencialmente engañosa, porque se necesitaba más información para realizar el etiquetado. Gracias a esta experiencia, se identificó la falta de la noción de contexto de los datos y de las variables de dominio en el metamodelo, lo cual se corrigió posteriormente [58].

En cuanto a los detalles técnicos, el generador de *dashboards* se ha probado con diferentes lenguajes: VegaLite [59] en [60], React (https://es.reactjs.org/) en [61] y, finalmente, D3.js [62] en [24, 26, 36, 43, 44, 48, 49] y en el desarrollo de MetaViz.

La decisión de utilizar D3.js fue impulsada por la expresividad proporcionada por este lenguaje, que se alinea con la estructura de grano fino del metamodelo. Sin embargo, el uso de plantillas de código para materializar las características de la SPL en código [63] proporciona la posibilidad de desarrollar los activos *software* a través de diferentes tecnologías.

El metamodelo también fue un recurso esencial para explorar la aplicación de la IA al ámbito de la visualización de datos [24]. En primer lugar, siguiendo el desarrollo dirigido modelos y la ingeniería de la línea de productos de software, fue posible generar automáticamente las visualizaciones que posteriormente se etiquetaron para construir un conjunto de entrenamiento. Además, en segundo lugar, el metamodelo proporcionó las características y relaciones para estructurar el esquema del conjunto de datos de entrenamiento, lo que supuso un paso crucial antes de aplicar cualquier algoritmo de ML.

Las métricas de precisión y exactitud muestran que el modelo de ML resultante aprendió de los conocimientos implícitos y las heurísticas que se utilizaron para etiquetar manualmente el conjunto de entrenamiento. Los criterios seguidos para entrenar los modelos podrían considerarse obvios o muy básicos (como no exagerar los valores de la escala o utilizar determinadas codificaciones) [24], pero este estudio estaba motivado en ofrecer un método para que los novatos o los usuarios no expertos sean conscientes de las configuraciones engañosas que pueden estar introduciendo inconscientemente en sus diseños.

Además, el metamodelo ha proporcionado la columna vertebral para el desarrollo de una plataforma de generación de *dashboards* a través de una interfaz gráfica. La plataforma MetaViz (https://metaviz.grial.eu/) representa la unificación de todo el conocimiento derivado de la investigación realizada durante esta tesis. La implementación de MetaViz ha servido para materializar el metamodelo en un recurso funcional que puede ser explorado interactivamente, instanciado y transformado de forma transparente en código real a través del sistema generativo.

El número de aplicaciones del metamodelo en diferentes dimensiones (teóricas y prácticas) y dominios (empleo y empleabilidad, salud y educación) es también un resultado en sí mismo. Todos los resultados asociados a esta tesis están impulsados por el metamodelo de *dashboards*, lo que demuestra también su versatilidad y flexibilidad a la hora de conceptualizar, generar y capturar conocimiento relacionado con los *dashboards* y las visualizaciones de datos.

En este sentido, es evidente concluir que los enfoques MDD y SPL son beneficiosos para mejorar los procesos de desarrollo de cuadros de mando de información y visualizaciones de datos a medida.

Esta tesis también ha servido no sólo para analizar la automatización del diseño e implementación de estas herramientas, sino para explorar cómo concienciar sobre buenas prácticas mientras se desarrollan.

La combinación de los resultados y las conclusiones anteriores han permitido afirmar que el objetivo principal y los subobjetivos derivados de esta tesis se han alcanzado, y tras sus resultados, que la hipótesis planteada al inicio de este proyecto de investigación es válida.

En lo que respecta al trabajo futuro, el desarrollo del metamodelo y el sistema generativo ha desbloqueado varias oportunidades de investigación relacionadas con diferentes áreas.

En primer lugar, el metamodelo ha demostrado ser un artefacto útil para impulsar la investigación relacionada con la visualización de datos. Varios conceptos capturados en el metamodelo pueden ser explorados más a fondo, específicamente los relacionados con las características y objetivos del usuario. Aunque modelar y detectar los sesgos y creencias de los usuarios es un reto, los beneficios derivados de la recolección de esta información serían muy valiosos para luchar contra las noticias falsas, la polarización y las visualizaciones de datos engañosas.

En el ámbito de la ingeniería de software, el metamodelo puede mejorarse con reglas más específicas [64] que recojan buenas prácticas y directrices relacionadas con el ámbito de la visualización de datos. Estas mejoras pueden sentar las bases de un sistema de recomendación basado en reglas o una base de conocimiento para el diseño de visualizaciones de datos.

Por otro lado, en el ámbito de la Interacción Persona-Ordenador, una línea de investigación futura es continuar con la validación del usuario en términos de rendimiento, usabilidad o satisfacción, entre otras métricas, de los *dashboards* generados, así como de la plataforma instanciadora gráfica.

Las ideas presentadas en esta tesis también pueden aplicarse en contextos educativos. El metamodelo de *dashboards* captura el conocimiento del dominio, proporcionando un recurso de aprendizaje para enseñar los elementos básicos de las visualizaciones de datos. Otros trabajos podrían consistir en medir el rendimiento del instanciador gráfico en cuanto a su capacidad para proporcionar una experiencia de aprendizaje a sus usuarios.

Finalmente, también hay oportunidades de investigación en el campo de la IA. La noción de identificar visualizaciones engañosas mediante el entrenamiento de algoritmos de IA a través de las entidades capturadas en el metamodelo puede seguirse explorando para mejorar el modelo, e incluso desarrollar un detector automatizado de visualizaciones de datos engañosas. Además, los modelos de IA también podrían aplicarse para ayudar al proceso de instanciación a través de configuraciones recomendadas dado el contexto del *dashboard* o visualización y las características del usuario.

Para concluir, a lo largo del desarrollo de esta tesis doctoral se han realizado una serie de publicaciones científicas para validar la propuesta. El proceso de publicación en diferentes medios ha permitido obtener la retroalimentación de expertos en la materia. En concreto, se han publicado 11 artículos en revistas indexadas y 23 ponencias en congresos internacionales, así como 1 capítulo de libro.

Cabe destacar la realización de dos estancias de investigación. La primera, una estancia de investigación virtual desde el 1 de julio de 2021 hasta el 10 de octubre de

2021, en Østfold University College, Departamento de Informática (Halden, Noruega). Esta estancia de investigación se centró en la validación del metamodelo.

La segunda estancia de investigación se realizó del 10 de enero de 2022 al 14 de abril de 2022 en el Departamento de Tecnología de Gráficos por Ordenador de Purdue University (West Lafayette, Indiana, Estados Unidos de América). La investigación estuvo relacionada con aplicaciones de visualización de datos y su dimensión educativa.

Por último, esta tesis doctoral ha recibido financiación del Ministerio de Educación y Formación Profesional de España con una beca FPU (FPU17/03276).

References

[1] A. Abran, J. W. Moore, P. Bourque, and R. Dupuis, Eds. *Guide to the Software Engineering Body of Knowledge. 2004 Version*. Los Alamitos, CA, USA: IEEE Computer Society, 2004.

[2] P. Cooper, "Data, information, knowledge and wisdom," *Anaesthesia & Intensive Care Medicine*, vol. 18, no. 1, pp. 55-56, 2017/01/01/ 2017, doi: https://doi.org/10.1016/j.mpaic.2016.10.006.

[3] J. Rowley, "The wisdom hierarchy: representations of the DIKW hierarchy," *Journal of Information Science*, vol. 33, no. 2, pp. 163-180, 2007/04/01 2007, doi: 10.1177/0165551506070706.

[4] C. Zins, "Conceptual approaches for defining data, information, and knowledge," *Journal of the American Society for Information Science and Technology*, vol. 58, no. 4, pp. 479-493, 2007, doi: https://doi.org/10.1002/asi.20508.

[5] D. Bawden and L. Robinson, "Information overload: An overview," *Oxford encyclopedia of political decision making*, 2020, doi: https://doi.org/10.17613/n4b2-1w48.

[6] D. Bawden, C. Holtham, and N. Courtney, "Perspectives on information overload," in *Aslib proceedings*, 1999: MCB UP Ltd.

[7] K. Lewin, "Action research and minority problems," *Journal of social issues*, vol. 2, no. 4, pp. 34-46, 1946.

[8] S. Kemmis, "Point-by-point guide to action research," *Victoria: Deakin University*, 1984.

[9] R. McTaggart and S. Kemmis, *The action research planner*. Deakin university, 1988.

[10] K. Schwaber, "Scrum development process," in *Business object design and implementation*: Springer, 1997, pp. 117-134.

[11] J. W. Creswell and J. D. Creswell, *Research design: Qualitative, quantitative, and mixed methods approaches*. Sage publications, 2017.

[12] B. Kitchenham, "Procedures for performing systematic reviews," Keele University Technical Report TR/SE-0401 and NICTA Technical Report 0400011T.1, 2004.

[13] B. Kitchenham and S. Charters, "Guidelines for performing Systematic Literature Reviews in Software Engineering. Version 2.3," School of Computer Science and Mathematics, Keele University and Department of Computer Science, University of Durham, Technical Report EBSE-2007-01, 2007. [Online]. Available: https://goo.gl/L1VHcw

[14] F. J. García-Peñalvo, "Developing robust state-of-the-art reports: Systematic Literature Reviews," *Education in the Knowledge Society*, vol. 23, 2022, doi: 10.14201/eks.28600.

[15] N. Anquetil *et al.*, "Traceability for model driven, software product line engineering," in *ECMDA Traceability Workshop Proceedings*, 2008, vol. 12: SINTEF, pp. 77-86.

[16] A. G. Kleppe, J. Warmer, and W. Bast, "MDA Explained. The Model Driven Architecture: Practice and Promise," ed: Addison-Wesley Longman Publishing Co., Inc., Boston, MA, 2003.

[17] S. Trujillo, D. Batory, and O. Diaz, "Feature oriented model driven development: A case study for portlets," in *Proceedings of the 29th international conference on Software Engineering*, 2007: IEEE Computer Society, pp. 44-53.

[18] S. J. Mellor, K. Scott, A. Uhl, and D. Weise, "Model-Driven Architecture," in *Advances in Object-Oriented Information Systems: OOIS 2002 Workshops Montpellier, France, September 2, 2002 Proceedings*, J.-M. Bruel and Z. Bellahsene Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2002, pp. 290-297.

[19] J. J. López-Fernández, E. Guerra, and J. De Lara, "Assessing the Quality of Meta-models," in *MoDeVVa 2014. Model-Driven Engineering, Verification and Validation. Proceedings of the 11th Workshop on Model-Driven Engineering, Verification and Validation co-located with 17th International Conference on Model Driven Engineering Languages and Systems (MODELS 2014) (Valencia, Spain, September 30, 2014)*, F. Boulanger, M. Famelis, and D. Ratiu, Eds., no. 1235, Aachen, Germany: CEUR-WS.org, 2014, pp. 3-12.

[20] R. Skjong and B. H. Wentworth, "Expert judgment and risk perception," in *Proceedings of the Eleventh (2001) International Offshore and Polar Engineering Conference (Stavanger, Norway, June 17-22, 2001)*: International Society of Offshore and Polar Engineers, 2001.

[21] A. Vázquez-Ingelmo, A. García-Holgado, F. J. García-Peñalvo, and R. Therón, "A Meta-Model Integration for Supporting Knowledge Discovery in Specific Domains: A Case Study in Healthcare," *Sensors*, vol. 20, no. 15, p. 4072, 2020. [Online]. Available: https://www.mdpi.com/1424-8220/20/15/4072.

[22] F. J. García-Peñalvo, "Education in knowledge society: A new PhD programme approach," in *Proceedings of the First International Conference on Technological Ecosystems for Enhancing Multiculturality (TEEM'13) (Salamanca, Spain, November 14-15, 2013)*, F. J. García-Peñalvo, Ed., New York, NY, USA: ACM, in ACM International Conference Proceeding Series (ICPS), 2013, pp. 575-577, doi: 10.1145/2536536.2536624.

[23] J. Brooke, "SUS-A quick and dirty usability scale," *Usability evaluation in industry*, vol. 189, no. 194, pp. 4-7, 1996.

[24] A. Vázquez-Ingelmo, A. García-Holgado, F. J. García-Peñalvo, and R. Therón, "Proof-of-concept of an information visualization classification approach based on their fine-grained features," *Expert Systems, e12872*, 2022, doi: 10.1111/exsy.12872.

[25] A. V. Pandey, A. Manivannan, O. Nov, M. Satterthwaite, and E. Bertini, "The persuasive power of data visualization," *IEEE transactions on visualization and computer graphics*, vol. 20, no. 12, pp. 2211-2220, 2014.

[26] A. Vázquez-Ingelmo, F. J. García-Peñalvo, and R. Therón, "Towards a Technological Ecosystem to Provide Information Dashboards as a Service: A Dynamic Proposal for Supplying Dashboards Adapted to Specific Scenarios," *Applied Sciences*, vol. 11, no. 7, art. 3249, 2021, doi: 10.3390/app11073249.

[27] A. Vázquez-Ingelmo, F. J. García-Peñalvo, and R. Therón, "Domain engineering for generating dashboards to analyze employment and employability in the academic context," in *Proceedings of the Sixth International Conference on Technological Ecosystems for Enhancing Multiculturality*, Salamanca, Spain, October 24th-26th 2018, New York, NY, USA: Association for Computing Machinery, 2018, pp. 896–901, doi: 10.1145/3284179.3284329. [Online]. Available: https://doi.org/10.1145/3284179.3284329

[28] A. Vázquez-Ingelmo, F. J. García-Peñalvo, and R. Therón, "Taking advantage of the software product line paradigm to generate customized user interfaces for decision-making processes: a case study on university employability," *PeerJ Computer Science*, vol. 5, e203, 2019, doi: 10.7717/peerj-cs.203.

[29] A. Vázquez-Ingelmo, J. Cruz-Benito, F. J. García-Peñalvo, and M. Martín-González, "Scaffolding the OEEU's Data-Driven Ecosystem to Analyze the Employability of Spanish Graduates," in *Global Implications of Emerging Technology Trends*, F. J. García-Peñalvo Ed. Hershey, PA, USA: IGI Global, 2018, pp. 236-255.

[30] F. Michavila, J. M. Martínez, M. Martín-González, F. J. García-Peñalvo, J. Cruz-Benito, and A. Vázquez-Ingelmo, *Barómetro de empleabilidad y empleo universitarios. Edición Máster 2017*. Madrid, España: Observatorio de Empleabilidad y Empleo Universitarios, 2018.

[31] F. Michavila, J. M. Martínez, M. Martín-González, F. J. García-Peñalvo, and J. Cruz Benito, "Empleabilidad de los titulados universitarios en España. Proyecto OEEU," *Education in the Knowledge Society*, vol. 19, no. 1, pp. 21-39, 2018, doi: 10.14201/eks20181912139.

[32] F. Michavila, J. M. Martínez, M. Martín-González, F. J. García-Peñalvo, and J. Cruz-Benito, *Barómetro de empleabilidad y empleo de los universitarios en España, 2015 (Primer informe de resultados)*. Madrid: Observatorio de Empleabilidad y Empleo Universitarios, 2016.

[33] P. A. Harris, R. Taylor, R. Thielke, J. Payne, N. Gonzalez, and J. G. Conde, "Research electronic data capture (REDCap)—A metadata-driven methodology and workflow process for providing translational research informatics support," *Journal of Biomedical Informatics*, vol. 42, no. 2, pp. 377-381, 2009/04/01/ 2009, doi: https://doi.org/10.1016/j.jbi.2008.08.010.

[34] J. W. Tukey, *Exploratory data analysis*. Reading, MA, 1977.

[35] J. I. Melero-Alegria *et al.*, "SALMANTICOR study. Rationale and design of a population-based study to identify structural heart disease abnormalities: a spatial and machine learning analysis," *BMJ Open*, vol. 9, no. 2, p. e024605, 2019, doi: 10.1136/bmjopen-2018-024605.

[36] A. Vázquez-Ingelmo *et al.*, "A platform to support the visual analysis of the SALMANTICOR study outcomes: conveying cardiological data to lay users," in *Proceedings TEEM'21. Ninth International Conference on Technological Ecosystems for Enhancing Multiculturality (Barcelona, Spain, October 27th – 29th, 2021)*, New York, NY, USA: Association for Computing Machinery, in ACM International Conference Proceedings Series, 2021, pp. 335–341, doi: 10.1145/3486011.3486471. [Online]. Available: https://doi.org/10.1145/3486011.3486471

[37] S. Berinato, *Good charts: The HBR guide to making smarter, more persuasive data visualizations*. Brighton, MA, USA: Harvard Business Review Press, 2016.

[38] A. Cairo, *The Functional Art: An introduction to information graphics and visualization*. San Francisco, CA, USA: New Riders, 2012.

[39] A. Cairo, *The truthful art: Data, charts, and maps for communication*. San Francisco, CA, USA: New Riders, 2016.

[40] C. O. Wilke, *Fundamentals of data visualization: a primer on making informative and compelling figures*. Sebastopol, CA, USA: O'Reilly Media, 2019.

[41] A. Vázquez-Ingelmo, A. García-Holgado, F. J. García-Peñalvo, and R. Therón, "Dashboard Meta-Model for Knowledge Management in Technological Ecosystem: A Case Study in Healthcare," *Proceedings*, vol. 31, no. 1, p. 44, 2019. [Online]. Available: https://www.mdpi.com/2504-3900/31/1/44.

[42] A. Vázquez Ingelmo, F. J. García-Peñalvo, and R. Therón Sánchez, "Aggregation Bias: A Proposal to Raise Awareness Regarding Inclusion in Visual Analytics," in *Trends and Innovations in Information Systems and Technologies, WorldCIST 2020*, Á. Rocha, H. Adeli, L. P. Reis, S. Costanzo, I. Orovic, and F. Moreira, Eds., 2020, vol. 3, Cham, Switzerland: Springer Nature, in Advances in Intelligent Systems and Computing Series Series, no. AISC 1161, doi: https://doi.org/10.1007/978-3-030-45697-9_40. [Online]. Available: http://hdl.handle.net/10366/148283

[43] A. Vázquez-Ingelmo, F. J. García-Peñalvo, and R. Therón, "Generating Dashboards Using Fine-Grained Components: A Case Study for a PhD Programme," in *Learning and Collaboration Technologies. Design, Experiences. 7th International Conference, LCT 2020, Held*

as Part of the 22nd HCI International Conference, HCII 2020, Copenhagen, Denmark, P. Zaphiris and A. Ioannou, Eds., July 19–24 2020, no. 12205, Cham, Switzerland: Springer Nature, in Lecture Notes in Computer Science, 2020, pp. 303-314, doi: 10.1007/978-3-030-50513-4_23.

[44] A. Vázquez-Ingelmo and R. Therón, "Beneficios de la aplicación del paradigma de líneas de productos software para generar dashboards en contextos educativos," *RIED. Revista Iberoamericana de Educación a Distancia*, vol. 23, no. 2, pp. 169-185, 07/01 2020, doi: 10.5944/ried.23.2.26389.

[45] A. Vázquez Ingelmo, A. García-Holgado, H. Hernández-Payo, F. J. García-Peñalvo, and R. Therón Sánchez, "Following up the progress of doctoral students and advisors' workload through data visualizations: a case study in a PhD program," *Proceedings of LASI-SPAIN 2021. Learning Analytics Summer Institute Spain 2021: Learning Analytics in times of COVID-19: Opportunity from crisis (Barcelona, Spain, July 7-9, 2021). CEUR Workshop Proceedings Series,* 2021. [Online]. Available: http://ceur-ws.org/Vol-3029/paper06.pdf.

[46] A. Vázquez-Ingelmo *et al.*, "A platform for management and visualization of medical data and medical imaging," in *Proceedings TEEM'20. Eighth International Conference on Technological Ecosystems for Enhancing Multiculturality (Salamanca, Spain, October 21st - 23rd, 2020)*, F. J. García-Peñalvo, Ed., New York, NY, USA: Association for Computing Machinery, in ACM International Conference Proceedings Series, 2020, pp. 518–522, doi: 10.1145/3434780.3436652. [Online]. Available: https://doi.org/10.1145/3434780.3436652

[47] F. J. García-Peñalvo *et al.*, "Application of Artificial Intelligence Algorithms Within the Medical Context for Non-Specialized Users: the CARTIER-IA Platform," *International Journal of Interactive Multimedia and Artificial Intelligence,* vol. 6, no. 6, pp. 46-53, 2021, doi: 10.9781/ijimai.2021.05.005.

[48] A. García-Holgado *et al.*, "User-Centered Design Approach for a Machine Learning Platform for Medical Purpose," in *HCI-COLLAB 2021*, Sao Paulo, Brazil, 8-10, September 2021, Cham, Switzerland: Springer International Publishing, in Human-Computer Interaction, pp. 237-249, doi: 10.1007/978-3-030-92325-9_18.

[49] A. Vázquez-Ingelmo *et al.*, "Bringing machine learning closer to non-experts: proposal of a user-friendly machine learning tool in the healthcare domain," in *Proceedings TEEM'21. Ninth International Conference on Technological Ecosystems for Enhancing Multiculturality (Barcelona, Spain, October 27th – 29th, 2021)*, New York, NY, USA: Association for Computing Machinery, in ACM International Conference Proceedings Series, 2021, pp. 324–329, doi: 10.1145/3486011.3486469. [Online]. Available: https://doi.org/10.1145/3486011.3486469

[50] R. Morgan, G. Grossmann, M. Schrefl, M. Stumptner, and T. Payne, "VizDSL: A Visual DSL for Interactive Information Visualization," Cham, 2018: Springer International Publishing, in Advanced Information Systems Engineering, pp. 440-455.

[51] I. Logre, S. Mosser, P. Collet, and M. Riveill, "Sensor data visualisation: a composition-based approach to support domain variability," in *European Conference on Modelling Foundations and Applications*, York, United Kingdom, R. J. Cabot J., Ed., 2014, vol. 8569: Springer, in Modelling Foundations and Applications. ECMFA 2014, pp. 101-116, doi: https://doi.org/10.1007/978-3-319-09195-2_7.

[52] T. Palpanas, P. Chowdhary, G. Mihaila, and F. Pinel, "Integrated model-driven dashboard development," *Information Systems Frontiers,* vol. 9, no. 2-3, pp. 195-208, 2007, doi: https://doi.org/10.1007/s10796-007-9032-9.

[53] M. Correll, E. Bertini, and S. Franconeri, "Truncating the y-axis: Threat or menace?," in *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, Honolulu, HI, USA, 2020: ACM, pp. 1-12, doi: 10.1145/3313831.3376222.

[54] S. L. Franconeri, L. M. Padilla, P. Shah, J. M. Zacks, and J. Hullman, "The Science of Visual Data Communication: What Works," *Psychological Science in the Public Interest*, vol. 22, no. 3, pp. 110-161, 2021, doi: 10.1177/15291006211051956.

[55] S. Few, *Information dashboard design*. Sebastopol, CA, USA: O'Reilly Media, 2006.

[56] C. Plaisant, "The challenge of information visualization evaluation," in *Proceedings of the working conference on Advanced visual interfaces*, 2004: ACM, pp. 109-116.

[57] A. Sarikaya, M. Correll, L. Bartram, M. Tory, and D. Fisher, "What Do We Talk About When We Talk About Dashboards?," *IEEE Transactions on Visualization and Computer Graphics,* vol. 25, no. 1, pp. 682-692, 2019, doi: 10.1109/TVCG.2018.2864903.

[58] A. Vázquez Ingelmo, A. García-Holgado, F. J. García-Peñalvo, and R. Therón Sánchez, "A Meta-modeling Approach to Take into Account Data Domain Characteristics and Relationships in Information Visualizations," in *Trends and Innovations in Information Systems and Technologies, WorldCIST 2021*, Azores, Portugal, Á. Rocha, H. Adeli, G. Dzemyda, F. Moreira, and A. M. R. Correia, Eds., 2021, vol. 2, Cham, Switzerland: Springer Nature, in Advances in Intelligent Systems and Computing Series Series, no. 1366, 2021, pp. 570-580, doi: 10.1007/978-3-030-72651-5_54. [Online]. Available: http://hdl.handle.net/10366/145626

[59] A. Satyanarayan, D. Moritz, K. Wongsuphasawat, and J. Heer, "Vega-Lite: A Grammar of Interactive Graphics," *IEEE Transactions on Visualization and Computer Graphics,* vol. 23, no. 1, pp. 341-350, 2017, doi: 10.1109/TVCG.2016.2599030.

[60] A. Vázquez-Ingelmo, F. J. García-Peñalvo, R. Therón, D. Amo Filvà, and D. Fonseca Escudero, "Connecting domain-specific features to source code: towards the automatization of dashboard generation," *Cluster Computing,* vol. 23, no. 3, pp. 1803-1816, 2020/09/01 2020, doi: 10.1007/s10586-019-03012-1.

[61] A. Vázquez-Ingelmo, F. J. García-Peñalvo, R. Therón, and A. García-Holgado, "Specifying information dashboards' interactive features through meta-model instantiation," in *LASI-SPAIN 2020. Learning Analytics Summer Institute Spain 2020: Learning Analytics. Time for Adoption?*, Valladolid, Spain, A. Martínez-Monés, A. Álvarez, M. Caeiro-Rodríguez, and Y. Dimitriadis, Eds., June 15-16 2020, Aachen, Germany: CEUR-WS.org, in CEUR Workshop Proceedings Series, no. 2671, pp. 47-59.

[62] M. Bostock, V. Ogievetsky, and J. Heer, "D³ Data-Driven Documents," *IEEE Transactions on Visualization and Computer Graphics,* vol. 17, no. 12, pp. 2301-2309, 2011, doi: 10.1109/TVCG.2011.185.

[63] A. Vázquez-Ingelmo, F. J. García-Peñalvo, and R. Therón, "Addressing Fine-Grained Variability in User-Centered Software Product Lines: A Case Study on Dashboards," in *Knowledge in Information Systems and Technologies*, Á. Rocha, H. Adeli, L. P. Reis, and S. Costanzo, Eds., 2019, vol. 1, Switzerland: Springer Nature, in Advances in Intelligent Systems and Computing, no. AISC 930, 2019, pp. 855-864, doi: 10.1007/978-3-030-16181-1_80.

[64] J. B. Warmer and A. G. Kleppe, *The object constraint language: getting your models ready for MDA*. Boston, MA, USA: Addison-Wesley Professional, 2003.