Representing Data Visualization Goals and Tasks Through Meta-Modeling to Tailor Information Dashboards

4 authors:

Andrea Vázquez-Ingelmo
Universidad de Salamanca
60 PUBLICATIONS **340** CITATIONS

Francisco José García-Peñalvo
Universidad de Salamanca
1,338 PUBLICATIONS **16,728** CITATIONS

Roberto Therón
Universidad de Salamanca
257 PUBLICATIONS **2,084** CITATIONS

Miguel Conde-González
Universidad de León
255 PUBLICATIONS **3,441** CITATIONS

Some of the authors of this publication are also working on these related projects:

Project   Recommender systems in E-Commerce View project

Project   Mobile Personal Learning Environments View project

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)

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.
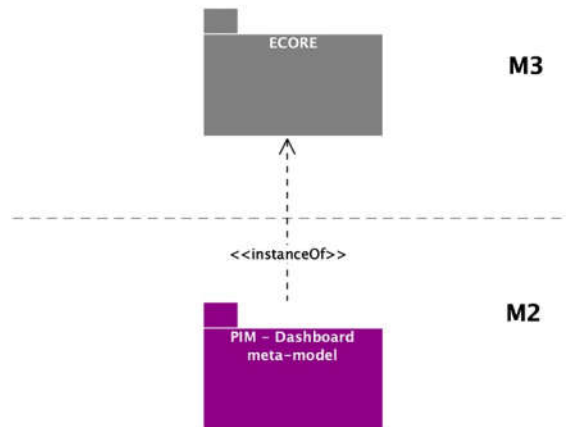
Figure 1. Location of the dashboard meta-model following the OMG architecture.

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.
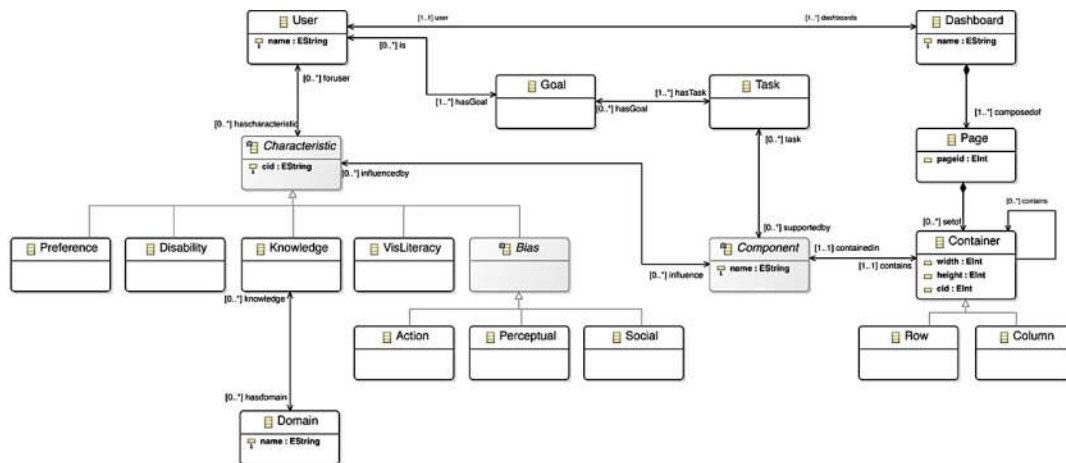

Figure 2. Overview of the dashboard meta-model, including the user, the layout, and the components. This image is available in high resolution at https://doi.org/10.5281/zenodo.3561320. Licensed under CC BY 4.0.

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.
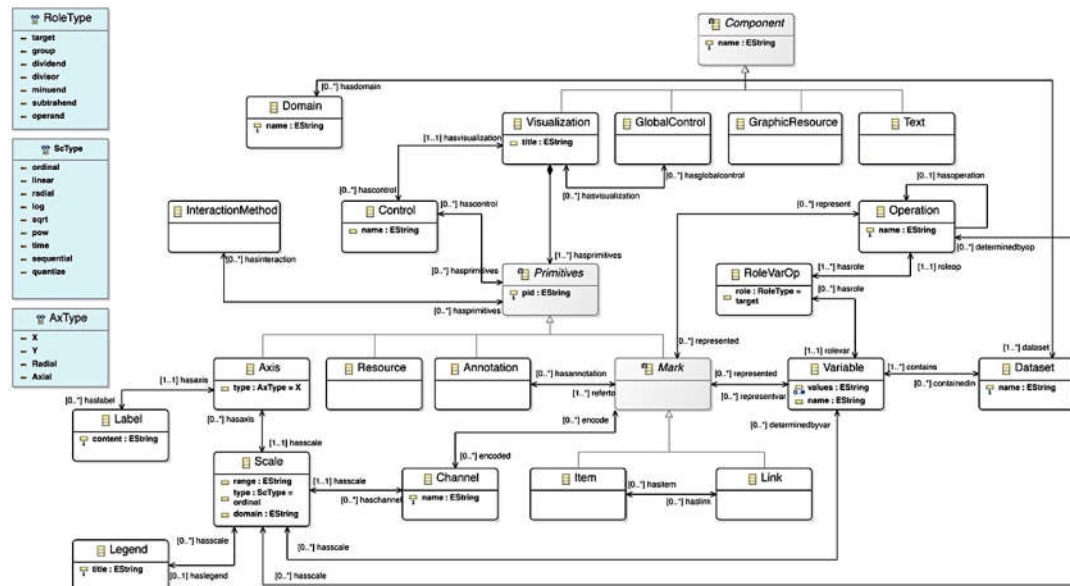
On the other hand, the user is modeled as an entity with two main elements that define her behavior: goals and characteristics. Goals refer to the purposes of the user regarding the displayed data, and they can be broken down into different lower-level analytic tasks that must be supported by the dashboard 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.
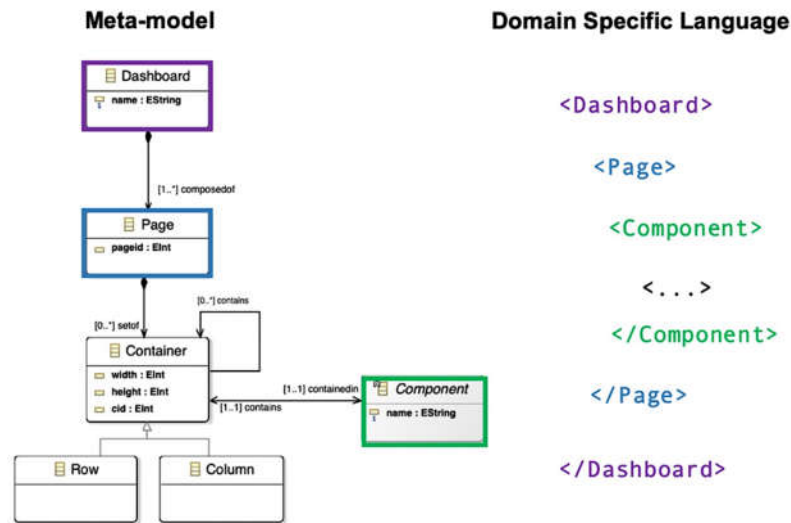
Figure 4. Correspondence between the meta-model entities and the XML entities that are part of the DSL. The subsequent primitives that are part of a component are materialized in the DSL through nested entities and properties, as will be presented in the results section.

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.
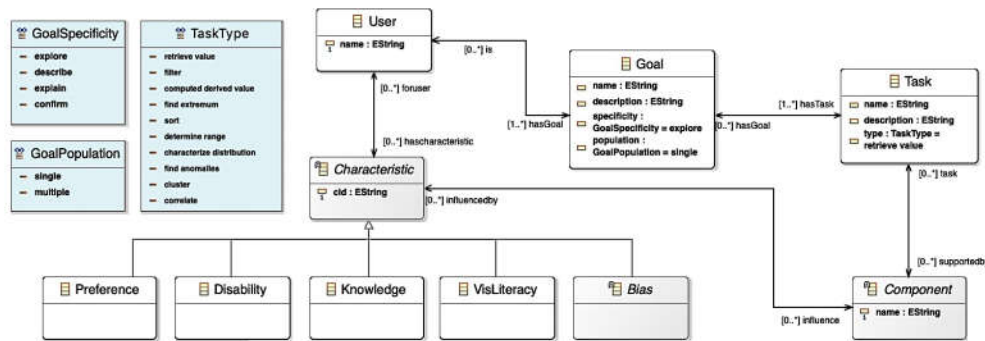
Figure 5. Extended user section of the meta-model. The rest of the meta-model has been omitted for legibility reasons. This image is available in high resolution at https://doi.org/10.5281/zenodo.3625703. Source: [57], licensed under CC BY 4.0.

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.

```xml
<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.
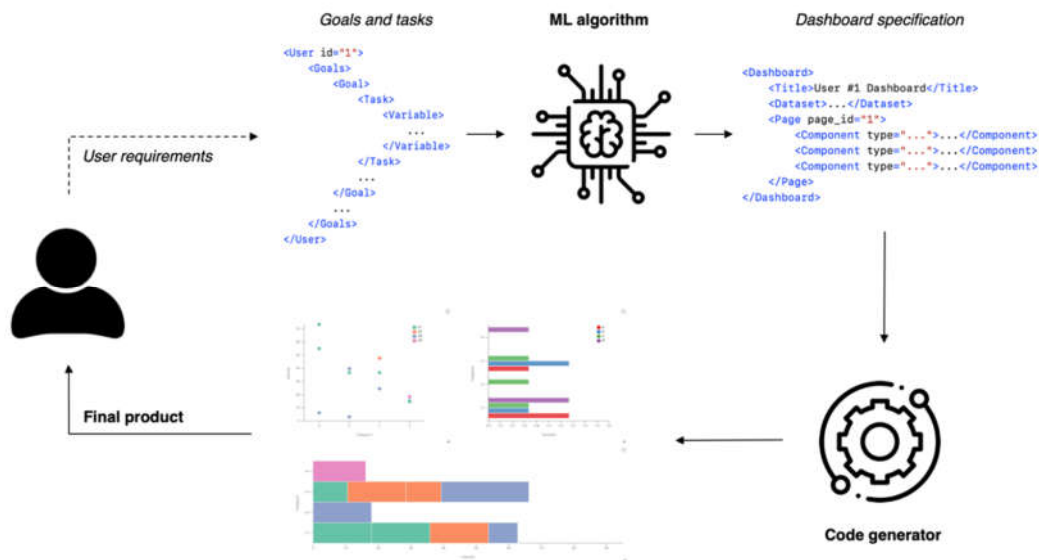

Figure 8. Generative pipeline proposal using the metamodel-based DSL. Icons made by Freepik (www.flaticon.com/authors/freepik).

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.
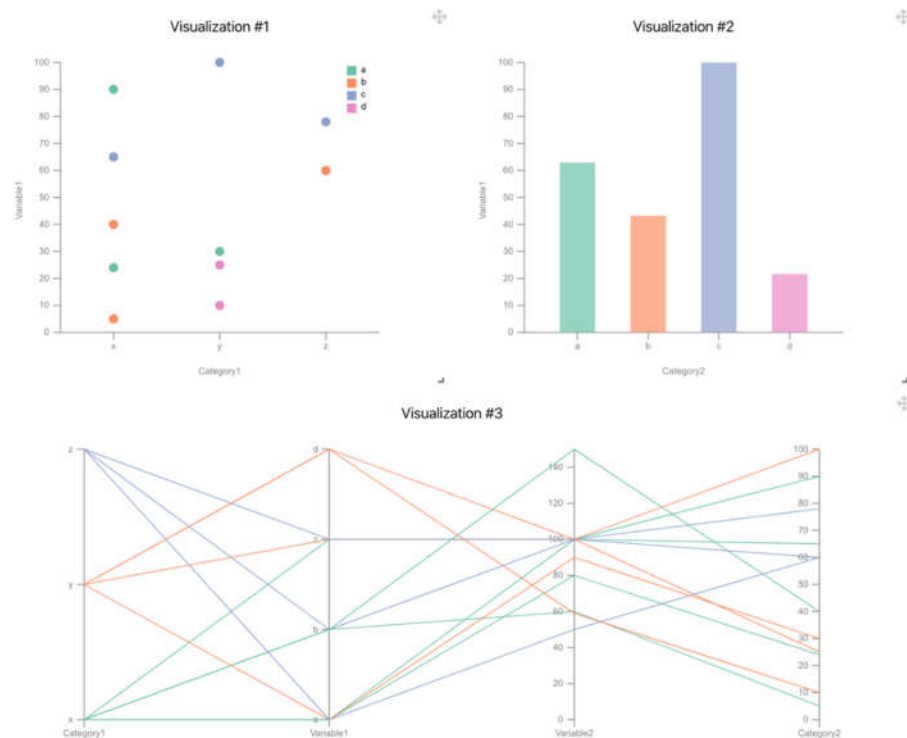

Figure 9. A sample dashboard generated by using the DSL.

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.
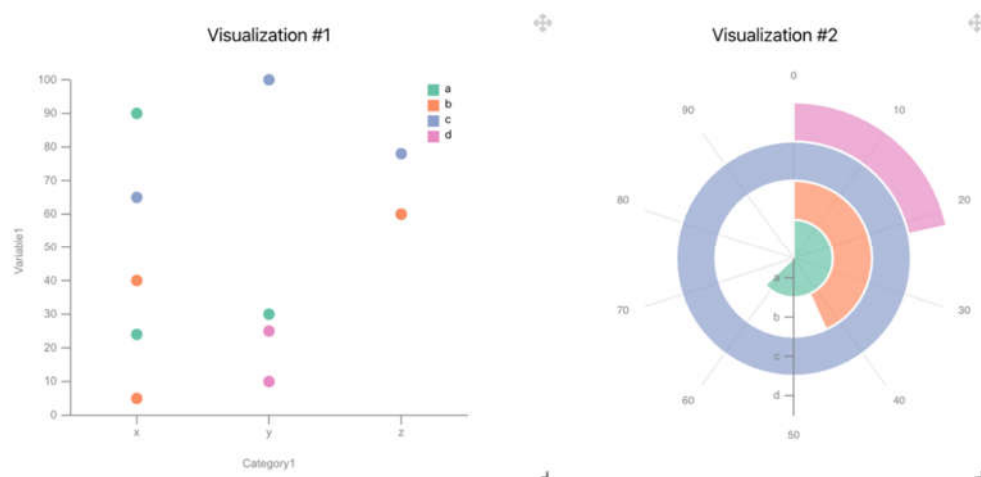

Figure 10. A sample dashboard modified by using the DSL. The second component shows the same information but using a polar coordinate system instead of a Cartesian coordinate system.

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.

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.