Vázquez-Ingelmo, A., García-Peñalvo, F. J., & Therón, R. (2020). Advances in the use of domain engineering to support feature identification and generation of information visualizations. In F. J. García-Peñalvo (Ed.), Proceedings TEEM'20. Eighth International Conference on Technological Ecosystems for Enhancing Multiculturality (Salamanca, Spain, October 21st - 23rd, 2020). ACM. https://doi.org/10.1145/3434780.3436640

Advances in the use of domain engineering to support feature identification and generation of information visualizations

Andrea Vázquez-Ingelmo[†]
GRIAL Research Group, Department of Computer Science, Research Institute for Educational Sciences, University of Salamanca, Salamanca, Spain
andreavazquez@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

Roberto Therón
GRIAL Research Group, Department of Computer Science, Research Institute for Educational Sciences, University of Salamanca, Salamanca, Spain
theron@usal.es

ABSTRACT

Information visualization tools are widely used to better understand large and complex datasets. However, to make the most out of them, it is necessary to rely on proper designs that consider not only the data to be displayed, but also the audience and the context. There are tools that already allow users to configure their displays without requiring programming skills, but this research project aims at exploring the automatic generation of information visualizations and dashboards in order to avoid the configuration process, and select the most suitable features of these tools taking into account their contexts. To address this problem, a domain engineering, and machine learning approach is proposed.

CCS CONCEPTS

• **Software and its engineering**; • **Reusability**; • **Human-centered computing**; • **Visualization toolkits**;

KEYWORDS

Automatic generation, Domain engineering, Meta-modeling, Information Dashboards, High-level requirements, Machine Learning

ACM Reference Format:
Andrea Vázquez-Ingelmo[†], Francisco José García-Peñalvo, and Roberto Therón. 2020. Advances in the use of domain engineering to support feature identification and generation of information visualizations. In *Eighth International Conference on Technological Ecosystems for Enhancing Multiculturality (TEEM'20), October 21–23, 2020, Salamanca, Spain.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3434780.3436640

1 INTRODUCTION

Information visualizations, infographics, dashboards, and any kind of medium that visually conveys information are part of every day's life. These tools support reasoning and decision-making processes through visual analysis.

The relevance of data visualization has grown over the years given the large amounts of data being continuously generating nowadays. These large quantities data ask for methodologies that enable people to understand, explore and process different data dimensions.

While data visualization provides a basis to generate knowledge, it is not miraculous. It is still necessary to account for the data context, the audience skills, beliefs and other several quantity of factors.

That is why research in visualization adaptation and generation of the best kind of dashboard features depending on the user and/or context is gaining relevance [1-9].

This research focuses on the application of conceptual frameworks such as meta-modeling and domain engineering to tackle the automatic generation and adaptation of information visualizations and information dashboards relying on their most fine-grained shared features.

2 HYPOTHESIS AND OBJECTIVES

The main hypothesis of this work is the following [10]:

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.

This hypothesis tries to answer the question on how the features of a visualization tool can modify the perception of a user and the insights reached through them.

However, to test this hypothesis it is necessary to identify the common and fine-grained features of information visualizations and dashboards. For these reasons, as we will explain in the remaining sections, the first objective of this research is to identify these common characteristics and arrange them into a meta-model that will allow the decomposition of these tools into abstract elements and components.

The next objective is to obtain a framework for the automatic generation of dashboards by varying the meta-model parameters and elements. This framework takes a Software Product Line (SPL) engineering approach, which ensures reutilization and flexibility when it comes to tailor products and systems.

Finally, by obtaining this framework, it will be possible to automatically generate dashboards within different domains and contexts. This will allow the study of how varying different features could impact the user efficiency and insights when visually analyzing data.

The outcomes of this process will enable the testing of the posed hypothesis, as well as the application of machine learning algorithms to learn from the performance of users depending on the selected visualization features. This will set the basis for obtaining a machine learning-driven approach to generate adapted and tailored information visualizations and dashboards.

3 METHODOLOGY

3.1 Domain engineering

As the title suggests, the main methodology employed for this project is domain engineering. Domain engineering [11] is referred to the thorough study of a specific domain in order to extract shared concepts or commonalities.

In the case of information systems, taking this approach means to search form common features and variability points among different software products that belong to the same domain or "family". In fact, the domain engineering phase is the first step to build a software product line [12, 13], as it will create the foundation of the family of products that will be built.

It is important to deeply understand the domain in which the product line will be framed, because identifying proper features, in a proper level of granularity is crucial to take advantage of this paradigm.

This methodology has been chosen to carry out the present project because dashboards and information visualizations vary widely, but they always share primitive and abstract elements, such as encodings, structures and visual elements [14, 15]. Combined with a meta-modeling approach, we have identified and arranged high-level features of these tools into a meta-model.

3.2 Meta-modeling

Meta-models are the backbone of model-driven paradigms [16-18]. These artefacts allow the mapping of abstract entities to more specific and detailed entities.

The Object Management Group (OMG) proposes the model-driven architecture (MDA) as a framework to implement this approach. The proposed architecture provides different guidelines for software development by employing abstract models to describe and define the target system [19].

Moreover, 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).

Using this approach along with the domain engineering methodology enables the identification of crucial commonalities in the dashboards domain, thus obtaining a generic skeleton to generate these powerful tools. Specifically, the dashboard meta-model is an instance of MOF (i.e., an M2-model), so it can be instantiated to obtain M1-models [20].

3.3 Machine Learning

One of the main goals of identifying structuring dashboard features into fixed relationships and shared concepts is to subsequently apply machine learning algorithms to them.

Machine learning allows the automatic recognition of patterns in data and the application of the derived knowledge to generalize problems and predict values based on those learned patterns.

This methodology has been chosen to carry out the present project because dashboards and information visualizations vary widely, but they always share primitive and abstract elements, such as encodings, structures and visual elements [14, 15].

Applying these algorithms could yield very interesting results, such as which configuration fits better the target audience, or which features are more important for a dashboard or information visualization given its context of application.

4 DISSERTATION STATUS

The first outcome of this work was a Systematic Literature Review (SLR) [21] which set the conceptual basis for the subsequently step. In this SLR, the authors identified and classified different methods for tailoring dashboards. The SPL paradigm and domain engineering approach was among these tailoring methodologies, but there was not any combined approach that mixed domain engineering approaches with machine learning algorithms.

In this section, the rest of the milestones achieved during the development of this dissertation will be presented.

4.1 Meta-model

As stated in the research objectives, the first step was to identify the common features existing within the dashboard domain.

These features are very fine-grained, and involve very primitive and conceptual elements such as scales, encodings, visual marks, etc.

By taking a model-driven approach, it has been possible to arrange all these fine-grained features that are common to information dashboards into a meta-model, including concepts regarding the users, such as their information goals and the visualization tasks needed to achieve them [23].

Figures 1 and 2 depict the outcome of this conceptualization process, which is an Ecore version of the proposed meta-model [22, 24].

4.2 Software Product Line

Once the dashboard features were identified, it has been possible to develop a programming framework to automatically generating information dashboards given a configuration file.

This framework is based on the software product line paradigm [12, 13], and it relies on core software assets that are combined to provide a final product that fits the input configuration.

A template-based approach has been taken to combine all these small software pieces (i.e., variability points in the SPL context [25-28]). This approach enables the configuration of fine-grained features, which fits the requirements of the dashboard domain.

A Python-based code generator (which relies in the Jinja2 template engine [29]) has been designed to parse the meta-model-compliant configurations and inject the code associated to each primitive feature.

The outcomes of the generator are a set of HTML and JavaScript files which hold the code for the final product.

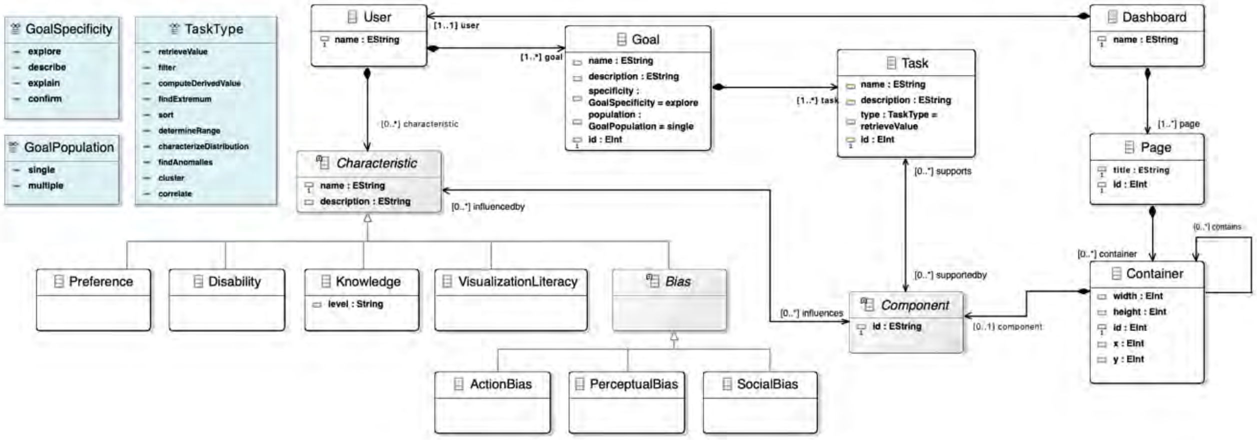

Figure 1: Overview of the Ecore version of the dashboard meta-model [22].

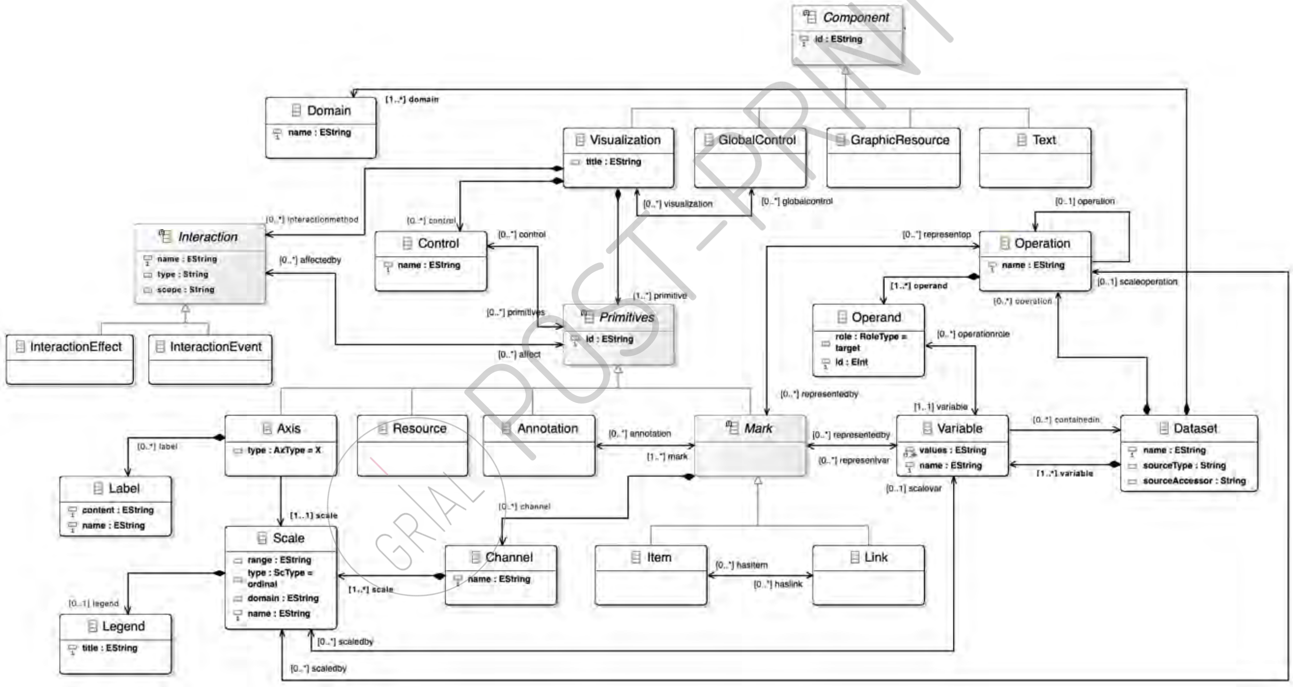

Figure 2: Detailed view of the dashboards' components primitive elements [22].

4.3 Machine Learning application

One of the strengths of using the previously depicted approaches is the possibility of connecting the dashboard generation pipeline to external algorithms that could infer the best features to configure.

Currently, different proofs of concept are being developed to validate the viability of applying machine learning algorithms to the configuration files. The main goal is to obtain a model to recommend the best configuration for an information dashboard or classifying information visualizations based on their most primitive features.

5 CONCLUSIONS

The proposed dissertation focuses on addressing information visualization and decision-making related issues, with the goal of obtaining a context-aware framework for generating information dashboards.

The reached milestones of this project include: a systematic literature review to gain knowledge about the dashboard domain and tailoring methodologies, a dashboard meta-model that captures high-level features and properties of these tools and a first version of a software product line to automatically generating code by using configurations that are compliant to the meta-model.

The next steps will involve the connection of external algorithms, such as machine learning models, to guide the generation pipeline with the goal of improving the effectiveness of the generated 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).

REFERENCES

[1] Kanit Wongsuphasawat, Dominik Moritz, Anushka Anand, Jock Mackinlay, Bill Howe, and Jeffrey Heer. 2015. Voyager: Exploratory analysis via faceted browsing of visualization recommendations. *IEEE transactions on visualization and computer graphics* 22, 1, 649-658.

[2] Belgin Mutlu, Eduardo Veas, and Christoph Trattner. 2016. Vizrec: Recommending personalized visualizations. *ACM Transactions on Interactive Intelligent Systems* 6, 4, 31.

[3] Kevin Hu, Michiel A Bakker, Stephen Li, Tim Kraska, and César Hidalgo. 2019. VizML: A Machine Learning Approach to Visualization Recommendation. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems* ACM, 128.

[4] Manasi Vartak, Silu Huang, Tarique Siddiqui, Samuel Madden, and Aditya Parameswaran. 2017. Towards visualization recommendation systems. *ACM SIGMOD Record* 45, 4, 34-39.

[5] Pawandeep Kaur and Michael Owonibi. 2017. A Review on Visualization Recommendation Strategies. In *12th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP 2017)* SCITEPRESS, 266-273.

[6] Martin Voigt, Stefan Pietschmann, Lars Grammel, and Klaus Meißner. 2012. Context-aware recommendation of visualization components. In *The Fourth International Conference on Information, Process, and Knowledge Management (eKNOW)*Citeseer, 101-109.

[7] Petra Kubernátová, Magda Friedjungová, and Max Van Duijn. 2018. Constructing a Data Visualization Recommender System. In *International Conference on Data Management Technologies and Applications* Springer, 1-25.

[8] David Gotz and Zhen Wen. 2009. Behavior-driven visualization recommendation. In *Proceedings of the 14th international conference on Intelligent user interfaces* ACM, 315-324.

[9] Ainhoa Álvarez-Arana, M Larrañaga-Olagaray, and M Villamañe-Gironés. 2020. Mejora de los procesos de evaluación mediante analítica visual del aprendizaje.

[10] Andrea Vázquez-Ingelmo, Francisco J García-Peñalvo, and Roberto Therón. 2019. Automatic generation of software interfaces for supporting decision-making processes. An application of domain engineering and machine learning. In *Proceedings of the Seventh International Conference on Technological Ecosystems for Enhancing Multiculturality*, 1007-1011.

[11] Guillermo Francisco Arango. 1988. Domain engineering for software reuse.

[12] Paul Clements and Linda Northrop. 2002. *Software product lines.* Addison-Wesley, Boston, MA, USA.

[13] 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.

[14] Colin Ware. 2012. *Information visualization: perception for design.* Elsevier.

[15] Tamara Munzner. 2014. *Visualization analysis and design.* AK Peters/CRC Press, Boca Raton, FL, USA.

[16] 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.

[17] 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.

[18] 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.

[19] Stephen J. Mellor, Kendall Scott, Axel Uhl, and Dirk Weise. 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:10.1007/3-540-46105-1_33.

[20] J. M. Álvarez, A. Evans, and P. Sammut. 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:10.1007/3-540-45441-1_4.

[21] Andrea Vázquez-Ingelmo, Francisco José García-Peñalvo, and Roberto Therón. 2019. Information Dashboards and Tailoring Capabilities - A Systematic Literature Review. *IEEE Access* 7, 109673-109688. DOI:10.1109/ACCESS.2019.2933472.

[22] Andrea Vázquez-Ingelmo, Alicia García-Holgado, Francisco José García-Peñalvo, and Roberto Therón. 2020. A Meta-Model Integration for Supporting Knowledge Discovery in Specific Domains: A Case Study in Healthcare. *Sensors* 20, 15, 4072.

[23] Andrea Vázquez-Ingelmo, Francisco José García-Peñalvo, Roberto Therón, and Miguel Ángel Conde. 2020. Representing Data Visualization Goals and Tasks Through Meta-Modeling to Tailor Information Dashboards. *Applied Sciences* 10, 7, 2306.

[24] Andrea Vázquez-Ingelmo, Alicia García-Holgado, Francisco José García-Peñalvo, and Roberto Therón. 2019. Dashboard Meta-Model for Knowledge Management in Technological Ecosystem: A Case Study in Healthcare. In *UCAmI 2019*, Mdpi Ed. MDPI, Toledo, Castilla-La Mancha, Spain. DOI: https://doi.org/10.3390/proceedings2019031044.

[25] Andreas Metzger and Klaus Pohl. 2007. Variability management in software product line engineering. In *Companion to the proceedings of the 29th International Conference on Software Engineering* IEEE Computer Society, 186-187.

[26] Seung-Hyun Heo and Eun Man Choi. 2006. Representation of variability in software product line using aspect-oriented programming. In *Software Engineering Research, Management and Applications, 2006. Fourth International Conference on* IEEE, 66-73.

[27] Jilles Van Gurp, Jan Bosch, and Mikael Svahnberg. 2001. On the notion of variability in software product lines. In *Software Architecture, 2001. Proceedings. Working IEEE/IFIP Conference on* IEEE, 45-54.

[28] Christoph Seidl, Ina Schaefer, and Uwe Aßmann. 2014. Capturing variability in space and time with hyper feature models. In *Proceedings of the Eighth International Workshop on Variability Modelling of Software-Intensive Systems* ACM, 6.

[29] Armin Ronacher. 2008. *Jinja2 Documentation.*Jinja2.