

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

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

andreaquez@usal.es

Francisco José 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.

KEYWORDS

Domain engineering; Meta-modeling; Information Dashboards; High-level requirements.

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.

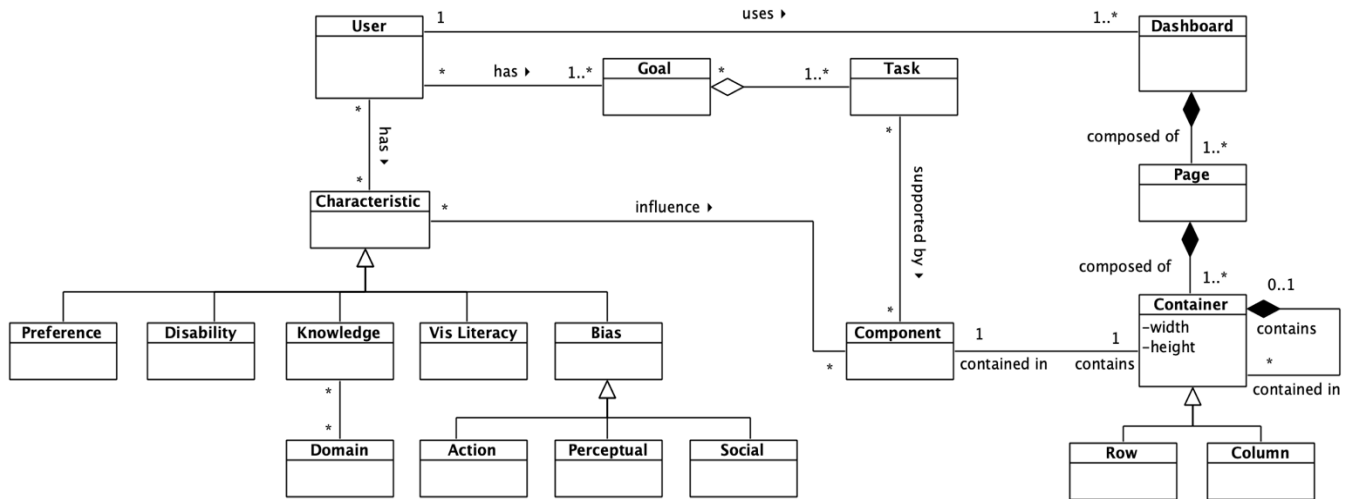


Figure 1: The user's section of the meta-model.

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.

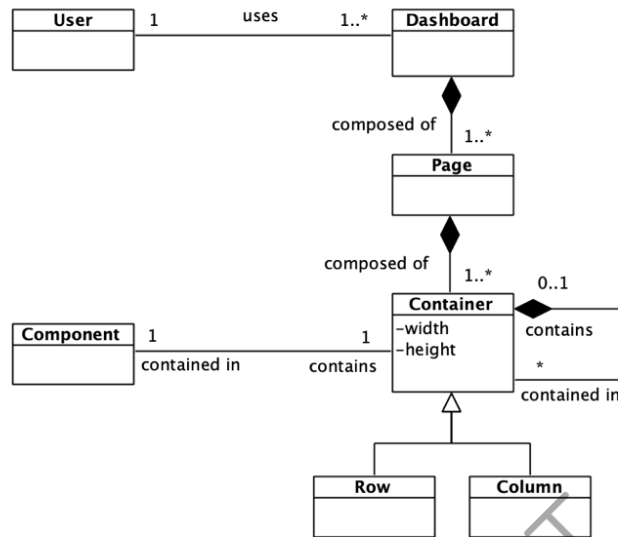


Figure 2: The layout's section of the meta-model.

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.

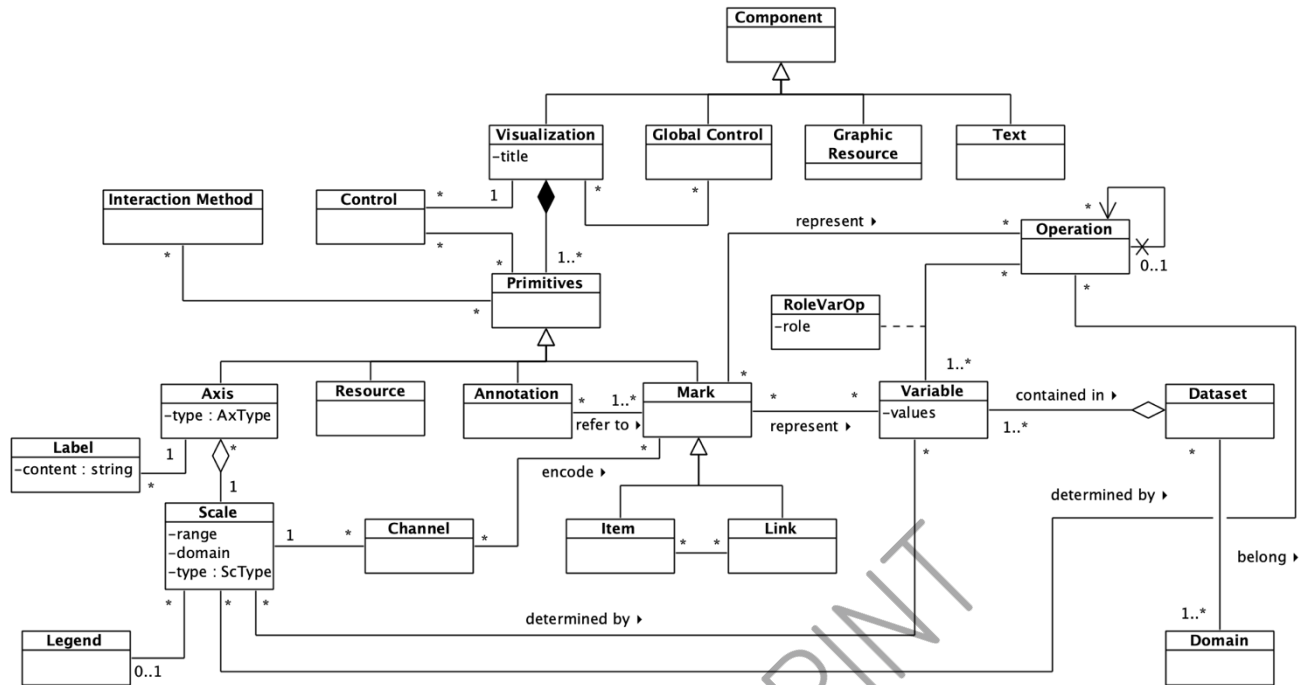


Figure 3: The components' section of the meta-model.

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.

GR/AL POST-PRINT

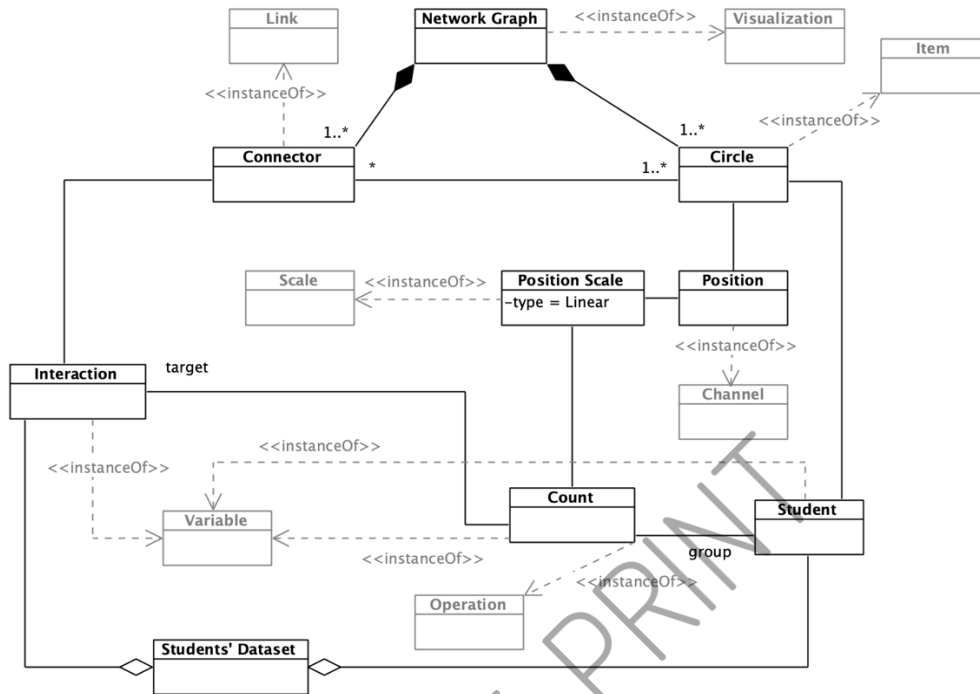


Figure 4: Possible instantiation of a network diagram to present students' interactions among them.

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.

