

# Generation of Customized Dashboards Through Software Product Line Paradigms to Analyse University Employment and Employability Data

Andrea Vázquez-Ingelmo<sup>1</sup>[0000-0002-7284-5593], Francisco J. García-Peñalvo<sup>1</sup>[0000-0001-9987-5584] and Roberto Therón<sup>1,2</sup>[0000-0001-6739-8875]

<sup>1</sup> GRIAL Research Group, Computer Sciences Department, Research Institute for Educational Sciences, University of Salamanca, Salamanca, Spain

<sup>2</sup> VisUSAL Research Group, University of Salamanca, Salamanca, Spain  
{andreavazquez, fgarcia, theron}@usal.es

**Abstract.** University employment and, specifically, employability has steadily gained relevance nowadays as the study of these fields can lead to improvement in the quality of life of individual citizens. However, the empirical research is still insufficient to make significant decisions within this domain. It is necessary to rely on powerful tools in order to reach insights about university employment and employability. Information dashboards have become a key software tool to reach insights and make informed decisions about a specific topic, domain or field of study. Nevertheless, dashboards' users can have several requirements that differ from each other (including displayed information itself, design features or even functionalities), and it is necessary to take into account all of these specifications, allowing users to exploit data with its own necessities and aiming to its own goals. Applying software product line paradigms, it is plausible to face different requirements regarding information dashboards' development in an efficient, scalable and maintainable way. To validate this approach, a case study is presented in the context of the Spanish Observatory of Employability and University Employment, an organization that aims to become an information reference for these fields.

**Keywords:** University Employment; University Employability; Domain Engineering; SPL; Information Dashboards; Information Systems.

## 1 Introduction

There are certain fields of study that are gaining relevance over the years as they can enhance access to employment, prevent unemployment and improve job quality, elements that are part of the main concerns in society nowadays. Employability is becoming a key pillar to explain the employment situation of individuals. However, this research area has not yet a strong theoretical foundation given the absence of agreement regarding the definition of employability, and, consequently, the complexity of acquiring indicators to evaluate it. Employability factors can vary depending on the research

perspective used and socioeconomic context, so it is necessary to take into account several variables to analyze this theoretical construct, from identifying the competencies that individuals would need in their career to even sociodemographic variables (taking a “broad” approach [1]).

Universities have a key role related to the employability of individuals (specifically, of their students) [2], as they are acquiring a series of skills that could improve their capacities to obtain a job. Recollecting employment and employability data in the academic context can help to reach insights about the linkage between the university training and the working career of the graduates. Research on these fields could offer methods for individuals to identify the factors that affect their employability and, consequently, their career path.

However, as it has been aforementioned, there are several variables that can be involved in the research of the student’s employability and employment. The recollection of employability and employment data is crucial, but the possession of large data volumes is not valuable by itself; it acquires real value when it is analyzed and exploited with the goal of extracting knowledge from it [3]. Data analysis has gained relevance in different sectors through data-driven [4] approaches with the vision of conducting well-informed knowledge management [5, 6] and decision-making processes [7].

In the academic context, this kind of processes could help policymakers and institutions to improve and promote the most influential factors that affect the employability and employment of the students, placing it in the scope of emergent areas like the Academic Analytics [8-12] or Institutional Intelligence [13, 14].

For that reason and given the social implications that these decision-making processes could have, it is crucial to count on powerful tools that allow decision makers to reach insights about certain fields of study (in this case university employability and employment) and to support their decisions with an informed foundation.

Information dashboards constitute one of the most commonly used software products for knowledge extraction and datasets exploitation [15], being composed by a series of graphic resources and metrics with the goal of showing information in an understandable manner [16] and allowing the identification of relevant patterns and indicators for making decisions.

But the development of information dashboards is not a trivial process. Not every decision maker has the same requirements and aims to the same goals. What is more, during the decision-making processes different profiles could be involved, generating communication gaps as they could not have the same knowledge regarding their specialties [17]. It is then, important to take into account the requirements of the target users (i.e. the users that will make use of the dashboard). However, developing a dashboard for every profile encountered could be a very complex and time-consuming task. In addition, not only the developing process but also the maintenance of the dashboards will end up being barely an impossible and not scalable job.

Software engineering paradigms like software product lines (SPL) [18, 19] offer potential solutions for managing diverse sets of requirements, focusing in the reutilization and combination of base software assets (core assets) to improve scalability and maintainability, as well as providing a systematic framework for efficiently developing software products. Given the benefits of software product lines, applying this approach

could be a potential solution for generating customized and flexible dashboards based on the users' requirements.

With regard to all these considerations, a proposal to automate the generation of customized dashboards for analyzing employability and employment data (in the context of the Observatory of Employability and Employment, OEEU, described in Section 2) is made through the design of a domain specific language (DSL) to control all the features and parameters of the dashboard product family that fuels a template-based source code generator.

The rest of this paper is structured as follows: Section 2 describes the issues faced by the OEEU regarding the dissemination of their studies' results and the personalization of its products for each user, followed by Section 3, in which the methodology used is described. Section 4 summarizes the results obtained by the application of the SPL paradigm to the OEEU ecosystem, followed by section 5, where these results are discussed. Finally, section 6 offers the conclusions derived from this work and future research lines.

## 2 Context

The work presented in this paper has its motivation behind the problems and issues regarding data analysis and visualization faced by the Observatory for University Employment and Employability (also known as OEEU, its Spanish acronym, <http://oeeu.org>). This organization aims to become an information reference for understanding and exploiting knowledge about employment and employability of students from Spanish universities, by conducting studies about these fields in the academic context [20-22].

To do so, the Observatory takes a data-driven approach to recollect, analyze, visualize and disseminate employment and employability data of graduates from Spanish universities. During the recollection process universities send their administrative data records, and then, their students answer a questionnaire in which they are asked about their sociodemographic context, their skills levels, their career path, etc. At the end of this process, the bank of knowledge of the Observatory counts on a significant set of variables from the students' sample. For instance, in the 2015 study edition more than 500 variables were collected from 13006 bachelor students. On the other hand, in the 2017 study edition 376 variables were collected from 6738 master degree students.

However, the volume of the data collected makes the presentation of the study results to the ecosystem's users (i.e. administrators, Spanish universities and general users) a challenge. The presentation of the results is not trivial, as the users have different requirements and permission levels to access the study data.

These are the very reasons why an approach based on domain engineering fits the OEEU necessities, as it will allow developers to efficiently generate customized dashboards to present the studies' results, meeting different requirements.

### 3 Methodology

As it has been aforementioned, the core of this work is the application of the software product lines (SPL) paradigm. To put it into practice, there are two main phases: the domain engineering phase and the application engineering phase.

In the domain engineering phase, the abstract features of the domain are identified. This process focuses on identifying the commonalities and variability of the different products (dashboards, in this case) belonging to the line. Commonalities and variability are modelled through feature models [23], offering a structured diagram with the mandatory, optional or alternative features that a product of the line could have. Taking into account the results of this modelling task, the core assets for the software product line are implemented. These core assets compose the foundation of the product line. They will materialize the variability points [24-26] specified in the feature model, so they can be configured and reutilized in specific products of the line based on the requirements of the stakeholders.

For example, for the software product line of dashboards designed for the Observatory, the top-level feature model is showed in the Fig. 1. Only this level of the feature model is showed given its magnitude. To sum up, a dashboard has a mandatory feature, which is the base system that gives support to the product line. Dashboards will have at least one page in which different visual and control components can be displayed: scatter diagrams, heat maps, chord diagrams or a data filter. Every individual component will have its own functionalities, which can be mandatory, optional or alternative.

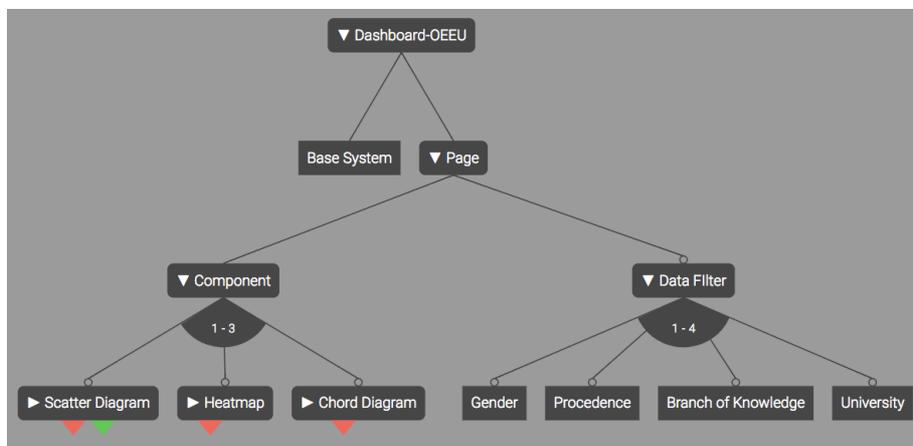


Fig. 1. Top-level feature model of the SPL for the Observatory's dashboards

Once this phase is complete, the application engineering phase starts. During this phase, the core assets previously implemented are retrieved and configured to meet the requirements of the final product being implemented. The configuration process could be automated to improve the efficiency of the generation of specific products of the software line [27-29].

This automatic generation was made in this work through a template-based code generator implemented in Python and Jinja2 [30]. Every component and functionality of the dashboard product line is implemented in templates in which the different features are injected according to the requirements through the definition of macros.

To specify the configuration of the dashboard being generated, and consequently, to inform the code generator which functionalities should add to the different components, a domain specific language (DSL) was implemented through XML technology [31]. This DSL is based on the feature model obtained during the domain engineering phase, and it fuels the code generator to allow the configuration of the product according to the requirements specified in the DSL.

The outcomes of the code generator (Fig. 2), in this case, are the different JavaScript and HTML files that will compose the final dashboard. These files are automatically deployed on the folder from which the dashboard will be served to the specific user which it has been generated for.

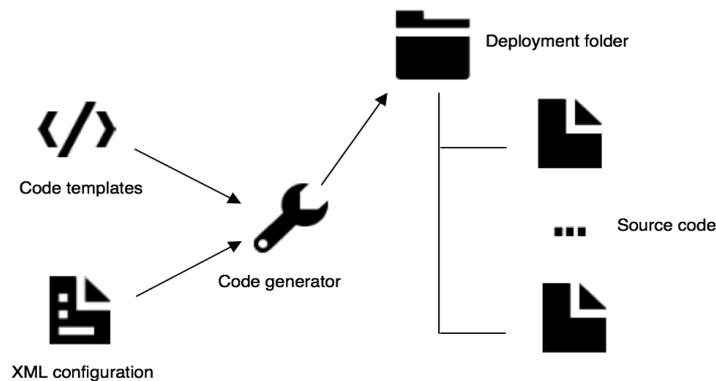


Fig. 2. Outline of the code generator inputs and outputs.

## 4 Results

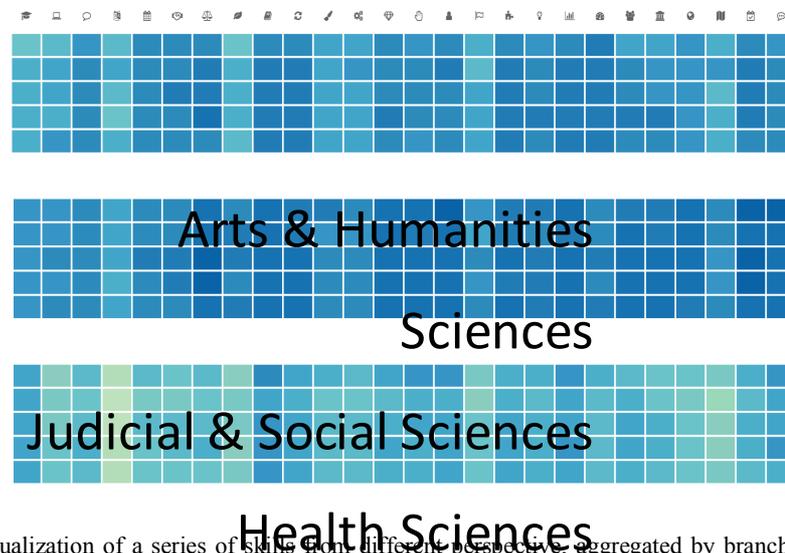
The application of this paradigm along with the automatic generation of code has brought valuable results in terms of development time, maintainability and scalability of the OEEU's dashboards.

The configuration of the components through XML files makes the personalization of dashboards a straightforward task, being the specification of the requirements the difficult part. Once the requirements are collected from the stakeholders, it is only necessary to specify them in terms of the DSL designed and execute the code generator to provide the dashboard.

To illustrate the results, a series of examples of the potential of this approach are described. One of the main parts of the Observatory studies are the evaluation of a series of skills from different perspectives. Students give a score to the level of the skills they felt they have, the level they felt they acquired during their studies and the level they

felt they required during their jobs. The visualization of these results was previously made through bar charts [20, 21], as they are simple and a common way to visualize this kind of variables.

However, this kind of visualization makes difficult the comparison of the different perspectives (possessed skills, skills acquired during the studies and skills required in job), so another method to visualize these variables was necessary. The representation of the skills through a heat map was proposed [32], as it allows the identification of values that stand out through a color codification (Fig. 3). Most intense colors represent greater values than the less intense ones. This visualization could help decision makers to identify at first sight, which branch of knowledge lack different competencies, for example.



**Fig. 3.** Visualization of a series of skills from different perspectives, aggregated by branch of knowledge of the studies pursued (top section shows required skills in job, middle section the possessed skills and the last and bottom section, the contribution of the studies to the skill).

To materialize the previous heat map, the configuration showed in the Fig. 4 was specified. Through the DSL, it is possible to specify that a heat map component is required with three dimensions, one for each skills' perspective. Each dimension has a particular data source (in this case, the Observatory's API codes to retrieve the necessary information about the students' skills).

The DSL also allow developers to specify specific functionalities, like the possibility to export the visualization as an image or different controls (data selectors to explore different data aggregations or data filters).

```

<Heatmap component_id="HeatMap_2">
  <Exportation>True</Exportation>
  <Dimensions>
    <Dimension dimension_id="1">
      <DataResource>
        <DataSource>
          <label>Possessed</label>
          <code>COMPP</code>
          <metric_code>oneLevelGroupedAverage</metric_code>
          <endpoint>stats2015</endpoint>
        </DataSource>
      </DataResource>
    </Dimension>
    <Dimension dimension_id="2">
      <DataResource>
        <DataSource>
          <label>Degree's contribution</label>
          <code>COMPU</code>
          <metric_code>oneLevelGroupedAverage</metric_code>
          <endpoint>stats2015</endpoint>
        </DataSource>
      </DataResource>
    </Dimension>
    <Dimension dimension_id="3">
      <DataResource>
        <DataSource>
          <label>Required in job</label>
          <code>COMPE</code>
          <metric_code>oneLevelGroupedAverage</metric_code>
          <endpoint>stats2015</endpoint>
        </DataSource>
      </DataResource>
    </Dimension>
  </Dimensions>
  <Controls type="Bar">
    <CollapseButton>True</CollapseButton>
    <DataSelectors>
      <Category>

```

Fig. 4. Example of the configuration for the component presented in the Fig. 3.

This heat map is a core asset of the SPL, so it can be reutilized for other purposes. For example, the representation of the most used methods to search job can be also done through the heat map (Fig. 5), being only necessary to specify in the XML configuration files the new data source for this component (Fig. 6).

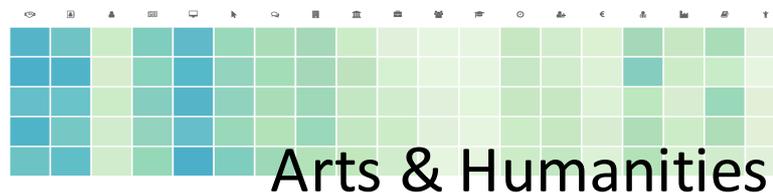


Fig. 5. Visualization of the frequency of use of a series of job search methods, aggregated by branch of knowledge of the studies pursued.

```

<Heatmap component_id="HeatMap_1">
  <Exportation>True</Exportation>
  <Dimension dimension_id="1">
    <DataResource>
      <DataSource>
        <label>Used method to search job</Label>
        <code>E2BUS01</code>
        <metric_code>oneLevelGroupedPercentage</metric_code>
        <endpoint>stats2017</endpoint>
      </DataSource>
    </DataResource>
  </Dimension>
  <Controls type="Panel">
    <OverviewPanel>True</OverviewPanel>
    <DataFilters>
      <Title>Data Filters</Title>
      <FilterGroup>
        <DataResource>
          <xi:include href="../include/filtros/filtro-sexo"/>
          <xi:include href="../include/filtros/filtro-ccaa.xml"/>
        </DataResource>
      </FilterGroup>
    </DataFilters>
  </Controls>
</Heatmap>

```

Fig. 6. Example of the configuration for the component presented in the Fig. 5.

On the other hand, a user could rather to explore this data with a scatter plot in order to find patterns and relations between the variables collected. In this case, the scatter diagram component would be selected during the configuration of the product to have a new perspective. Or even a set of components could be part of the same page (Fig. 7) to allow comparisons and different perspectives simultaneously, obtaining the final dashboard presented in the Fig. 8.

```

<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>
      <ColumnGroup>
        <Column width="100%" height="100%">
          <Component ref="Chord_1"/>
        </Column>
        <Column width="100%" height="100%">
          <Component ref="ScatterDiagram_1"/>
        </Column>
      </ColumnGroup>
      <Row width="100%" height="100%">
        <Component ref="HeatMap_1"/>
      </Row>
    </RowGroup>
  </Layout>
</Page>

```

Fig. 7. Example of the configuration for the whole dashboard presented in the Fig. 8.

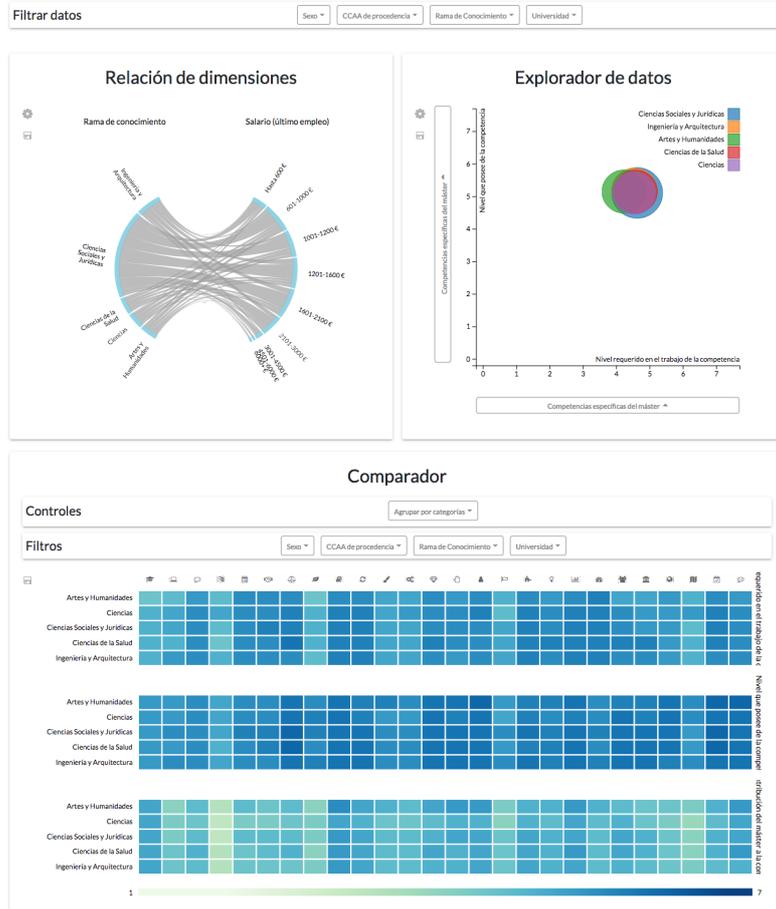


Fig. 8. Example of a dashboard generated given a particular XML configuration (contents in Spanish).

To accomplish the previous visualizations, it has only been necessary to change the XML configuration for a target user and regenerate the code automatically through the code generator.

## 5 Discussion

Applying the software product line paradigm in the context of the Observatory for University Employment and Employability has proved to be an interesting approach to manage different and dynamic requirements among users with varied profiles.

Once the core assets are implemented, the reconfiguration and maintainability of the code that gives support to the dashboards is straightforward. Adding new components

from scratch to the product line is also easy, as it does not affect the rest of the features previously implemented.

This pilot framework for generating this kind of software products given a set of requirements could lead to the construction of more effective dashboards. The tough part in the design of a dashboard is not the implementation, although it is, obviously, also a time-consuming process. The main problem in the design of dashboards is the recollection and specification of requirements from the users that will end up employing the final product. Dashboards are not a set of visualizations and graphical resources that show arbitrary data; they need to seek for a goal that could give support to decision-making processes.

This kind of approach could ease the refinement of the dashboards implemented, as it allows the evolution of requirements in a smooth and effective way. It could also help testing different configurations of the dashboards to find the right one for every profile involved without consuming significant resources, or even add the capacity of generating dashboards in different languages on demand.

Domains continuously evolving like employment and employability or academic analytics could benefit from this paradigm by having powerful and adaptive visualization tools to support decision-makers and policy-makers.

In this case, new components could be added to the software product line or new product could be configured in order to exploit and explore new perspectives of employability and employment in the academic context.

There are some challenges, however, to face. The automatic generation of user interfaces is not trivial [33-35]. Interfaces not only need to be functional but also usable, and modelling usability is a major challenge. Usability could lead to a good or a bad experience with the dashboard, no matter how many functionalities it has. Further research will involve usability tests on the automatically generated dashboards.

Another challenge are data sources. Data is the fuel of dashboards and sources need to be well integrated in order to maintain interoperability and scalability, as data sources can be heterogeneous and can be presented in different formats.

## **6 Conclusions**

The software product line approach has been applied to the Spanish Observatory for University Employment and Employability to manage the presentation of their studies' results taking into account different requirements. This approach allows the automatic generation of dashboards by composing different visualizations with different functionalities that consume from a variety of data sources.

Data visualization is important to reach insights about different fields (in this case, university employment and employability), but it is crucial to take into account the requirements of the final users of the dashboards, as they will be the ones involved in decision-making processes.

Providing powerful tools and frameworks to visualize this kind of data could help policy-makers and institutions to identify areas to improve or promote and report more benefits by having well-informed foundations.

**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. McQuaid, R.W., Lindsay, C.: The concept of employability. *Urban studies* 42, 197-219 (2005)
2. García-Peñalvo, F.J.: The Third Mission. *Education in the Knowledge Society* 17, 7-18 (2016)
3. Albright, S.C., Winston, W., Zappe, C.: *Data analysis and decision making*. Cengage Learning (2010)
4. Patil, D., Mason, H.: *Data Driven*. "O'Reilly Media, Inc." (2015)
5. Fidalgo-Blanco, Á., Sein-Echaluce, M.L., García-Peñalvo, F.J.: Knowledge spirals in higher education teaching innovation. *International Journal of Knowledge Management* 10, 16-37 (2014)
6. Fidalgo-Blanco, Á., Sein-Echaluce, M.L., García-Peñalvo, F.J.: 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, 266-288 (2015)
7. Sharda, R., Delen, D., Turban, E.: *Business intelligence: a managerial perspective on analytics*. Prentice Hall Press (2013)
8. Baepfer, P., Murdoch, C.J.: Academic analytics and data mining in higher education. *International Journal for the Scholarship of Teaching and Learning* 4, 17 (2010)
9. Bichsel, J.: Analytics in higher education: Benefits, barriers, progress, and recommendations. *EDUCAUSE Center for Applied Research* (2012)
10. Campbell, J.P., DeBlois, P.B., Oblinger, D.G.: Academic Analytics. A new tool for a new era. *Educause Review* 42, 40-42,44,46,48,50,52,54,56-57 (2007)
11. Gómez-Aguilar, D.A., García-Peñalvo, F.J., Therón, R.: Analítica Visual en eLearning. *El Profesional de la Información* 23, 236-245 (2014)
12. Michavila, F., Martín-González, M., Martínez, J.M., García-Peñalvo, F.J., Cruz-Benito, J.: Analyzing the employability and employment factors of graduate students in Spain: The OEEU Information System. 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. 277-283. ACM, New York, USA (2015)
13. Oficina de Cooperación Universitaria: *Libro Blanco Inteligencia Institucional en Universidades*. OCU (Oficina de Cooperación Universitaria), Madrid, España (2013)
14. García Peñalvo, F.J.: *Inteligencia Institucional para la Mejora de los Procesos de Enseñanza-Aprendizaje*. (2015)
15. Few, S.: *Information dashboard design*. (2006)
16. Tufte, E., Graves-Morris, P.: *The visual display of quantitative information*.; 1983. (2014)
17. Eppler, M.J.: Knowledge communication problems between experts and decision makers: An overview and classification. *Electronic Journal of Knowledge Management* 5, (2007)
18. Clements, P., Northrop, L.: *Software product lines*. Addison-Wesley (2002)

19. Gomaa, H.: Designing Software Product Lines with UML: From Use Cases to Pattern-Based Software Architectures. Addison Wesley Longman Publishing Co., Inc. (2004)
20. 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)
21. 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)
22. Michavila, F., Martínez, J.M., Martín-González, M., García-Peñalvo, F.J., Cruz Benito, J.: Empleabilidad de los titulados universitarios en España. Proyecto OEEU. Education in the Knowledge Society 19, 21-39 (2018)
23. Kang, K.C., Cohen, S.G., Hess, J.A., Novak, W.E., Peterson, A.S.: Feature-oriented domain analysis (FODA) feasibility study. Carnegie-Mellon Univ Pittsburgh Pa Software Engineering Inst (1990)
24. Metzger, A., Pohl, K.: Variability management in software product line engineering. In: Companion to the proceedings of the 29th International Conference on Software Engineering, pp. 186-187. IEEE Computer Society, (2007)
25. Van Gurp, J., Bosch, J., Svahnberg, M.: On the notion of variability in software product lines. In: Software Architecture, 2001. Proceedings. Working IEEE/IFIP Conference on, pp. 45-54. IEEE, (2001)
26. Gacek, C., Anastasopoulos, M.: Implementing product line variabilities. In: ACM SIGSOFT Software Engineering Notes, pp. 109-117. ACM, (2001)
27. Tajali, S.B., Coriveau, J.-P., Shi, W.: A Template-Based Approach to Modeling Variability. In: Proceedings of the International Conference on Software Engineering Research and Practice (SERP), pp. 1. The Steering Committee of The World Congress in Computer Science, Computer Engineering and Applied Computing (WorldComp), (2013)
28. Greifenberg, T., Müller, K., Roth, A., Rumpe, B., Schulze, C., Wortmann, A.: Modeling variability in template-based code generators for product line engineering. arXiv preprint arXiv:1606.02903 (2016)
29. Rodger, R.J.: Jostraca: a template engine for generative programming. In: European Conference for Object-Oriented Programming. Citeseer, (2002)
30. Ronacher, A.: Jinja2 Documentation. Welcome to Jinja2—Jinja2 Documentation (2.8-dev) (2008)
31. Bray, T., Paoli, J., Sperberg-McQueen, C.M., Maler, E., Yergeau, F.: Extensible markup language (XML). World Wide Web Journal 2, 27-66 (1997)
32. Harvard Business Review, <https://hbr.org/2016/09/we-need-a-better-way-to-visualize-peoples-skills>
33. 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)
34. Kramer, D., Oussena, S., Komisarczuk, P., Clark, T.: Graphical user interfaces in dynamic software product lines. In: Product Line Approaches in Software Engineering (PLEASE), 2013 4th International Workshop on, pp. 25-28. IEEE, (2013)
35. 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-Volume 1, pp. 6-10. ACM, (2012)