A Model-Driven Approach to Automate Data Visualization in Big Data Analytics

A Model-Driven Approach to Automate Data Visualization in Big Data Analytics

Matteo Golfarelli[1] and Stefano Rizzi[1]

Abstract

In big data analytics, advanced analytic techniques operate on big data sets aimed at complementing the role of traditional OLAP for decision making. To enable companies to take benefit of these techniques despite the lack of in-house technical skills, the H2020 TOREADOR Project adopts a model-driven architecture for streamlining analysis processes, from data preparation to their visualization. In this paper we propose a new approach named SkyViz focused on the visualization area, in particular on (i) how to specify the user's objectives and describe the dataset to be visualized, (ii) how to translate this specification into a platform-independent visualization type, and (iii) how to concretely implement this visualization type on the target execution platform. To support step (i) we define a visualization context based on seven prioritizable coordinates for assessing the user's objectives and conceptually describing the data to be visualized. To automate step (ii) we propose a skyline-based technique that translates a visualization context into a set of most-suitable visualization types. Finally, to automate step (iii) we propose a skyline-based technique that, with reference to a specific platform, finds the best bindings between the columns of the dataset and the graphical coordinates used by the visualization type chosen by the user. SkyViz can be transparently extended to include more visualization types on the one hand, more visualization coordinates on the other. The paper is completed by an evaluation of SkyViz based on a case study excerpted from the pilot applications of the TOREADOR Project.

Keywords

Big data visualization, Skyline, Model-driven architecture

Introduction

Big data analytics is the process of collecting and analyzing large volumes of data to extract hidden useful information using advanced analytic techniques. In the last few years it has become more and more popular in companies of all sizes to complement the role of traditional OLAP and data warehouses by taking advantage of the increasing amounts of valuable data generated by sensors, devices, social media, etc.[1]. Unfortunately, companies are often discouraged from running analytics because it requires technical skills that they lack, while the costs for outsourcing would be too high. Aimed at filling this gap, the H2020 TOREADOR (TrustwOrthy model-awaRE Analytics Data platfORm) Project adopts a *model-driven architecture*[2] to speed up and simplify the analysis process so as to make it widely available to companies via an analytics-as-a-service approach. Following the basic principles of model-driven architectures, TOREADOR relies on three models[3]:

1. **CIM (Computation-Independent Model)**: an abstract and platform-independent model that specifies the user objectives (*what* big data analytics should achieve) in terms of data collection, preparation, analysis, and visualization.
2. **PIM (Platform-Independent Model)**: a platform-neutral, vendor-independent model that specifies the algorithms for data preparation and for parallelizing and executing the analytics, as well as the way to present the results to users (*how* big data analytics should work).
3. **PSM (Platform-Specific Model)**: the computational components and other resources for the process on a

[1]DISI — CINI, University of Bologna, Italy

Corresponding author:
Stefano Rizzi, DISI — CINI, University of Bologna, Viale Risorgimento 2, 40136 Bologna, Italy.
Email: stefano.rizzi@unibo.it

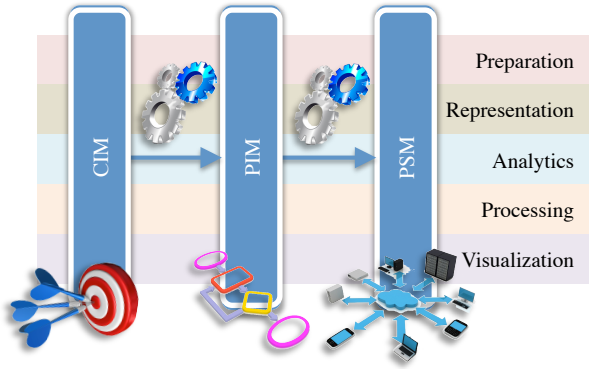

Figure 1. The framework of the TOREADOR Project

specific target execution platform (e.g., Hadoop-as-a-service).

In compliance with model-driven architectures, each model can be semi-automatically derived from the previous one.

Figure 1 shows how, in the TOREADOR Project, these three models are split into five conceptual areas: *preparation*, *representation*, *analytics*, *processing*, and *visualization*. The focus of our work is on visualization, which has a key role in big data analytics to enable users understand the problem, generate hypotheses, and define the solution, as well as to steer the analysis process when dealing with massive, incomplete, and incorrect data[4,5]. Specifically, we investigate (i) how to specify the user's objectives and describe the dataset to be visualized within the CIM (e.g., comparison-oriented visualization of n-dimensional numerical data with low cardinality), (ii) how to translate this specification into a platform-independent visualization type (e.g., bar chart) within the PIM, and (iii) how to concretely implement this visualization type into a PSM on the target execution platform (e.g., stacked-to-group bar chart in the D3 Javascript library[6]). In a previous paper[7] a preliminary solution to the first part of the problem, i.e., how to move from the CIM to the PIM, has been sketched. In this paper we propose the complete approach, named SkyViz; the main contributions are:

- We formalize the CIM in terms of a *visualization context* based on seven prioritizable coordinates for assessing the user's objectives and conceptually describing the data to be visualized (Section "An objective-based CIM for data visualization"). With reference to what was done in a previous paper[7], we enhance the approach to let users select multiple values for some coordinates and adopt a simpler formalization.
- We describe a skyline[8]-based technique for automatically translating a visualization context from the CIM onto the PIM in the form of a set of most-suitable visualization types (Section "Translating the CIM into the PIM").
- We describe a skyline-based technique for finding the best bindings between the columns of the dataset and the graphical coordinates used by the visualization type chosen by the user (among those determined at the previous step), so as to bridge the gap between the PIM and the PSM (Section "Translating the PIM into the PSM")

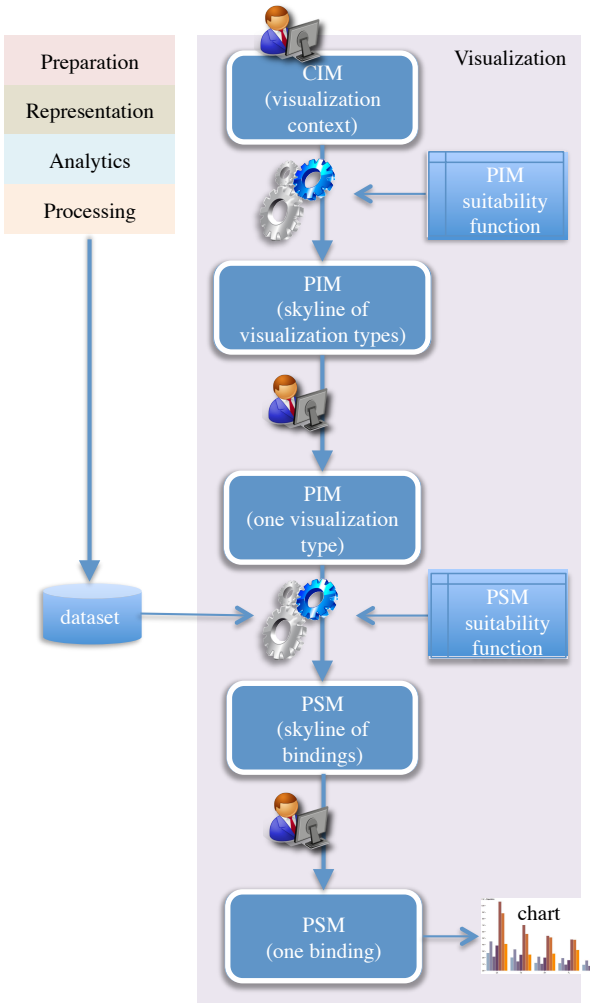

Figure 2. Approach overview

The overall approach is sketched in Figure 2. The user drives the process by first declaring the visualization context, then by choosing one visualization type among those proposed, and finally by choosing one binding among those proposed; this binding is then directly translated into a call to the graphical library adopted. Both CIM-to-PIM and PIM-to-PSM translations are based on a suitability function that rates visualization types and bindings; in particular, the set of possible bindings can be determined only after the dataset has been made available.

Thanks to the use of skyline computation to find the most suitable visualization(s), SkyViz can be transparently extended to include more visualization types on the one hand, more visualization coordinates on the other. Besides, since the visualization best practices are not hard-coded but modeled in the suitability function using a table of explicit scores, SkyViz can be tailored to the need of specific types of users by simply changing the scores (e.g., if users feel uncomfortable with reading dendrograms, the corresponding scores can be decreased). Finally, although in the paper we adopt D3 as a reference graphic library, SkyViz can be easily plugged into any other graphic library as long as the signatures for invoking its visualization services are known. Noticeably, all these possible extensions do not undermine the performance of the approach; indeed, as we will discuss in the paper, skyline computation still gives real-time performances when working with sets of objects that are orders of magnitude larger that those used in SkyViz.

The paper outline is completed by Section "Related Work", which discusses the related literature, by Section "Case Study and Evaluation", which evaluates SkyViz mainly through a real case study excerpted from the pilot applications of the TOREADOR Project, and by Section "Conclusions", which draws the conclusions.

Related Work

Principles and taxonomies to classify the different approaches for visualizing data and interacting with them have been proposed in the literature. First of all, Shneiderman proposed a classification taxonomy for data visualization based on the *task* (e.g., zoom and relate) and *data type* coordinates (e.g., multidimensional and tree)[9]. Similarly, visualization problems had been previously classified based on the *operation* to be performed (e.g., categorize and correlate) and on the *object* to be visualized (e.g., nominal and position)[10]. A few years later, a different classification of data visualization techniques[11] was suggested by considering, besides the data type (which mostly overlaps with the homonym coordinate of Shneiderman's work), the *visualization technique* (which corresponds to the tasks[9]) and the *interaction and distortion technique* (which distinguishes between standard displays, icon-based displays, dense displays, and stacked displays). Abela also listed four possible *goals* for visualization, namely relationship, comparison, distribution, and composition[12]. Tory et al.[13] introduced a high-level visualization taxonomy based on *design models*. A design space of visualization task was proposed taking six dimensions into account: goal ("why is a task pursued?"), means ("how is a task carried out?"), data characteristics ("what does a task seek?"), target ("where in the data does a task operate?"), order ("when is a task performed?"), and user type ("who is executing a task?")[14]. More recently, Börner surveyed the main classifications proposed in the literature to integrate them into a single framework[15] based on six coordinates, namely *insight need type*, *data scale type*, *visualization type*, *graphical symbol type*, *graphical variable type*, and *interaction type*. Data types were further classified with specific reference to the visualization of linked open data[16]; the paper also suggests suitable user goals some for some common chart types. Finally, the *user type* coordinate (which distinguishes users into *lay-users* and *techies*) was introduced for visualizing linked open data[17].

During the last 30 years, several approaches have been focused on the criteria for suggesting the most suitable type of chart for each data type, dimensionality, user goal, etc., and on methods and tools for automating visualization, using a variety of techniques that range from natural language processing (NLP) to genetic algorithms. A seminal approach in this direction is APT[18], which automatically designs effective graphical presentations of relational information; the underlying idea is that graphical presentations are sentences of graphical languages, and that the graphic design issues are codified as expressiveness and effectiveness criteria for graphical languages. A few years later, Vista[19] extended the design methodology of APT[18] from 2-dimensional to 3-dimensional graphics. Vista automatically generates an interactive visualization of a given data set by heuristically composing primitive visualization techniques (e.g., size and color).

Besides APT and VISTA, some other approaches can be classified as *data-driven*, since they do not explicitly consider the specific goal of the user for the current analysis, thus mainly relying on the dataset features to select a suitable visualization. Among these, Show Me[20] incorporates automatic presentation into the Tableau tool. It presents data within multiple displays, basically by applying visualization best practices based on the properties of the data fields. The DataVizard system[21] recommends the most appropriate visual presentation for the structured data either resulting from a SQL query or arranged within a data table taken for instance from the web. In the first case, the best visualization type is determined by first classifying the data columns into independent and dependent variables, then by considering their data type. In the second case, an NLP-based analysis of the table caption and of the table content is made. The VISO visualization ontology[22] formalizes the

vocabulary for the interdisciplinary visualization domain and properly annotates both data and visualization components. VISO is used to determine the applicable mappings between data variables and graphic coordinates; then, mappings are ranked taking into account the user and device context so as to eventually recommend a set of visualizations.

Several other approaches can be classified as *problem-driven*, since they directly take into account the various aspects that influence the effectiveness of a visualization, including the user's goal. In SAGE[23], a composite presentation for data is selected based on the data characteristics (e.g., their domain and their ordering), on the properties of the relational structure of data, and on the user's goal. BOZ[24] designs a visualization for data based on a user-provided logical description of the analysis task to be executed. The logical operators in this description are then turned into perceptual operators that can be graphically rendered, aimed at supporting efficient and accurate performance of the user's perceptual procedure. Vis-Wizz[25] recommends a visualization based on data characteristics and users goal as well as on an evaluation of the visual representation to be generated. A relevant approach was proposed by Zhang[26]; here, scale types (i.e., ratio, interval, ordinal, and nominal) are used to determine effective mappings between represented dimensions (columns of the dataset) and representing dimensions of the chart type (e.g., length, color, and shape). IMPROVISE[27] uses a data-analysis taxonomy plus some presentation context information to produce a user-centered visual design. The process is guided by a set of design principles that ensure the expressiveness and effectiveness of a design. Abela[12] proposes a decision tree to select the best visualization according to the user's goal and to the main features of data (number of variables, cyclicality, and size). This work inspired *Chart Chooser* (`labs.juiceanalytics.com/chartchooser`), a web site which returns the subset of Excel/PowerPoint charts compatible with one or more visualization goals selected by the user. ViA[28] is a visualization assistant that supports users in identifying perceptually salient visualizations for large, multidimensional datasets; this is done by applying knowledge of low-level human vision to evaluate visualizations given the dataset features (e.g., the spatial frequency of the attribute values) and analysis task (e.g., search and estimate). Marty[29] provides a description of the pros and cons of different chart types in the security domain, taking into account the data dimensionality, cardinality, and type. A flow-chart is proposed to help users in choosing the right visualization for different goals and data dimensionality, but not all combinations are covered. In Articulate[30] a conversational user interface enables users to verbally describe their analysis task; natural language sentences are then translated into explicit expressions and a visualization is heuristically selected using a decision tree inspired by Abela's work[12]. In the context of big data, a framework for choosing the best visualization is outlined[31]; the main types of charts are related to the user goals they fulfill and to the data dimensionality, cardinality, and type they support. VizAssist[32] is a user assistant that aims at improving the data-to-visualization mapping in data mining by means of an interactive genetic algorithm. To propose suitable visualizations for data it relies on a model of data (data type and importance of each variable in the dataset, and data cardinality), on a model of data mining objectives, and on a model of visualizations (which quantifies, for each visualization type, to what degree it is suitable for each data type, cardinality, and objective).

A separate mention is due for behavior-driven visualization recommendation[33]; here, the user's behavior is analyzed to detect meaningful interaction patterns, then these patterns are used to infer the user's intention for the current visual task and to suggest possible visualizations. In a more cognitive direction, Rogowitz et al. use perceptual rules to ensure that the structure of the data is faithfully represented in the visualization and to transform the structure of data so as to highlight specific features[34]. Their following work[35] introduces the PRAVDAColor tool, which is specifically focused at improving the user's selection of colormaps based on the structure of the data and on the visualization goal.

Table 1 shows a comparison of the above-mentioned approaches in terms of the coordinates they use for determining the best visualization. It emerges that, to the best of our knowledge, no previous approach took into account all the coordinates we consider. Besides, SkyViz is the first approach that uses skyline computation to find the best visualizations, which ensures full extensibility in terms of both the coordinate set and the set of visualization charts. Overall, the approaches that are more strictly related to ours are:

1. Vis-Wizz[25], which —similarly to SkyViz— relies on suitability functions to assess to what degree each visualization technique is suitable for each possible objective; however, differently from SkyViz, it gives no model of this function.

2. The approach by Zhang[26] can be seen as a way to bridge the gap between the PIM and the PSM;

Table 1. A comparison of approaches to select the best visualization

Approach	Technique	Goal	Interaction	User	Dimensionality	Cardinality	Data Type
APT [18]	composition rules, perceptual rules				✓	✓	✓
SAGE [23]	composition rules	✓				✓	✓
BOZ [24]	task description language, perceptual operators	✓	✓		✓		✓
Vista [19]	composition rules, perceptual rules				✓	✓	✓
PRAVDAColor [35]	perceptual rules	✓			✓		✓
Vis-Wizz [25]	suitability vectors	✓					✓
Zhang [26]	exact match	✓			✓		✓
IMPROVISE [27]	design rules	✓		✓	✓	✓	✓
Show Me [20]	defaulting						✓
Abela [12]	decision tree	✓			✓	✓	✓
ViA [28]	mixed-initiative strategy	✓					✓
BDVR [33]	pattern detection	✓	✓				✓
Marty [29]	flow-chart	✓			✓	✓	✓
Articulate [30]	NLP, decision tree	✓			✓	✓	✓
VISO [22]	discovery and ranking		✓	✓	✓	✓	✓
IBA [31]	flow-chart	✓			✓	✓	✓
VizAssist [32]	genetic algorithm	✓				✓	✓
DataVizard [21]	heuristic rules, NLP						✓
SkyViz	suitability function, skyline	✓	✓	✓	✓	✓	✓

however, it is limited to considering data types, user goals, and dimensionality.

3. Like SkyViz, VizAssist relies on extensible models created by domain expert; however, its models do not cover interactions, user type, and dimensionality.

4. The approaches by Abela [12] and Marty [29] give precise suggestions about the degree to which the most common visualization types are fit for different user goals and data features; hence we incorporated them into our suitability functions, though we had to extend them since they do not cover all our coordinates.

A different line of approaches is the one that proposes a framework to recommend a set of low-cost visualizations to users based on statistical properties of the dataset to be visualized such as its selectivity, data distribution, and number of distinct values [36]. This approach has a different goal from SkyViz since it actually aims at finding the most interesting variables of a large dataset to be visualized rather than the most appropriate visualization type for them. In the same direction, VizRank [37] is a method to automatically select the most useful data projections (i.e., those that best visually discriminate between classes) of 2-dimensional datasets. Similarly, AutoVis [38] is an automatic visualization system aimed at giving analysts a first view of any data source; as such, it is more concerned with determining the most interesting views of a dataset than on finding the most effective way to visualize them. The same is true for SeeDB [39], which is focused on efficiently finding the most interesting views of a multidimensional dataset (a view is considered to be interesting if it shows a deviation from a reference). More recently, an approach to recommend aggregate data visualization was proposed [40]; however, the emphasis is not on choosing a visualization type (only column charts are used) but rather on determining the most effective ways to aggregate data for generating interesting, usable, and accurate views. Similarly, a recommender system was used [41] to suggest visualizations; this was done through an ad-hoc query language and by introducing methods for choosing, ranking, and grouping recommended visualizations. Finally, Streit et al. [42] propose a comprehensive approach to the codesign of data, view, analytics, and tasks for heterogeneous data. The approach is based on a domain-independent model of the setup in which the analysis takes place, on a model of the domain that captures what can be done with a given setup in the context of a specific domain, and on a model of the analysis session that lists what has to be done to pursue a given analysis goal. The emphasis here is more on delivering an end-to-end guide to the user through the analysis process than on selecting the most appropriate visualization for data.

An objective-based CIM for data visualization

As already mentioned, the CIM is an abstract and platform-independent model that specifies the user objectives for visualizing the analysis results. In SkyViz, the CIM is defined in terms of a set of *visualization coordinates* whose values are specified by the user aimed at declaring her objectives and describing the dataset to be visualized. To select these coordinates in the context of the TOREADOR Project we adopt a requirement elicitation method that can be summarized as follows:

1. Based on the literature on the taxonomies of data visualization and interaction paradigms, we derived a set of candidate *coordinates* (e.g., data type) and, for each coordinate, a set of candidate *values* (e.g., ordinal). Each coordinate/value pair corresponds to a requirement.

2. From these requirements we derived a questionnaire which was submitted to users for requirement elicitation. More specifically, we involved 27 users of the pilot applications of the TOREADOR Project; of these, 13 were domain experts, 11 data engineers, and 3 data scientists.

3. Based on the results of requirement elicitation, we selected the final set of coordinates and values.

For requirement elicitation we adopted the Kano model[43], which enables designers to understand the needs and expectations of a stakeholder based on how they affect his/her satisfaction. The Kano model classifies requirements in the following classes:

- *Must-be*, which customers take for granted; if these requirements are not achieved, the stakeholder will be severely dissatisfied and not interested in the product at all.

- *One-dimensional*, those for which the level of functionality is proportional to the degree of satisfaction: the better a requirement is achieved, the higher the stakeholder will be satisfied, and vice versa.

- *Attractive*, which are usually unexpected by the stakeholders but have the greatest influence on how satisfied they will be. As the level of functionality achieved by these requirement increases, the stakeholder's satisfaction increases more than proportionally.

- *Indifferent*, which are rated as neither good nor bad.

- *Reverse*, which cause dissatisfaction when present and satisfaction when absent.

The Kano model is typically constructed using a survey methodology, where requirements are first classified at the individual stakeholder level through a questionnaire and then aggregated. The Kano questionnaire contains a list of question pairs for each requirement; the question pair includes a *functional question*, asking how the user would feel if a certain requirement were met, and a *dysfunctional question*, asking how the user would feel if that requirement were not met. An example of requirement and of the two related questions posed to users is shown in Table 2. To answer each question, the user had the options listed below:

- *Like*: "This would be helpful to me"

- *Expect*: "This is a basic requirement to me"

- *Neutral*: "This would not affect me"

- *Tolerate*: "This would be a minor inconvenience"

- *Dislike*: "This would be a major problem for me"

The answers to all questions were collected and analyzed using the DuMouchel methodology[44]. This methodology assumes the use, together with the Kano questionnaire, of a *self-stated importance questionnaire* which makes the respondents rank each requirement on a scale of importance aimed at determining the relative importance of each individual requirement. Then it assigns three scores to each requirement: the *functional score* maps each answer given to a functional question onto the range from 4 (Like) to -2 (Dislike); the *dysfunctional score* maps each answer given to a dysfunctional question onto the range from -2 (Like) to 4 (Dislike); the *importance score* maps each answer in the self-stated importance questionnaire onto the range from 1 to 5. The three scores obtained for each requirement, averaged over the set of all respondents, enable the categorization of that requirement as either must-be, one-dimensional, attractive, indifferent, or reverse according to its positioning within a two-dimensional grid[44]. As a consequence of the process described above, all the candidate coordinates were deemed to be either must-be (e.g., user), one-dimensional (e.g., goal), or attractive (e.g., cardinality). Conversely, some coordinate values (e.g, the history and projection interactions) were categorized as reverse, since they were considered to be too specific and possibly misleading, so we had to exclude them.

In the following we list the seven coordinates we selected, see Table 3 for the complete list of values each coordinate can take:

Table 2. A requirement and the related functional and dysfunctional questions

Code	REQ02
Requirement	The TOREADOR platform will enable users to declare the number of variables they wish to visualize (1, 2, …, N, tree, graph)
Rationale	To enable the platform to suggest a visualization that can support the chosen number of variables
Scenario	You are about to analyze the effectiveness of a promotional campaign for ice-creams. The impact of this campaign on sales could be investigated from different perspectives aimed at gaining insights using a different number of variables. For instance you could be interested in • reading the total sales-to-date of ice-creams since the beginning of the campaign (1 dimension); • analyzing the trend of ice-creams sales during the campaign (2 dimensions); • visualizing, for each nation, the daily trends of ice-creams sales and costs during the campaign (4 dimensions)
Functional question	The CIM allows to declare the number of visualized variables (1D, 2D, 3D, nD, tree, graph) aimed at suggesting the most suitable visualization
Dysfunctional question	The CIM does not allow to declare the number of visualized variables

Table 3. Visualization coordinates

Value	*Description*	*Example*
Goal		
Composition	highlighting the way in which distinct parts of data are composed to form a total	stacked column chart
Order	analyzing objects by emphasizing their ordering	alphabetical list of names
Relationship	analyzing the correlation between two or more objects or attribute values	point graph
Comparison	examining two or more objects or values to establish their similarities and dissimilarities	column chart
Cluster	analyzing data in such a way as to emphasize their grouping into categories	dendrogram
Distribution	analyzing how objects are dispersed in space	histogram
Trend	examining a general tendency of data variables	line graph
Geospatial	analyzing data values using a geographical map as a graphical context	choropleth map
Interaction		
Overview	gain an overview of the entire data collection	dendrogram
Zoom	focus on items of interest	network map
Filter	quickly focus on interesting items by eliminating unwanted items	area chart
Details-on-demand	select an item and get its details	choropleth map
User		
Lay	computer-literates who may have troubles in understanding complex visualizations	line graph
Tech	skilled users with a deeper understanding of analytics	tree map
Dimensionality		
1-dimensional	a single numerical value or a string	gauge
2-dimensional	one dependent variable as a function of one independent variable	single line graph
n-dimensional	each data object is a point in an n-dimensional space	bubble graph
Tree	a collection of items, each having a link to one parent item	dendrogram
Graph	a collection of items, each linked to an arbitrary number of other items	network map
Cardinality		
Low	from a few items to a few dozens items	pie chart
High	some dozens items or more	heat map
Independent/Dependent Type		
Nominal	qualitative, each data variable is assigned to one category (e.g., “male” and “female”)	pie chart
Ordinal	qualitative, categories can be sorted (e.g., “small”, “medium”, “large”)	column chart
Interval	quantitative, it supports the determination of equality of intervals or differences (e.g., a temperature)	line graph
Ratio	quantitative, with a unique and non-arbitrary zero point (e.g., an income)	point graph

(1) *Goal*, which enables users to declare their main analysis goal(s). This classification follows the one into *basic task types*[15]; examples of goals are that of analyzing data based on their order (in which case, a sorted list of data could be a good choice) and that of comparing pieces of data to assess how similar they are (e.g., using a column chart).

(2) *Interaction*, which enables users to declare the type of interactions to be supported by the visualization. This classification derives from a previous one[15]; specifically, based on requirement elicitation, we selected a subset of most common and intuitive interaction types[11]. For instance, the user may wish to gain an overview of the data using a dendrogram, or may need to get further

details about a selected piece of data by clicking on a mark in a marked line graph.

(3) *User*, which enables users to declare their skill[17]. We distinguish lay users, for which simple visualization types such as line graphs are more suitable, and tech users, who can also understand more complex visualization types such as tree maps.

(4) *Dimensionality*, which enables users to declare the number of variables they wish to visualize. Here, as done by Abela[12], we count all variables without distinguishing between independent and dependent variables. Clearly, while a few visualization types are suitable for 1-dimensional datasets (e.g., gauges and alerts), most of them require n-dimensional datasets (e.g., histograms and bubble graphs). Also trees and graphs are considered here, which can be visualized using dedicated approaches like dendrograms and networks map, respectively.

(5) *Cardinality*, which enables users to qualitatively declare the cardinality of the data to be visualized[12]. Since the user at this stage will probably have only a rough idea of the cardinality, here we just distinguish between low cardinality, up to a few dozens rows (which are better visualized using a pie chart, for instance) and high cardinality (which should be shown using dense visualization types such as line graphs and heat maps).

(6) *Independent Type*, which enables users to declare the type of the independent variable(s) to be visualized. The classification we adopt here[45] includes four data types: nominal (qualitative and unordered, can be visualized using for instance the colors in a pie chart), ordinal (qualitative and ordered, shown for instance through the row labels in a pivot table), interval (quantitative with no zero point, can be visualized using for instance the X-axis of a column graph), and ratio (quantitative with zero point, shown for instance through the X-axis of a point graph).

(7) *Dependent Type*, which enables users to declare the type of the dependent variable(s) to be analyzed. The classification we adopt here is the same of the independent type. Using two separate coordinates for independent and dependent variables enables a finer specification of the CIM[29] and a more accurate translation into the PIM and the PSM; for instance, while the color in a pie chart is suitable to show a nominal variable, the width of each sector should represent a ratio variable.

Note that, while for coordinates User, Dimensionality, and Cardinality, one single value can be specified by the user because the possible values have disjunctive semantics, for coordinates Goal, Interaction, Independent Type, and Dependent Type the semantics of values is conjunctive, so the user can specify multiple values (e.g., the user might be interested in interacting with the visualization using both overview and details-on-demand).

We now formalize the CIM in terms of a *visualization context* based on the seven coordinates listed above for assessing the user's objectives and conceptually describing the data to be visualized. The context has variable size to accommodate both the case in which the user does not specify a value for some coordinate(s) and that in which she specifies multiples values for some coordinate(s). Besides, the user can prioritize coordinate values to express her higher or lower confidence and interest in each value.

Definition 1. Visualization Context. *Let*

$$
\begin{aligned}
O_{\text{goa}} &= \{\text{Composition}, \text{Order}, \text{Relationship}, \text{Comparison}, \\
&\qquad \text{Cluster}, \text{Distribution}, \text{Trend}, \text{Geospatial}\} \\
O_{\text{int}} &= \{\text{Overview}, \text{Zoom}, \text{Filter}, \text{Details-on-demand}\} \\
O_{\text{use}} &= \{\text{Lay}, \text{Tech}\} \\
O_{\text{dim}} &= \{\text{1-dimensional}, \text{2-dimensional}, \text{n-dimensional}, \\
&\qquad \text{Tree}, \text{Graph}\} \\
O_{\text{car}} &= \{\text{Low}, \text{High}\} \\
O_{\text{ind}} &= \{\text{Nominal-i}, \text{Ordinal-i}, \text{Interval-i}, \text{Ratio-i}\} \\
O_{\text{dep}} &= \{\text{Nominal-d}, \text{Ordinal-d}, \text{Interval-d}, \text{Ratio-d}\}
\end{aligned}
$$

be the sets of values for coordinates goals, interactions, users, dimensionalities, cardinalities, independent types, and dependent types, respectively; let $O = O_{\text{goa}} \cup O_{\text{int}} \cup O_{\text{use}} \cup O_{\text{dim}} \cup O_{\text{car}} \cup O_{\text{ind}} \cup O_{\text{dep}}$. *A* visualization context *is defined as* $C, \overset{C}{\succ}$, *where* $C \subset O$ *is a subset that includes at most one element from* O_{use}, O_{dim}, *and* O_{car}, *and* $\overset{C}{\succ}$ *is a weak order on* C *that expresses the priorities between the different coordinate values.*

Example 1. *An example of visualization context is* $C, \overset{C}{\succ}$ *where*

$$
\begin{aligned}
C = \{&\text{Comparison}, \text{Tech}, \text{n-dimensional}, \text{High}, \text{Interval-i}, \\
&\text{Ratio-d}, \text{Nominal-d}\}
\end{aligned}
$$

and

$$(\text{Tech} \overset{C}{\sim} \text{Interval-i}) \overset{C}{\succ} \text{Comparison} \overset{C}{\succ}$$
$$\overset{C}{\succ} (\text{n-dimensional} \overset{C}{\sim} \text{High} \overset{C}{\sim} \text{Ratio-d} \overset{C}{\sim} \text{Nominal-d})$$

where the user expresses three levels of priority: high (for the user and independent type coordinates), medium (for the goal coordinate), and low (for the dimensionality, cardinality, and dependent type coordinates). No value is specified for the interaction coordinate. □

Translating the CIM into the PIM

In this section we discuss the CIM-to-PIM transformation, specifically, how the visualization context stated by the user in the CIM can be transformed into a set of suitable visualization types in the PIM. The first step in this direction requires to assess to which extent each visualization type is suitable for each value of each visualization coordinate introduced in Section "An objective-based CIM for data visualization".

Definition 2. PIM Suitability Function. *A* PIM suitability function *is a total function* $\sigma : O \times V \rightarrow s$ *where* O *is the set of all coordinate values,* V *is the set of all visualization types, and* $s \in \{\text{unfit}, \text{discouraged}, \text{acceptable}, \text{fit}\}$ *is a score.*

The semantics of the scores is as follows:

- **unfit** means that the visualization type *should not be used* for the coordinate value. For instance, a pie chart cannot be used to represent 1-dimensional data.
- **discouraged** means that the visualization type *can be used* in principle for the coordinate value, but it may distort the very nature of that specific goal, interaction, user, dimensionality, cardinality, or type. For instance, a pie chart should not be used to fulfill the distribution goal because it does not emphasize how objects are dispersed in space.
- **acceptable** means that the visualization type *is compatible* with the coordinate value, though it may fail to emphasize some of the features of that specific goal, interaction, user, dimensionality, cardinality, or type. For instance, a pie chart can successfully be used to visualize an ordinal independent variable such a S/M/L/XL tag, but it will give no specific emphasis to the ordering of values.
- **fit** means that the visualization type is fully compatible with the coordinate value, and has been declared in the literature to be a best visualization practice for that specific goal, interaction, user, dimensionality, cardinality, or type. For instance, the pie chart is perfectly fit to visualize a nominal independent variable (such as Continent) and a ratio dependent variable (such as SalesRevenues).

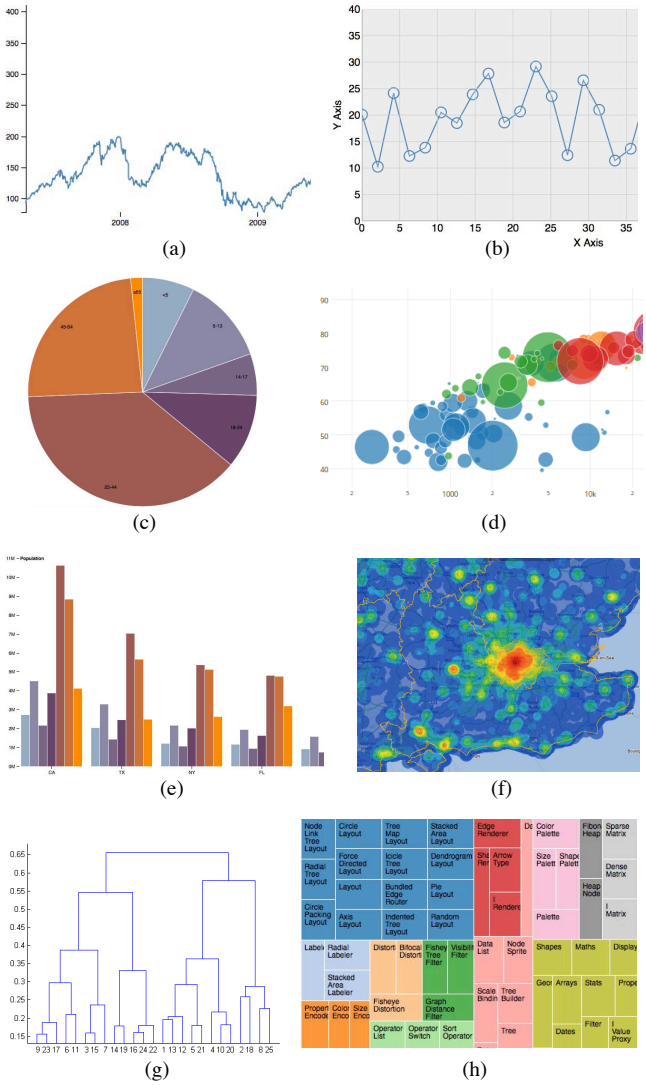

Figure 3. A single line graph (a), a marked line graph (b), a pie chart (c), a bubble graph (d), a grouped column graph (e), a heat map (f), a dendrogram (g), and a tree map (h)

SkyViz can be applied to each possible visualization type v as long as a suitability evaluation is done by a visualization expert for v based on our seven coordinates. In this paper we focus on the eight popular visualization types shown in Figure 3, namely, single line graph, marked line graph, pie chart, bubble graph, grouped column graph, heat map, dendrogram, and tree map. For each of them we assigned a score to each every coordinate-value pair, so as to define a PIM suitability function as shown in Table 4. The scores were mainly derived from the best practices found in the literature[12,15,29]; where we could not find any specific

prescription in the literature, we fell back on common sense to complete the function assignments.

The PIM suitability function can now be used to find one or more "most suitable" visualization types for a given visualization context $C, \overset{C}{\succ}$. To this end we start by observing that, with reference to $C = \{c_1, \ldots, c_p\}$, visualization type v is evaluated through a set $\{\sigma(c_1, v), \ldots, \sigma(c_p, v)\}$ of scores, where each element expresses the suitability of v for C along one coordinate value. We also note that the scores introduced in Definition 2 are obviously related by a strict total order that expresses a preference:

$$\text{fit} > \text{acceptable} > \text{discouraged} > \text{unfit}$$

This enables a comparison between any two possible visualization types $v, v' \in V$ along each single coordinate value: for the i-th value, v *is preferred to (i.e., is strictly better than)* v' *if* $\sigma(c_i, v) > \sigma(c_i, v')$.

The next step is to understand how to combine the p resulting one-dimensional preferences for each visualization coordinate into a single one for the whole visualization context. A very reasonable way to cope with this problem is to look for visualization types that are Pareto-optimal. A visualization types is *Pareto-optimal* when no other visualization types *dominates* it, being better along one coordinate and not worse along all the other coordinates. In the database community, when multiple preferences are defined over a set of tuples, the set of tuples (in our context, visualization types) satisfying Pareto-optimality is called a *skyline*[46].

The definition of dominance is given below in flat (non-prioritized) form first; then, we will generalize it to cope with the presence of priorities.

Definition 3. Flat Dominance. *Given visualization context* $C, \overset{C}{\succ}$ *and two visualization types* v *and* v'*, we say that* v *is* equivalent *to* v' *on* C*, denoted* $v \sim_C v'$*, iff* $\sigma(c_j, v) = \sigma(c_j, v')$ *for all* $c_j \in C$*. We say that* v flat-dominates v' *on* C*, denoted* $v \rhd_C v'$*, iff*

(a) $\exists c_i \in C : \sigma(c_i, v) > \sigma(c_i, v')$ *and*

(b) for all other $c_j \in C$ *it is* $\sigma(c_j, v) = \sigma(c_j, v')$

Example 2. *Consider again the visualization context in Example 1, which we match with the eight visualization types in Table 4. The eight corresponding suitability sets are singled out in Table 5. The resulting flat dominance relationships are shown in Figure 4.a. For instance, heat map flat-dominates single line graph (*heat map $\rhd_C$ single line graph*) because it is equivalent on all coordinates except goal and dimensionality, on which it is better. On the other hand, there is no flat-dominance or equivalence between bubble graph and heat map because the first is better on the goal coordinate, while the second is better on the cardinality coordinate. So overall, if coordinate priorities are not considered, bubble graph, heat map, and tree map are Pareto-optimal and would belong to the skyline, while the others would not.* □

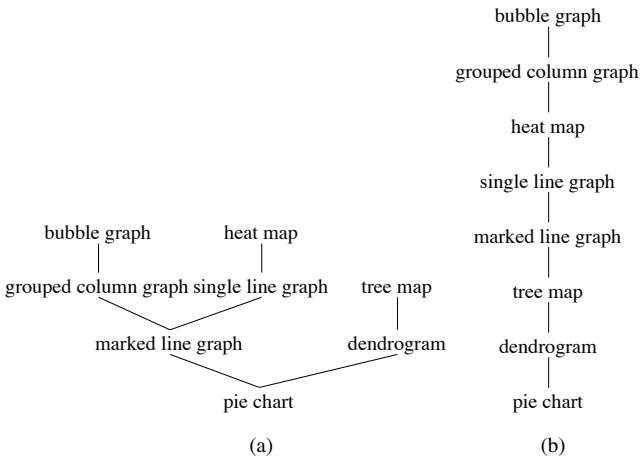

Figure 4. Flat dominance (a) and dominance (b) relationships for Example 2

The definition of dominance is now generalized to cope with the priorities $\overset{C}{\succ}$ declared by the user. To this end we resort to the concept of *prioritized skyline*[46] and redefine dominance as follows. Intuitively, if v is better than v' with reference to the coordinate values that take highest priority for the user, then it is unconditionally better than v'; otherwise, if v is equivalent to v' with reference to those coordinate values, we have to check if it is better with reference to the coordinate values taking second priority, and so on.

Definition 4. Dominance. *Given visualization context* $C, \overset{C}{\succ}$ *and two visualization types* v *and* v'*, and given the set of coordinate values* $\overline{C} \subseteq C$*, we say that* v dominates v' *on* $\overline{C}$ *(denoted* $v \blacktriangleright_{\overline{C}} v'$*) iff either (a)* $v \rhd_{max(\overline{C})} v'$ *or (b)* $(v \sim_{max(\overline{C})} v') \wedge (v \blacktriangleright_{\overline{C} \setminus max(\overline{C})} v')$*, where* $max(\overline{C})$ *denotes the top coordinate values in the* $\overset{C}{\succ}$ *order restricted to* $\overline{C}$*.*

Definition 5. PIM Skyline. *The* PIM skyline *for* $C, \overset{C}{\succ}$ *is the set of visualization types in* V *that are not dominated on* C *by any other visualization type.*

It is easy to prove that $v \rhd_C v'$ implies $v \blacktriangleright_C v'$ for any $\overset{C}{\succ}$; as a consequence, the skyline for flat dominance always includes the skyline for dominance, i.e, *prioritizing coordinate values leads to reducing the skyline*.

Table 4. PIM suitability scores for eight visualization types

		single line graph	marked line graph	pie chart	bubble graph
Goal:	Composition	unfit	unfit	fit	discouraged
	Order	discouraged	unfit	unfit	unfit
	Relationship	unfit	unfit	unfit	fit
	Comparison	unfit	unfit	unfit	fit
	Cluster	unfit	unfit	unfit	acceptable
	Distribution	acceptable	acceptable	unfit	fit
	Trend	fit	fit	unfit	acceptable
	Geospatial	unfit	unfit	unfit	discouraged
Interaction:	Overview	fit	fit	fit	fit
	Zoom	acceptable	acceptable	unfit	acceptable
	Filter	discouraged	discouraged	acceptable	discouraged
	Details-on-dem	acceptable	fit	acceptable	acceptable
User:	Lay	fit	fit	fit	acceptable
	Tech	fit	fit	acceptable	fit
Dimens.:	1-dimensional	unfit	unfit	unfit	unfit
	2-dimensional	fit	fit	fit	unfit
	n-dimensional	unfit	unfit	unfit	fit
	Tree	unfit	unfit	unfit	unfit
	Graph	unfit	unfit	unfit	unfit
Cardinality:	Low	acceptable	fit	fit	acceptable
	High	fit	discouraged	discouraged	discouraged
Ind. Type:	Nominal-i	unfit	unfit	fit	unfit
	Ordinal-i	discouraged	discouraged	acceptable	discouraged
	Interval-i	fit	fit	discouraged	fit
	Ratio-i	fit	fit	discouraged	fit
Dep. Type:	Nominal-d	unfit	unfit	unfit	fit
	Ordinal-d	unfit	unfit	unfit	fit
	Interval-d	fit	fit	discouraged	acceptable
	Ratio-d	fit	fit	fit	fit

		grouped column graph	heat map	dendrogram	tree map
Goal:	Composition	acceptable	unfit	acceptable	acceptable
	Order	acceptable	unfit	discouraged	unfit
	Relationship	discouraged	unfit	acceptable	fit
	Comparison	fit	acceptable	discouraged	acceptable
	Cluster	acceptable	acceptable	fit	fit
	Distribution	acceptable	fit	discouraged	discouraged
	Trend	fit	unfit	unfit	unfit
	Geospatial	unfit	fit	unfit	unfit
Interaction:	Overview	fit	fit	fit	fit
	Zoom	unfit	fit	fit	acceptable
	Filter	acceptable	acceptable	acceptable	acceptable
	Details-on-dem	acceptable	acceptable	acceptable	acceptable
User:	Lay	fit	acceptable	acceptable	discouraged
	Tech	fit	fit	fit	fit
Dimens.:	1-dimensional	unfit	unfit	unfit	unfit
	2-dimensional	unfit	unfit	unfit	unfit
	n-dimensional	fit	fit	unfit	acceptable
	Tree	unfit	unfit	fit	fit
	Graph	unfit	unfit	unfit	unfit
Cardinality:	Low	fit	acceptable	fit	fit
	High	discouraged	fit	acceptable	acceptable
Ind. Type:	Nominal-i	fit	acceptable	fit	fit
	Ordinal-i	fit	acceptable	discouraged	discouraged
	Interval-i	acceptable	fit	discouraged	discouraged
	Ratio-i	acceptable	fit	unfit	unfit
Dep. Type:	Nominal-d	unfit	unfit	unfit	fit
	Ordinal-d	unfit	discouraged	unfit	discouraged
	Interval-d	discouraged	fit	acceptable	discouraged
	Ratio-d	fit	fit	fit	fit

Example 3. *Considering again the visualization context in Examples 1 and 2, and taking now into account the coordinate priorities, the dominance relationships are shown in Figure 4.b. Indeed, all visualization types are equivalent on the two top-priority coordinates of* $\overset{C}{\succ}$ $(max(C) = \{\text{Tech}, \text{Interval-i}\})$ *except pie chart, dendrogram, and tree map, which are flat-dominated by other visualization types and can be immediately excluded from the PIM skyline. For*

Table 5. Suitability tuples for eight visualization types with reference to the visualization context in Example 1

		single line graph	marked line graph	pie chart	bubble graph
Goal:	Comparison	unfit	unfit	unfit	fit
User:	Tech	fit	fit	acceptable	fit
Dimens.:	n-dimensional	unfit	unfit	unfit	fit
Cardinality:	High	fit	discouraged	discouraged	discouraged
Ind. Type:	Interval-i	fit	fit	discouraged	fit
Dep. Type:	Ratio-d	fit	fit	fit	fit
Dep. Type:	Nominal-d	unfit	unfit	unfit	fit

		grouped column graph	heat map	dendrogram	tree map
Goal:	Comparison	fit	acceptable	discouraged	acceptable
User:	Tech	fit	fit	fit	fit
Dimens.:	n-dimensional	fit	fit	unfit	acceptable
Cardinality:	High	discouraged	fit	acceptable	acceptable
Ind. Type:	Interval-i	acceptable	fit	discouraged	discouraged
Dep. Type:	Ratio-d	acceptable	fit	fit	fit
Dep. Type:	Nominal-d	unfit	unfit	unfit	fit

the remaining six visualization types we have to check the second-priority coordinate ($max(C \setminus \{\text{Tech}, \text{Interval-i}\}) = \{\text{Comparison}\}$*), on which bubble graph and grouped column graph are better than heat map; thus, heat map is dominated and excluded from the PIM skyline. Single and marked line graph are in turn dominated by heat map, so they too can be excluded. Finally, we find that bubble graph and grouped column graph are equivalent on the remaining coordinates, except for the dependent type coordinate on which bubble graph is better. So, taking into account priorities, the PIM skyline only includes bubble graph.* □

We close this section by recalling that skyline approaches are normally applied to rank the tuples of a database based on the users preferences. As such, they give real-time performance over thousands of objects. The performance and scalability of an algorithm for computing prioritized skylines have been measured[46], and it turned out that the time for computing the result is always below 1 second, with a dataset including 50000 tuples and 20 attributes —well beyond the maximum number of visualization types we are expected to manage and the seven coordinates we currently use in SkyViz.

Translating the PIM into the PSM

In the model-driven approach, defining the PSM requires first of all to choose a target execution platform. In our context, this means choosing a specific platform that implements visualization services. In the following we pick the well-known D3 Javascript library[6] as a reference platform. Then, translating a PIM into a PSM means, given a dataset D to be visualized and a visualization type v picked by the user among those in the PIM skyline, deciding how each variable in D will be visualized, i.e., establishing a binding between each variable and a graphical coordinate of v. For instance, if the user has picked pie chart as her preferred visualization type out of the PIM skyline to visualize a dataset including variables Continent and SalesRevenue, two bindings are possible: using colors to represent continents and arc widths to represent revenues, or the opposite.

To discuss how this translation can be automated, we need some preliminary definitions.

Definition 6. Dataset and Variable. *A dataset D is a list of tuples, where each tuple consists of n variables. Each variable a_i has a type, $type(a_i) \in T$, with* $T = \{\text{Nominal}, \text{Ordinal}, \text{Interval}, \text{Ratio}, \text{Tree}, \text{Graph}\}$.

Given a dataset, determining the types of its variables can be done automatically to some extent, since nominal and ordinal variables are normally represented by strings, interval variables are represented by either numbers or dates/timestamps, and ratio variables are represented by numbers. To distinguish nominal from ordinal variables we must resort to the user's judgement (a qualitative variable is ordinal if there is a meaningful ordering of values, nominal otherwise). Similarly to distinguish interval from ratio variables (a quantitative variable is ratio if it has a meaningful zero, interval otherwise).

Note that we added Tree and Graph to the set of simple types introduced in Table 3. This is to effectively deal with visualization types which operate on trees and graphs, such as dendrograms and chords, respectively. Indeed, in this case, a graphical coordinate of the visualization type has to be fed with a complex variable that uses some conventional notation to code a topology (for instance, in the D3 library a tree-like topology can be expressed using the dot notation to represent each path in the tree, while a graph topology can be expressed as a couple of labels to denote each arc). Though the complex

types we consider are those most commonly used in big data analytics, SkyViz may be gracefully extended to cope with more sophisticated types (such as hypergraphs) by adding them to T.

Example 4. *Figure 5 shows an excerpt of two sample dataset available on the D3 site (at* `http://bl.ocks.org/josiahdavis/a3534073492ca37b3682` *and* `https://bl.ocks.org/mbostock/3883245`*, respectively). The first dataset includes 6 variables, the first three with type nominal, the remaining three with type ratio. The second dataset includes 2 variables with type interval and ratio respectively.*

In the context of a specific platform that implements visualization services, each visualization type v is characterized by a set of *graphical coordinates*, $G(v)$. Each graphical coordinate $g \in G(v)$ can be either mandatory or optional (denoted by Boolean function $mand(g)$), independent or dependent (denoted by Boolean function $indep(g)$). Like in Definition 2, and consistently with a previous approach[26], the suitability of g to be used for displaying a variable of type t is assessed by a *PSM suitability function*:

Definition 7. PSM Suitability Function. *A* PSM suitability function *is a (total) function* $\tau : G(v) \times T \rightarrow s$ *where* $s \in \{\text{unfit}, \text{discouraged}, \text{acceptable}, \text{fit}\}$ *is a score.*

Here, the semantics of the scores (to be defined by a visualization expert for the specific platform adopted) is as follows:

- **unfit** means that the graphical coordinate *cannot be used* to display the variable type, typically because of the parameter type required by the visualization service. For instance, the X coordinate of a line graph cannot be used to display a nominal variable in D3 because the service only accepts numbers.
- **discouraged** means that the graphical coordinate *can be used* to display the variable type, but this distorts the very nature of that variable. For instance, the label coordinate of a pie chart can be used to display a number in D3, but —conceptually speaking— the label should be mapped onto a qualitative rather than a quantitative variable.
- **acceptable** means that the graphical coordinate *is compatible* with the variable type, though it may fail to emphasize some of the features of that variable. For instance, the label coordinate of a pie chart can be used to display an ordinal variable in D3, but it will give no specific emphasis to the ordering of values.
- **fit** means that the graphical coordinate is *fully compatible* with the variable type. For instance, the arc coordinate of a pie chart is perfectly suitable for visualizing a ratio variable.

Table 6 shows the graphical coordinates and the related scores for eight visualization types in their D3 implementation (all scores for Graph are unfit because no graph-oriented visualization types are included among the eight ones we picked as a reference in the paper).

A binding is an assignment of all or some of the variables of a dataset to the graphical coordinates of a visualization type. To be feasible, a binding must assign one variable to each mandatory graphical coordinates; besides, the scores for all assignments must be different from unfit.

Definition 8. Binding. *Given visualization type v with graphical coordinates $G(v)$, and dataset D with variables $A = \{a_1, \ldots, a_n\}$, a* binding *of D onto v is an injective, partial function $\beta : A \rightarrow G(v)$ such that*

(a) the image of β includes all the $g \in G(v)$ for which $mand(g) = TRUE$, and

(b) for all $a_i \in \hat{A}$, where $\hat{A} = \{a_i \in A : \exists \beta(a_i)\}$ ($\hat{A} \subseteq A$, called the active domain *of β), it is $\tau(\beta(a_{i_j}), type(a_{i_j})) >$* unfit.

For instance, reconsidering the sales revenue example mentioned above to be visualized with a pie chart, the two possible bindings (sketched in Figure 6) are

$$\beta(\text{Continent}) = \text{Label}$$
$$\beta(\text{SalesRevenue}) = \text{Arc}$$

and

$$\beta(\text{Continent}) = \text{Arc}$$
$$\beta(\text{SalesRevenue}) = \text{Label}$$

Of these, only the first one is actually compatible with the visualization type.

As done in Section "Translating the CIM into the PIM" to compare visualization types, to compare bindings we introduce a notion of dominance aimed at proposing to the user only the best bindings, i.e., those in the skyline. Intuitively, a binding is better than another if it assigns at least the same variables, and if the related scores are not worse.

giniDummy.csv

```
Metric,SubCategory,Category,TotalValue,ProductConcentration,CustomerConcentration
Cost,Copiers,Technology,93910.21,0.030137063,0.087770725
Cost,Machines,Technology,185853.87,0.727932545,0.234178493
Cost,Supplies,Office Supplies,47862.64,0.869843455,0.967766883
Cost,Bookcases,Furniture,118352.55,0.185026777,0.627607548
Cost,Envelopes,Office Supplies,9512.23,0.080804313,0.914777644
Cost,Fasteners,Office Supplies,2074.76,0.303772145,0.029366578
Cost,Tables,Furniture,224691.01,0.742161958,0.886106234
Cost,Labels,Office Supplies,6940.06,0.06212297,0.655249515
```

data.tsv

```
date  close
24-Apr-07 93.24
25-Apr-07 95.35
26-Apr-07 98.84
27-Apr-07 99.92
30-Apr-07 99.80
1-May-07  99.47
2-May-07  100.39
```

Figure 5. Two sample datasets

Table 6. Graphical coordinates (in the D3 library) and PSM suitability scores for eight visualization types (m=mandatory, o=optional, i=independent, d=dependent)

			Nominal	Ordinal	Interval	Ratio	Tree	Graph
Single line graph:	X	(m, i)	unfit	unfit	fit	fit	unfit	unfit
	Y	(m, d)	unfit	unfit	fit	fit	unfit	unfit
Marked line graph:	X	(m, i)	unfit	unfit	fit	fit	unfit	unfit
	Y	(m, d)	unfit	unfit	fit	fit	unfit	unfit
Pie chart:	Label	(m, i)	fit	acceptable	discouraged	discouraged	unfit	unfit
	Arc	(m, d)	discouraged	discouraged	discouraged	fit	unfit	unfit
Bubble graph:	X	(m, i)	acceptable	acceptable	fit	fit	unfit	unfit
	Y	(m, i)	acceptable	acceptable	fit	fit	unfit	unfit
	Size	(o, d)	unfit	unfit	discouraged	fit	unfit	unfit
	Shape	(o, d)	fit	discouraged	unfit	unfit	unfit	unfit
	Color	(o, d)	fit	acceptable	acceptable	acceptable	unfit	unfit
Grouped column graph:	X	(m, i)	acceptable	fit	discouraged	discouraged	unfit	unfit
	Height	(m, d)	unfit	unfit	discouraged	fit	unfit	unfit
	Group	(o, i)	acceptable	fit	discouraged	discouraged	unfit	unfit
	Color	(o, i)	fit	acceptable	acceptable	acceptable	unfit	unfit
Heat map:	X	(m, i)	acceptable	acceptable	fit	fit	unfit	unfit
	Y	(m, i)	acceptable	acceptable	fit	fit	unfit	unfit
	Value	(m, d)	unfit	unfit	fit	fit	unfit	unfit
Dendrogram:	Hierarchy	(m, i)	unfit	unfit	unfit	unfit	fit	unfit
	Value	(o, d)	unfit	unfit	acceptable	fit	unfit	unfit
Tree map:	Hierarchy	(m, i)	unfit	unfit	unfit	unfit	fit	unfit
	Size	(m, d)	unfit	unfit	discouraged	fit	unfit	unfit
	Color	(o, d)	fit	acceptable	acceptable	acceptable	unfit	unfit
	Label	(o, d)	fit	discouraged	discouraged	discouraged	unfit	unfit

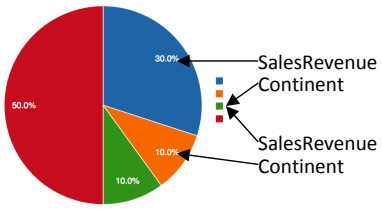

Figure 6. The two bindings for the sales revenue example

Definition 9. Binding Dominance. *Given two distinct bindings of D onto v, β and β' with active domains $\hat{A}$ and $\hat{A}'$ respectively, we say that* β dominates β', *denoted* $\beta \blacktriangleright \beta'$, *iff either*

(1a) $\hat{A} \equiv \hat{A}'$,

(1b) $\exists j : a_{i_j} \in \hat{A} \cap \hat{A}' \wedge$

$$\tau(\beta(a_{i_j}), type(a_{i_j})) > \tau(\beta'(a_{i_j}), type(a_{i_j})),$$

and

(1c) for all other $j : a_{i_j} \in \hat{A} \cap \hat{A}'$ *it is*

$$\tau(\beta(a_{i_j}), type(a_{i_j})) = \tau(\beta'(a_{i_j}), type(a_{i_j}))$$

or

(2a) $\hat{A} \supset \hat{A}'$ *and*

(2b) for all $j : a_{i_j} \in \hat{A} \cap \hat{A}'$ *it is*

$$\tau(\beta(a_{i_j}), type(a_{i_j})) \geq \tau(\beta'(a_{i_j}), type(a_{i_j}))$$

This is to say that β dominates β' if either (1a) β and β' assign the same variables, (1b) β is better than β' on at least one coordinate, and (1c) β and β' are equivalent on all other coordinates, or (2a) β assigns more coordinates than β' and (2b) β is not worse than β' on all the coordinates assigned by β'.

Definition 10. PSM Skyline. *The* PSM skyline *for D and v is the set of bindings of D onto v that are not dominated by any other binding.*

Example 5. *Consider again the first, n-dimensional dataset in Example 4, featuring 3 nominal and 3 ratio variables. We assume that, based on her analysis objectives, the user has selected bubble graph out of the PIM skyline as the preferred visualization type. As summarized in Table 6, in D3 a bubble graph has 5 graphical coordinates:* X, Y *(both mandatory and requiring to be preferably either an interval or a ratio, but possibly also a nominal or an ordinal),* Shape *(optional, to be preferably bound to a nominal),* Size *(optional, to be preferably bound to a ratio but possibly also to an interval), and* Color *(optional and compatible with all variable types). Based on these constraints, a possible binding (corresponding to the visualization in Figure 7) is as follows:*

$$\beta(\mathsf{ProductConcentration}) = \mathrm{X}$$
$$\beta(\mathsf{CustomerConcentration}) = \mathrm{Y}$$
$$\beta(\mathsf{TotValue}) = \mathrm{Size}$$
$$\beta(\mathsf{Category}) = \mathrm{Color}$$

Note that binding

$$\beta(\mathsf{ProductConcentration}) = \mathrm{X}$$
$$\beta(\mathsf{CustomerConcentration}) = \mathrm{Y}$$
$$\beta(\mathsf{TotValue}) = \mathrm{Size}$$

is dominated by the previous one because its active domain is smaller. Overall, the PSM skyline for this example consists of all the bindings where X, Y, *and* Size *are bound to any permutation of* ProductConcentration, CustomerConcentration, *and* TotValue, *while* Color *and* Shape *are bound to either* Metric, SubCategory, *or* Category.

Case Study and Evaluation

To evaluate SkyViz we have implemented a Java prototype whose web interface (developed in Javascript) supports the declaration of the visualization context and returns the best visualizations; the underlying database is MySQL and the reference graphical library for visualizations is D3. Both the PIM and the PSM skylines are computed using the *Maintaining the Window as a Self-organizing List* variant of the block-nested-loops algorithm[8]. Then we have let the users of the four pilot applications of the TOREADOR Project use this prototype to express a visualization context for their analytics use cases, asked them to select one preferred visualization type out of those proposed by the system, and showed them the visualizations produced using the bindings in the PSM skyline. Here we will describe two use cases out of those evaluated, namely, the one related to threat detection and prevention in software ecosystems, and the one related to predictive maintenance of solar farms.

Threat Detection Systems

Threat Detection Systems (TDS) in software ecosystems[47] detect potential attacks on the application landscape by gathering and analyzing log data, such as user change logs, security audit logs, remote function call gateway logs, and transaction logs. Logs are pre-processed, anonymized, translated into a common format, and analyzed by pattern or anomaly detection algorithms, which can highlight suspicious events. On top of the generated events and alerts, a detailed investigation is performed by a human expert to decide if a real attack was detected or was a false positive. However, with the increasing size and complexity of software systems, the volume and diversity of log data are becoming major issues. Customers use a large spectrum of different systems and adopt a wide range of data security policies. As a result, including and managing these heterogeneous log files currently requires a significant customization effort, especially when they contain sensitive and personal information (e.g., user IDs, IP addresses), come from logs of multiple customers, or are accessed via a third party (e.g., a cloud provider) running the TDS. Similarly, customers often need different security analyses depending on the security context, industrial sector, and risk management policies.

For simplicity, we focus on a simple, but relevant, scenario for TDS: security incident analysis through usage of anomaly detection analytics. The major challenge when searching for security incidents lies in the ability either to detect a deviation from a normal, standard behavior (unplanned anomalous activity) during or outside an exceptional process (planned anomalous activity), or to detect regular malicious activity merged into the normal state of operations (unplanned ordinary activity such as advanced persistent threat or repeated fraud). In this context, starting from a dataset where each row corresponds to a network node and is labelled with the size of data exchanged, the transaction type, and the user who activated the service, a clustering algorithm is applied. The users' declaration for the data visualization

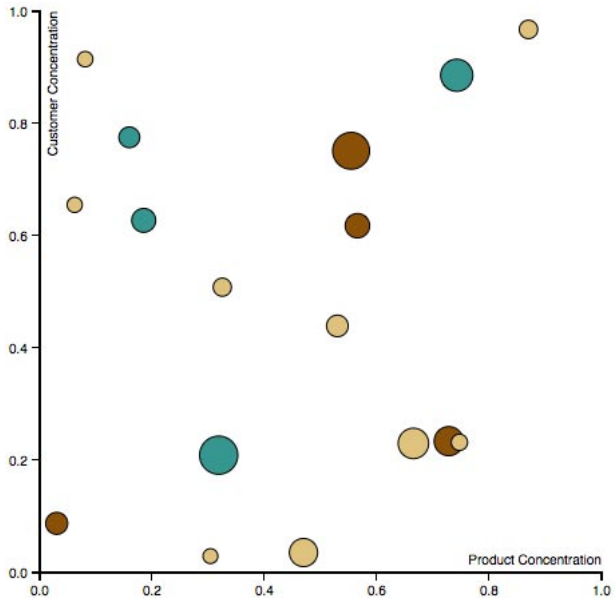

Figure 7. Bubble graph for Example 5

area is as follows:

$$\begin{aligned} \textit{Goal} &= \text{Cluster} \\ \textit{Interaction} &= \text{Overview} \\ \textit{User} &= \text{Tech} \\ \textit{Dimensionality} &= \text{n-dimensional} \\ \textit{Cardinality} &= \text{High} \\ \textit{Independent Type} &= \text{Ratio} \\ \textit{Dependent Type} &= \text{Nominal} \end{aligned}$$

with no priorities, which translates into the following visualization context:

$$C = \{\text{Cluster}, \text{Overview}, \text{Tech}, \text{n-dimensional}, \text{High}, \text{Ratio-i}, \text{Nominal-d}\}$$

$$\text{Cluster} \overset{C}{\sim} \text{Overview} \overset{C}{\sim} \text{Tech} \overset{C}{\sim} \text{n-dimensional} \overset{C}{\sim} \text{High} \overset{C}{\sim} \text{Ratio-i} \overset{C}{\sim} \text{Nominal-d}$$

The dominance relationships induced on visualization types by C, computed as in Section "Translating the CIM into the PIM" based on the scores excerpted in Table 7, are shown in Figure 8. The corresponding PIM skyline includes bubble graph, heat map, and tree map. Out of these three, the user selected bubble graph.

To move to the PSM, we consider the details of the dataset resulting from clustering, whose main variables are

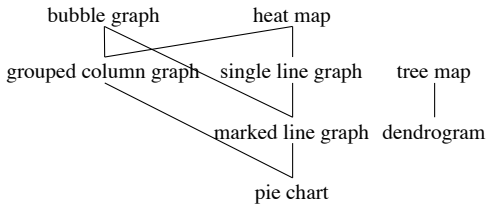

Figure 8. Dominance relationships for the TDS case study

ScaleDataTransfer (ratio), User (nominal), TransactionType (nominal), and Cluster (nominal). Here is an excerpt from the dataset:

ID	DataSent	ScaleDataTransfer	User	Trans.Type	Cluster
32	916	−0.46290193656	u1	t3	6
232	967	−0.23589474886	u2	t5	5
432	1130	0.489638027516	u1	t2	6
632	1063	0.191412898577	u2	t3	4
832	1121	0.449577935569	u1	t1	2

As shown in Table 6, the graphical coordinates for bubble graphs are X, Y (mandatory and independent), Size, Shape, and Color (optional and dependent). Two possible bindings are

$$\begin{aligned} \beta(\mathsf{ScaleDataTransfer}) &= \text{X} \\ \beta(\mathsf{TransactionType}) &= \text{Y} \\ \beta(\mathsf{User}) &= \text{Shape} \\ \beta(\mathsf{Cluster}) &= \text{Color} \end{aligned}$$

Table 7. Suitability tuples for eight visualization types with reference to the visualization context of the TDS case study

		single line graph	marked line graph	pie chart	bubble graph
Goal:	Cluster	unfit	unfit	unfit	acceptable
Interaction:	Overview	fit	fit	fit	fit
User:	Tech	fit	fit	acceptable	fit
Dimens.:	n-dimensional	unfit	unfit	unfit	fit
Cardinality:	High	fit	discouraged	discouraged	discouraged
Ind. Type:	Ratio-i	fit	fit	discouraged	fit
Dep. Type:	Nominal-d	unfit	unfit	unfit	fit

		grouped column graph	heat map	dendrogram	tree map
Goal:	Cluster	acceptable	acceptable	fit	fit
Interaction:	Overview	fit	fit	fit	fit
User:	Tech	fit	fit	fit	fit
Dimens.:	n-dimensional	fit	fit	unfit	acceptable
Cardinality:	High	discouraged	fit	acceptable	acceptable
Ind. Type:	Ratio-i	acceptable	fit	unfit	unfit
Dep. Type:	Nominal-d	unfit	unfit	unfit	fit

and

$$\beta(\mathsf{ScaleDataTransfer}) = \text{Size}$$
$$\beta(\mathsf{TransactionType}) = \text{Y}$$
$$\beta(\mathsf{User}) = \text{X}$$
$$\beta(\mathsf{Cluster}) = \text{Color}$$

Of these, the first one (corresponding to the visualization in Figure 9) dominates the second one and is the one proposed to the user. Though this visualization is probably not 100% optimal since it does not clearly show shapes (which represent the User variable), it is the one actually preferred by the user since it properly emphasizes clusters.

Predictive Maintenance of Solar Farms

This pilot is related to a global market leader in the development, acquisition, and long-term management of international large-scale solar projects and smart energy solutions. It has developed an asset management platform whose main goal is to provide, in a timely and concise manner, information to the users on the operation of the solar farms. All data originating from the field are forwarded to this platform, where they are stored and processed.

The use case we discuss here is related to the prediction of equipment maintenance based on historical data about equipment anomalies in work cycles of the devices (inverters, transformers, smart meter failures). To this end, data from the large-scale solar plants of three years and from residential assets are analyzed to prevent anomalies regarding spikes in voltage, giving frequency response of the grid quality, and receiving temperature of inverters and ampere information. Specifically, analyses are focused on the *mean time between failures*, i.e., the predicted elapsed time between inherent failures of a mechanical or electronic system during normal system operation.

Here we focus on a 3-dimensional dataset that includes, for 65 customers, a customer identifier, the mean time (in days) between battery charging failures, ChargingMTBF, and the one for which no power was generated by the solar panels, NoPowerMTBF. The visualization context declared by the users is as follows:

$$C = \{\text{Comparison}, \text{Filter}, \text{Lay}, \text{n-dimensional}, \text{High}, \text{Nominal-i}, \text{Ratio-d}\}$$

$$\text{Nominal-i} \overset{C}{\sim} \text{Ratio-d} \overset{C}{\succ} \text{Comparison} \overset{C}{\sim} \text{Filter} \overset{C}{\sim} \text{Lay} \overset{C}{\sim} \text{n-dimensional} \overset{C}{\sim} \text{High}$$

The PIM skyline for this visualization context includes grouped column graph, dendrogram, and tree map; by discarding the visualization types that feature one or more unfit scores, only grouped column graph and tree map are left. The user clearly selected grouped column graph, which is more well-known and intuitive for a lay user.

To call the D3 library for creating grouped column graphs, we had to transform the dataset by replacing the two variables ChargingMTBF and NoPowerMTBF with two variables MTBF and FailureType; the former stores the values of the two previous ratio variables, while the latter describes the values of the former and can take two nominal values, *charging* and *no power*. There are two bindings in the PSM skyline: in both, MTBF is bound to Height; Customer and FailureType are bound to X and Color or vice versa. Of the two corresponding visualizations, users selected the one that binds Customer to X, since the customers are too many

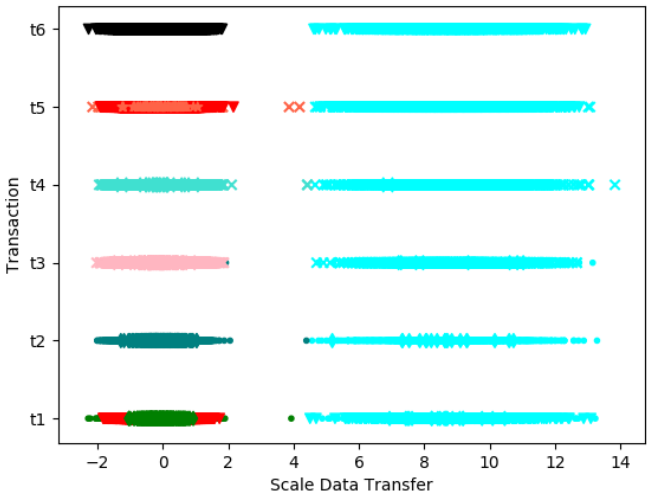

Figure 9. Data visualization using a bubble graph for the TDS case study

to be displayed using colors. The result is shown in Figure 10.

The fact that users assigned to customers the role of the independent variable and to the mean times that of the dependent variables, led SkyViz to miss the chance of proposing a bubble graph using the X and Y axis to represent mean times as shown in Figure 11; such bubble graph, coupled with a detail-on-demand interaction for seeing the details of single customers, may indeed turn out to be the most effective visualization for the dataset. In fact, our choice of distinguishing independent from dependent data types in visualization contexts has two consequences: on the one hand, it gives users a closer control of the PIM skyline and creates a better connection to the next stage, i.e., the computation of the PSM skyline; on the other, should users fail in properly identifying the roles of variables, it may lead to missing interesting visualizations.

Evaluation

For a more critical evaluation of SkyViz, in this section we simulate some challenging scenarios.

First of all, we show a simple example for which SkyViz can suggest an effective visualization which a lay user would hardly think of. Consider a dataset including three variables: Quarter (interval), Country (nominal), and DollarSales (ratio). To visualize this dataset, a lay user would probably choose either a grouped line graph (e.g., using different colored lines for countries like in Figure 13.a), or a grouped column graph (using colors to represent countries like in Figure 13.b), or maybe even a bubble graph (placing countries and quarters on the X and Y axis, and using the bubble size to represent sales like in Figure 13.c). These intuitive bindings are summarized in Figure 12. In SkyViz, the user would declare the following visualization context:

$$C = \{\text{n-dimensional}, \text{Low}, \text{Interval-i}, \text{Nominal-i}, \text{Ratio-d}\}$$

Note that, to avoid biases, we have not specified the goal, interaction, and user coordinates. The PIM skyline for this context includes bubble graph, grouped column graph, and heat map (also single and marked line graph would be part of the skyline, but they can be excluded because they are scored as unfit on both the dimensionality and independent type coordinates). Indeed, heat maps can provide an effective visualization for the dataset at hand as shown in Figure 13.d.

In the next example we discuss the impact of user-defined priorities on the PIM skyline. As already mentioned, the skyline for flat dominance always includes the skyline for dominance, i.e, prioritizing coordinate values leads to reducing the PIM skyline. As a consequence, users must be aware that providing priorities for coordinates may lead them to miss some effective visualization types for their dataset. On the other hand, priorities are useful to deal with situations where requirements are potentially conflicting, and users are willing to sacrifice the effectiveness of visualization from some points of view to increase it from other points of view. For instance, consider the following visualization context:

$$C = \{\text{Comparison}, \text{Zoom}, \text{Lay}, \text{2-dimensional}, \text{Low}, \text{Interval-i}, \text{Interval-d}\}$$

declared by a lay user who wants to analyze a small 2-dimensional dataset providing daily temperatures in a given location during one month. The eight corresponding suitability sets are singled out in Table 8. The flat

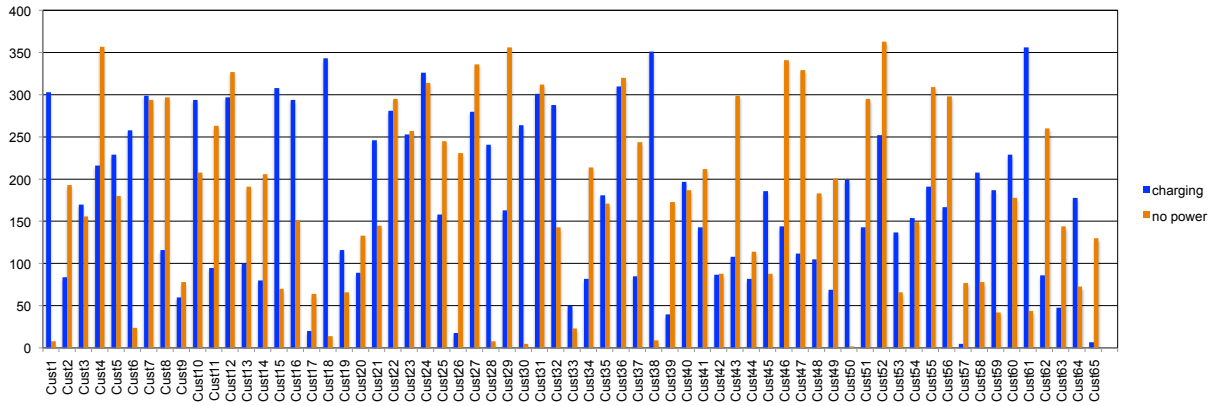

Figure 10. Data visualization using a grouped column graph for the predictive maintenance case study

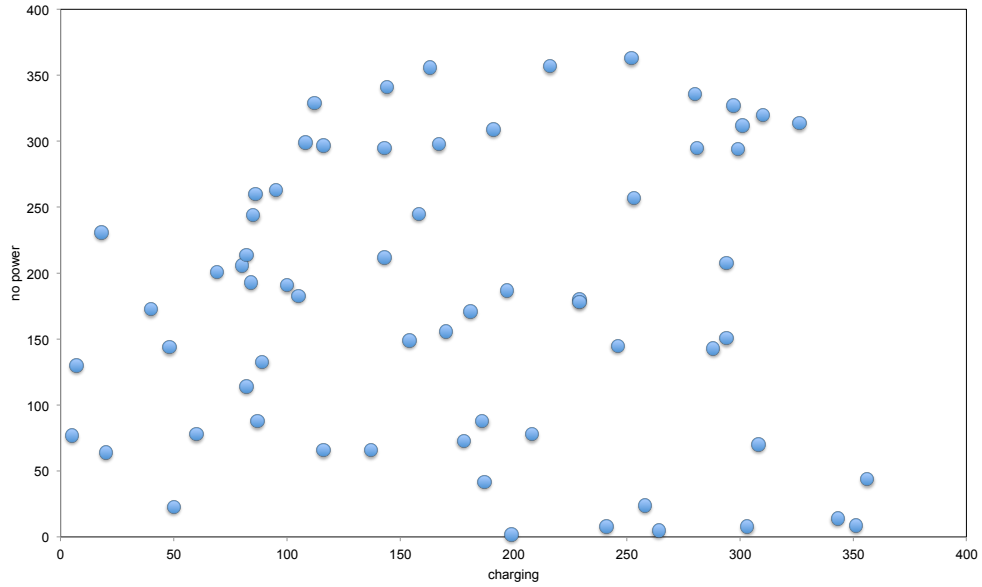

Figure 11. Data visualization using a bubble graph for the predictive maintenance case study

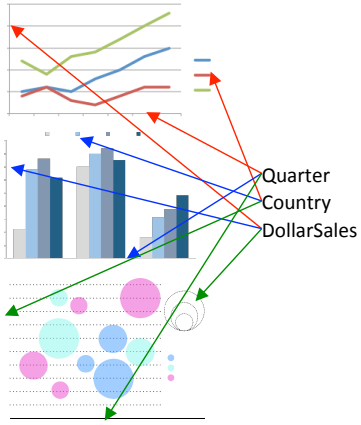

Figure 12. Intuitive bindings for the dollar sales example

PIM skyline for this visualization context includes all visualization types except single line graph and pie chart (both dominated by marked line graph). All visualization types feature at least one unfit, so there is no obvious way to reduce the PIM skyline. The reason for this is that the comparison goal is somehow conflicting with the nature of the dataset; indeed, a more reasonable choice for the goal coordinate would be trend (in which case, the only visualization type featuring no unfit in the PIM skyline would be marked line graph). Now, the PIM skyline can be reduced in different ways depending on the priorities given by the user. For instance, if she is mostly sure of the dataset features but not so much of her goals, she can declare priorities as follows:

$$\text{2-dimensional} \overset{C}{\sim} \text{Low} \overset{C}{\sim} \text{Interval-i} \overset{C}{\sim} \text{Interval-d} \overset{C}{\succ} \text{Comparison} \overset{C}{\sim} \text{Zoom} \overset{C}{\sim} \text{Lay}$$

In this case, the PIM skylines drastically reduces to include only marked line graph. Conversely, if the user privileges her goals but is unsure of the dataset features, she might declare

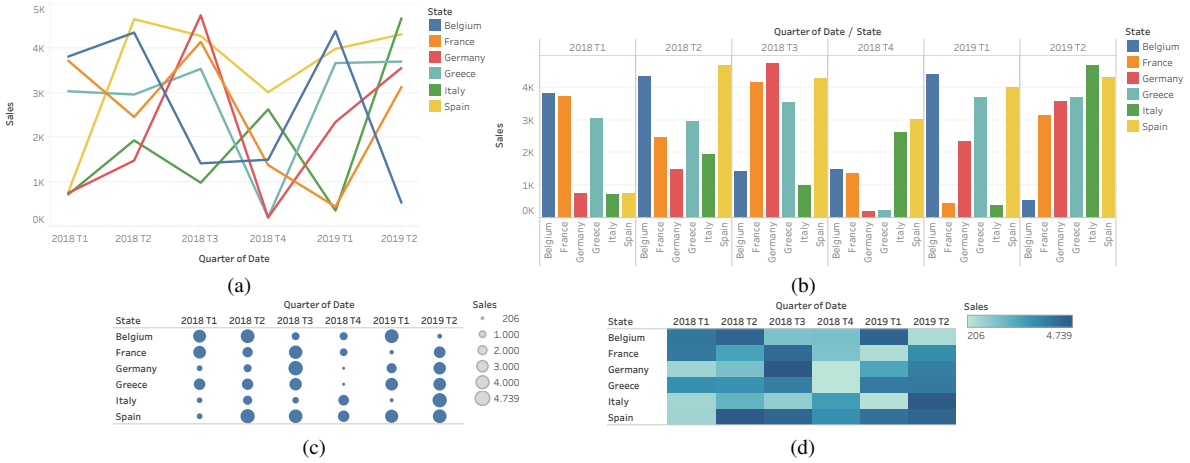

Figure 13. Alternative visualization for a 3-dimensional dataset: multiple line graph (a), grouped column graph (b), bubble graph (c), and heat map (d)

Table 8. Suitability tuples for eight visualization types with reference to the visualization context in Section "Evaluation"

		single line graph	marked line graph	pie chart	bubble graph
Goal:	Comparison	unfit	unfit	unfit	fit
Interaction:	Zoom	acceptable	acceptable	unfit	acceptable
User:	Lay	fit	fit	fit	acceptable
Dimens.:	2-dimensional	fit	fit	fit	unfit
Cardinality:	Low	acceptable	fit	fit	acceptable
Ind. Type:	Interval-i	fit	fit	discouraged	fit
Dep. Type:	Interval-d	fit	fit	discouraged	acceptable

		grouped column graph	heat map	dendrogram	tree map
Goal:	Comparison	fit	acceptable	discouraged	acceptable
Interaction:	Zoom	unfit	fit	fit	acceptable
User:	Lay	fit	acceptable	acceptable	discouraged
Dimens.:	2-dimensional	unfit	unfit	unfit	unfit
Cardinality:	Low	fit	acceptable	fit	fit
Ind. Type:	Interval-i	acceptable	fit	discouraged	discouraged
Dep. Type:	Interval-d	discouraged	fit	acceptable	discouraged

opposite priorities:

$$\overset{C}{\sim} \text{Comparison} \overset{C}{\sim} \text{Zoom} \overset{C}{\succ}$$

$$\text{Lay} \overset{C}{\sim} \text{2-dimensional} \overset{C}{\sim} \text{Low} \overset{C}{\sim} \text{Interval-i} \overset{C}{\sim} \text{Interval-d}$$

In this case, marked line graph is dominated, and the PIM skyline only includes heat map and dendrogram.

Finally, to provide a challenging example for the binding of variables, we consider the case in which the number of variables is larger than that of graphical coordinates. Consider an n-dimensional dataset including eight variables, of which two nominals, two ordinals, two intervals, and two ratios, and assume that the user selected bubble graph, which features five graphic coordinates, as the preferred visualization type. There are $\binom{8}{5} = 56$ different ways to select five variables out of the eight available for visualization; for each selection of five variables, these variables can be assigned to the five graphic coordinates in several ways. Overall, there are 6720 possible complete bindings, i.e., bindings that involve all five graphic coordinates. If also incomplete bindings are considered, i.e., bindings that leave optional graphical coordinates unassigned, the overall number of bindings increases to 8792. Of these, some can be excluded because the Size graphic coordinate cannot be bound to a nominal or ordinal variable, while the Shape graphic coordinate cannot be bound to an interval or ratio variable. Using SkyViz, the number of complete bindings returned to the user for each of the 56 possible selections of five variables decreases drastically. For instance, if both nominal variables and both interval variables are selected together with one ratio variable, the PSM skyline only includes the four all-fit bindings in which (i) the two nominal variables are bound to Shape and Color, (ii) the two interval variables are bound to X and Y, and (iii) the ratio variable is bound to Size. Clearly, the real problem when the number n of variables in the dataset is significantly larger that the number $|G(v)|$ of graphical coordinates is related the exponential growth

of the number $\binom{n}{|G(v)|}$ of possible selections of variables. In practical cases, we argue that this number will be drastically limited for two main reasons:

- The datasets we consider in big data analytics are typically resulting from data mining processes, which inherently include pre-selections of variables aimed at returning informative but concise patterns.
- The user may not be an expert in visualization techniques, but she is assumed to know the semantics of the data she is analyzing, so she will presumably pick for visualization those that she deems to be more relevant for her current analysis task.

Besides, we observe that users could even combine multiple visualization of different subsets of variables from the dataset into a single dashboard, so as to analyze the same phenomenon from different related points of view.

Conclusions

In this paper we have described SkyViz, a model-driven approach to automate the translation of the objectives declared by the user for visualizing the results of big data analytics into a set of most suitable, concrete visualizations. SkyViz enables users to specify a value for seven visualization coordinates and determines the set of Pareto-optimal visualization types. Then, the Pareto-optimal bindings between the variables of the dataset and the graphical coordinates used by the chosen visualization type are found and showed to the user in preview so she can pick the preferred ones. The suitability of visualization types for visualization coordinates, and that of graphical coordinates for variable types, is evaluated a-priori by an expert in visualization.

We observe that the separation between the two translations (from CIM to PIM and from PIM to PSM) enforced by model-driven approaches may introduce, like all divide-and-conquer approaches, some suboptimality in the result obtained. Indeed, there is a possibility that a specific binding for a visualization type not included in the PIM skyline is more suitable than the best binding for the visualization type chosen by the user among those in the PIM skyline. However, we remark that separating the two translations (i) enables a dramatic reduction of the computational complexity for computing skylines; (ii) lets the user intervene in the middle of the process and state her preferences by choosing one visualization type; and (iii) introduces a clean borderline between the (conceptual) declaration of the user's objectives, the (logical) specification of the visualization type, and the (physical) implementation of the visualization, thus decoupling the first two stages from the choice of the visualization platform.

We close the paper by recognizing that, although a basic characterization of data (dimensionality, cardinality, and data type) is actually directly obtained from the data itself, specifying visualization goals and interactions may indeed be a challenge for non-expert users. To fill this gap, we are currently working to extend SkyViz with a goal-oriented approach based on the i^* framework[48] to guide users across the process of expressing their requirements for visualization and automatically derive the coordinate values to be used for computing the skyline.

Another interesting direction for improving SkyViz lies in introducing some fine-tuning of specific features of visualizations, e.g., the color scale. While currently we rely on the default mechanisms provided by the graphics library, considering the context (e.g., by recognizing that a rainbow color scale is not optimal for continuous data due to the non-monoticity of luminance) and some fine-grained data features (e.g., the range and the number of distinct values for each single variable) would presumably lead to a visualization that is more intuitive and more incisive from the perceptual point of view.

Finally, some relevant questions arise in relationship to scalability, not in terms of performance but in terms of effectiveness. Indeed, should a very large number of possible visualization types be considered, our seven coordinates might no longer be sufficient to distinguish them (i.e., several visualization types might be described by exactly the same suitability tuples), in which case the PIM skyline would include a large number of (probably similar) visualization types. To cope with this situation, other coordinates should be added, but then the research question to be addressed would be how to select them in order to actually improve the discriminatory power of SkyViz. Answering this question is left for future work.

Acknowledgements

This work was partly supported by the EU-funded TOREADOR Project (contract n. H2020-688797).

References

1. Russom P. Big data analytics. Technical report, TDWI Best Practices Report, 2011.
2. Kleppe A, Warmer J and Bast W. *MDA explained - the Model Driven Architecture: practice and promise*. Addison-Wesley, 2003.

3. Ardagna C, Bellandi V, Damiani E et al. A model-driven methodology for big data analytics-as-a-service. In *Proc. IEEE Int. Congress on Big Data*. Honolulu, Hawaii, 2017.
4. Keim D. Exploring big data using visual analytics. In *Proc. EDBT/ICDT Workshops*. 2014.
5. Correll M, Li M, Kindlmann G et al. Looks good to me: Visualizations as sanity checks. In *Proc. IEEE InfoVis*. Berlin, Germany, 2018.
6. Bostock M. D3 — data-driven documents. https://d3js.org, 2017.
7. Golfarelli M, Pirini T and Rizzi S. Goal-based selection of visual representations for big data analytics. In *Proc. MoBiD*. Valencia, Spain, pp. 47–57, 2017.
8. Börzsönyi S, Kossmann D and Stocker K. The skyline operator. In *Proc. ICDE*. Heidelberg, Germany, pp. 421–430, 2001.
9. Shneiderman B. The eyes have it: A task by data type taxonomy for information visualizations. In *Proc. IEEE Symposium on Visual Languages*. pp. 336–343, 1996.
10. Wehrend S and Lewis C. A problem-oriented classification of visualization techniques. In *Proc. IEEE Visualization*. San Francisco, California, USA, pp. 139–143, 1990.
11. Keim DA. Information visualization and visual data mining. *IEEE Trans Vis Comput Graph* 2002; 8(1): 1–8.
12. Abela A. *Advanced presentations by design*. Pfeiffer, 2008.
13. Tory M and Möller T. Rethinking visualization: A high-level taxonomy. In *Proc. InfoVis*. Austin, USA, pp. 151–158, 2004.
14. Schulz H, Nocke T, Heitzler M et al. A design space of visualization tasks. *IEEE Trans Vis Comput Graph* 2013; 19(12): 2366–2375.
15. Börner K. *Atlas of knowledge: anyone can map*. MIT Press, 2015.
16. Peña O, Aguilera U and López-de-Ipiña D. Exploring LOD through metadata extraction and data-driven visualizations. *Program* 2016; 50(3): 270–287.
17. Dadzie AS and Rowe M. Approaches to visualising linked data: A survey. *Semant web* 2011; 2(2): 89–124.
18. Mackinlay JD. Automating the design of graphical presentations of relational information. *ACM Trans Graph* 1986; 5(2): 110–141.
19. Senay H and Ignatius E. A knowledge-based system for visualization design. *IEEE Computer Graphics and Applications* 1994; 14(6): 36–47.
20. Mackinlay JD, Hanrahan P and Stolte C. Show me: Automatic presentation for visual analysis. *IEEE Trans Vis Comput Graph* 2007; 13(6): 1137–1144.
21. Ananthanarayanan R, Lohia PK and Bedathur S. DataVizard: Recommending visual presentations for structured data. In *Proc. WebDB*. Houston, USA, pp. 3:1–3:6, 2018.
22. Voigt M, Pietschmann S, Grammel L et al. Context-aware recommendation of visualization components. In *Proc. eKNOW*. Valencia, Spain, pp. 101–109, 2012.
23. Roth S and Mattis J. Automating the presentation of information. In *Proc. Conf. on Artificial Intelligence Application*. Miami Beach, FL, pp. 90–97, 1991.
24. Casner SM. Task-analytic approach to the automated design of graphic presentations. *ACM Trans Graph* 1991; 10(2): 111–151.
25. Lange S, Schumann H, Müller W et al. Problem-oriented visualization of multi-dimensional data sets. In *Scientific Visualization*. World Scientific, 1995. pp. 1–15.
26. Zhang J. A representational analysis of relational information displays. *Int J Hum-Comput Stud* 1996; 45(1): 59–74.
27. Zhou MX and Feiner S. Automated visual presentation: From heterogeneous information to coherent visual discourse. *J Intell Inf Syst* 1998; 11(3): 205–234.
28. Healey CG, Kocherlakota S, Rao V et al. Visual perception and mixed-initiative interaction for assisted visualization design. *IEEE Trans Vis Comput Graph* 2008; 14(2): 396–411.
29. Marty R. *Applied security visualization*. Addison-Wesley, 2009.
30. Sun Y, Leigh J, Johnson AE et al. *Articulate*: A semi-automated model for translating natural language queries into meaningful visualizations. In *Proc. Smart Graphics*. Banff, Canada, pp. 184–195, 2010.
31. Chandra J and Madhu Shudan S. IBA graph selector algorithm for big data visualization using defense data set. *International Journal of Scientific & Engineering Research* 2013; 4(3): 1–7.
32. Bouali F, Guettala AE and Venturini G. VizAssist: an interactive user assistant for visual data mining. *The Visual Computer* 2016; 32(11): 1447–1463.
33. Gotz D and Wen Z. Behavior-driven visualization recommendation. In *Proce. IUI*. Sanibel Island, Florida, USA, pp. 315–324, 2009.
34. Rogowitz BE and Treinish L. An architecture for rule-based visualization. In *Proce. IEEE Visualization*. San Jose, California, USA, pp. 236–244, 1993.
35. Bergman LD, Rogowitz BE and Treinish L. A rule-based tool for assisting colormap selection. In *Proc. IEEE Visualization*. Atlanta, Georgia, USA, pp. 118–125, 1995.
36. Ibrahim IA, Albarrak AM and Li X. Constrained recommendations for query visualizations. *Knowl Inf Syst* 2017; 51(2): 499–529.
37. Leban G, Zupan B, Vidmar G et al. VizRank: Data visualization guided by machine learning. *Data Min Knowl Discov* 2006; 13(2): 119–136.
38. Wills G and Wilkinson L. AutoVis: Automatic visualization. *Information Visualization* 2010; 9(1): 47–69.

39. Vartak M, Rahman S, Madden S et al. SEEDB: efficient data-driven visualization recommendations to support visual analytics. *PVLDB* 2015; 8(13): 2182–2193.
40. Ehsan H, Sharaf MA and Chrysanthis PK. Efficient recommendation of aggregate data visualizations. *IEEE Trans Knowl Data Eng* 2018; 30(2): 263–277.
41. Wongsuphasawat K, Moritz D, Anand A et al. Towards a general-purpose query language for visualization recommendation. In *Proc. HILDA*. San Francisco, CA, p. 4, 2016.
42. Streit M, Schulz H, Lex A et al. Model-driven design for the visual analysis of heterogeneous data. *IEEE Trans Vis Comput Graph* 2012; 18(6): 998–1010.
43. Kano N, Nobuhiku S, Fumio T et al. Attractive quality and must-be quality. *Journal of the Japanese Society for Quality Control* 1984; 14(2): 39–48.
44. Berger C et al. Kano's method for understanding customer-defined quality. *Center for quality management* 1993; 2(4): 3–35.
45. Stevens SS. On the theory of scales of measurement. *Science* 1946; 103(2684): 677–680.
46. Mindolin D and Chomicki J. Preference elicitation in prioritized skyline queries. *VLDB J* 2011; 20(2): 157–182.
47. Oprea A, Li Z, Yen T et al. Detection of early-stage enterprise infection by mining large-scale log data. In *Proc. DSN*. Rio de Janeiro, Brazil, pp. 45–56, 2015.
48. Bresciani P, Perini A, Giorgini P et al. Tropos: An agent-oriented software development methodology. *Autonomous Agents and Multi-Agent Systems* 2004; 8(3): 203–236.