

RESEARCH ARTICLE

ModelViz: A Model-Driven Engineering Approach for Visual Analytics System Design

ALIREZA KHAPOUR¹, ANDREA VÁZQUEZ-INGELMO², RICARDO COLOMO-PALACIOS³, FRANCISCO J. GARCÍA-PEÑALVO², (Member, IEEE), AND ANTONIO MARTINI⁴

¹Faculty of Computer Science, Engineering and Economics, Østfold University College, 1757 Halden, Norway

²Research Group of Interaction and eLearning (GRIAL), Computer Science Department, Research Institute for Educational Sciences, University of Salamanca, 37008 Salamanca, Spain

³Escuela Técnica Superior de Ingenieros Informáticos, Universidad Politécnica de Madrid, 28660 Madrid, Spain

⁴Department of Informatics, University of Oslo, 0316 Oslo, Norway

Corresponding author: Alireza Khakpour (alireza.khakpour@hiof.no)

This work was supported by the DigiMat Project funded by the Research Council of Norway (RCN) under Project 296686.

ABSTRACT Visual analytics systems should be able to consolidate data from disparate sources, conduct exploratory analysis, create visualizations that suit different users, and integrate seamlessly with decision-making activities to support data-driven decision-making. However, current mainstream visual analytics solutions often lack support for all these requirements. To address this gap, we propose the use of model-driven engineering to design visual analytics systems. To demonstrate the feasibility of this approach, we developed a Domain-Specific Modeling Language (DSML) named ModelViz to design visual analytics systems for consumer goods supply chain applications. Furthermore, we present the work of our DSML, using data from a manufacturing company as a case study. Finally, we evaluated ModelViz quantitatively by comparing it with other similar works from the literature. Our results demonstrate that this approach meets the requirements and provides a promising direction for designing visual analytics systems by considering domain-specific aspects to help achieve business goals.

INDEX TERMS Decision support systems, domain-specific languages, model-driven engineering, supply chain management, visual analytics.

I. INTRODUCTION

In recent years, the information systems community has designed many Visual Analytics (VA) systems [1], [2], [3], [4], [5]. However, they are either general-purpose systems that do not meet specific domain requirements or are tailored exclusively to one application area, making them unsuitable for other domains.

On the other hand, VA requires specific skills and knowledge to develop the most effective visualizations for specific application domain requirements. However, users are typically either application domain experts who lack beneficial skills for VA tasks, such as programming [6], or VA experts who lack the necessary domain knowledge [7], such as sales forecasting. In this context, an appropriate VA system should support various user groups with varying levels of

expertise. In addition, a successful VA may support various decision-making activities and align with the activity and goals of the application domain [8]. This provides a goal-oriented VA that can contribute to improving the analytical results [9].

For example, supply chain VA includes conducting an exploratory analysis of data collected from heterogeneous data sources with the aim of supporting the business goals and decisions of various supply chain partners. Based on the study presented by Khakpour et al. [10], a VA system should be able to fulfill the following requirements:

- 1) The ability to streamline exploratory analysis, including unknown analytics tasks.
- 2) Consolidating data from disparate sources,
- 3) Assisting with creating optimal visual encodings for diverse users, and
- 4) Integrating into decision-making activities to achieve relevant business objectives.

The associate editor coordinating the review of this manuscript and approving it for publication was Asad Waqar Malik¹.

In general, data analysis can be categorized as either exploratory or directed [11]. Exploratory analysis involves investigating data to gain insights that may not be apparent to data scientists [12]. For instance, identifying hidden periodic patterns within data variables can help diagnose certain inconsistencies [13]. On the other hand, directed analysis typically aims to answer known questions through data analytics, such as analyzing the sales data to understand the sales variations in a year. Direct analysis also includes examining data to prove some known beliefs, such as analyzing market data to show why adding a product feature increased customer satisfaction. This type of analysis is termed direct because the analyst has a hypothesis or answer in mind and seeks evidence in the data to support it. This involves examining how data change within specific variables of interest. The example of supply chain VA involves unknown exploratory analysis tasks, mainly due to uncertainties associated with the nature of the supply chain. For example, an unexpected social phenomenon can affect the flow of supply and may only be identified through exploratory analysis.

Moreover, supply chain processes generate data through the work and collaboration of various partners, such as manufacturers, warehouses, distributors, and retailers. Therefore, data collected by various organizations should be gathered and consolidated for an effective VA. However, the nature of the data varies and each supply chain partner handles them differently. For example, retailers collect sales data from their customers' transactions and use it for sales forecasting [14]. By contrast, warehouses collect data on products moving out from their facilities to make decisions regarding their warehouse locations [15]. Therefore, supply chain VA should be flexible in collecting data from different sources and analyzing it for various purposes.

From a software engineering perspective, the process of designing a software system customized for a specific domain is referred to as domain engineering [16]. In this regard, Domain-Specific Modeling Languages (DSML) are used to abstract domain knowledge when designing a software system to meet the specific needs of that domain. Simultaneously, this approach can be applied to other domains to develop similar systems. In this paper, we propose an approach to model-driven engineering that supports the development of domain-specific VA systems. In order to demonstrate the feasibility of this approach, we designed a prototype, called ModelViz. ModelViz is a graphical DSML based on the metamodeling technique that supports the development of a supply chain VA. For this purpose, we examined VA as a comprehensive modeling activity in which VA architects can create VA models for various supply chain applications. Moreover, we developed a graphical workbench as a tool to aid users in interacting with the DSML and creating VA dashboards without the need to write code.

In the remainder of this paper, we first present the background and related work of this study. We then present the methods used to implement our proposal. Subsequently, we introduce the tool that was created and demonstrate its

usage through a case study. We will then evaluate the proposal by discussing how it meets the system requirements and a quantitative comparison with similar works from the literature. Finally, we conclude the study and discuss potential future research directions.

II. BACKGROUND AND RELATED WORK

In this section, we investigate previous utilizations of Model-driven Engineering (MDE) for VA applications. We also briefly explain what supply chain visual analytics means, which is the focus of our case studies.

A. MODEL-DRIVEN ENGINEERING FOR VISUAL ANALYTICS

In recent years, VA has been widely applied in various fields, and developers have utilized various techniques to create an optimal system for each specific application. For example, a study presented by González-Torres et al. [17] developed a VA system for software project management using a Java-based architecture. While the proposed system provided specific capabilities to end-users, changing user requirements necessitated extensive modifications of the architecture of the system. Logre and Déry-Pinna [18] proposed the use of MDE to facilitate the design of VA systems by employing Domain-Specific Languages (DSL) for modeling and prototyping. In order to support the designers in choosing the correct visualization, the authors proposed using the Software Product Line (SPL) paradigm. A feature model is created for each visualization type that guides the designer in choosing the relevant visuals based on the visualization library in use and the corresponding analytics concern. In our proposal, we have developed a catalog based on the developed framework for decision areas, visualization goals and tasks, and their corresponding visualization types. In this way, the analyst's thinking process, from the targeted decision area to the choice of data and visualization type, can be captured.

Smart cities are another application domain that utilize VA systems to leverage the potential of data. However, existing visualization systems often have a predefined set of capabilities developed based on the requirements gathered before system development [19], [20]. This renders them unsuitable for similar domains. To address this issue, Rojas et al. [20] proposed a DSL for creating smart-city dashboards. This DSL can be utilized for any city by integrating domain concepts into the system-development process.

Despite its generality, this approach may not be suitable for other IoT applications that require domain-specific VA systems, such as industrial sensor analytics. To achieve this goal, Logre et al. [21] utilized DSLs to incorporate domain-specific variability and simplicity, thereby enhancing the reusability. Both of the proposed DSLs in [20] and [21] can only be used to create visualization dashboards for specific information and do not support the analytics activity. In our proposal, we provide a guide for analytics activity by integrating the decision-making process into the abstract syntax of the DSL.

In conclusion, previous attempts to use MDE for VA have focused on addressing only some of the challenges related to VA without considering the interconnectedness of these challenges. These attempts have addressed specific domain requirements to facilitate VA design or domain variability separately. However, there is still a gap in the literature regarding a comprehensive approach that considers all aspects of using an MDE for VA system design. This implies that there is a need for further research that considers all of these aspects to develop a more effective approach to using MDE for VA system design.

B. SUPPLY CHAIN VISUAL ANALYTICS

Supply chain activities are data rich and require an integrated VA system to fully leverage their data potential. For instance, supply chain partners in the fast-moving consumer goods industry, such as manufacturers, distributors, and retailers, generate and collect enormous amounts of data from their daily operations in production, logistics, and sales. These data contain hidden insights and information that are challenging to reveal without a proper VA system in place [10].

Moreover, to effectively relate and utilize the analysis in operations and decision support, the VA system should be customized and integrated to meet specific domain requirements [22]. Previously, various studies have explored the application of VA in decision support systems for supply chain management [10]. These studies often focus on specific application domains such as warehousing [23], manufacturing [24], and retail [25]. Alternatively, they may explore generic VA systems, as presented by Bostock and Heer [1], and Satyanarayan et al. [5]. In general, studies have considered three main aspects that a supply chain VA system tends to address: 1) business-driven analytics, 2) data handling, and 3) visualization recommendations. Therefore, we aimed to address these issues using a unified approach.

Moreover, data-driven decision-making is a time-constrained task, and decision-makers must respond rapidly to different situations to have a sustainable business process [26]. In this regard, Teruel et al. [26] suggest using goal-oriented decision-making based on business processes. Other previous works explored how VA can be used to support decision-making activities [27], [28]. Our focus in this study is the integration of the decision-making process into the VA system development, which we discuss next.

III. METHODS AND MATERIALS

In this section, we first present the VA development framework, which serves as the basis for this study. We then discuss how we adapted the MDE approach as the methodology for our study.

A. VISUAL ANALYTICS DEVELOPMENT FRAMEWORK

In a recent study [29] regarding the development of VA systems, the authors argued that designing VA systems for decision support systems requires addressing the following significant questions:

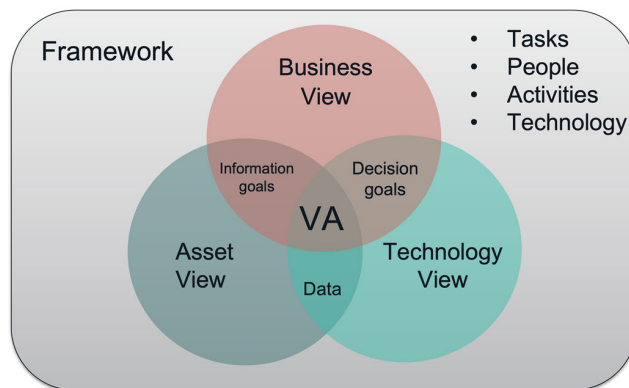


FIGURE 1. Visual analytics design framework.

- 1) What is the specific VA task, and how does it support a decision area? This question explores the connection between VA and business problems.
- 2) What data is required, and where does this data come from? This question investigates the available data assets.
- 3) What is the most suitable visualization design, and how can we effectively design it? This question identifies the technology to be used for creating visualizations.

To address these questions, the authors propose a development framework for VA based on three distinct perspectives of the problem domain. They also suggested using an MDE methodology to implement these perspectives. Fig. 1 presents an overview of the framework that consists of different views: 1) *the Business View*, 2) *the Asset View*, and 3) *the Technology View*. The *Business View* primarily focuses on answering the initial question by identifying the decision area that the VA aims to support, the specific VA task that aligns with the corresponding decision, and information goals. *The Asset View* addresses the second question by identifying the available data sources and finding a way to consolidate the required data from various sources. *The Technology View* deals with identifying the optimal visualization design for a given task and implementing it into a VA dashboard.

The MDE methodology addresses these views through multiple approaches, including, 1) characterizing and abstracting the decision areas, goals, and tasks of the VA, 2) incorporating data ingestion and VA design into the modeling activity, and 3) automating the design and implementation of a VA dashboard. In this case, we consider the VA system development framework as a blueprint or template for creating VA systems, and the MDE serves as an instantiation of the framework as it offers specific guidelines and practices for implementing it.

B. MODEL-DRIVEN ENGINEERING METHODOLOGY

MDE is an approach to system development that utilizes multiple layers of abstraction to facilitate the modeling of interactions between design logic and implementation [30].

We instantiated the MDE process and created a multi-layered approach for VA system development that

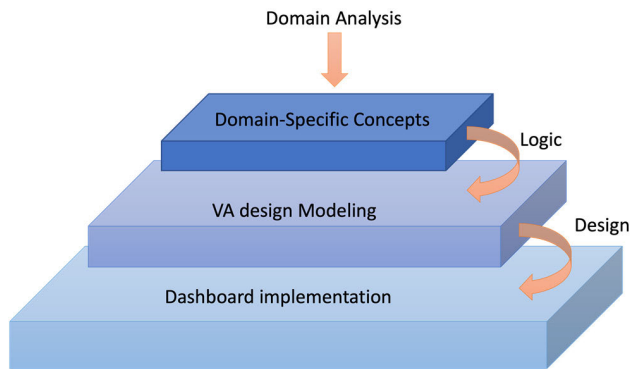


FIGURE 2. Multi-layer approach for VA system design.

begins with domain analysis to identify domain-specific concepts that form the basis for the design logic of the VA system. However, there is a significant gap between the high-level domain concepts that the domain experts use to define their requirements and the low-level VA tasks. Considering the dynamic nature of the requirements, it would be challenging to overcome this gap without a suitable method [31]. MDE addresses this challenge by employing a modeling technique that facilitates the separation of domain concepts and the automatic generation of key system elements [32], such as visualization artifacts (in the case of using MDE for VA), directly from the models.

Adding to that, MDE approaches the VA as a modeling activity that provides the following advantages to the VA development:

- 1) Modeling VA provides guidance for VA development.
- 2) Using domain-specific modeling can create a balance between generalization and specificity.
- 3) The created VA model can be used as a way of describing the analytical activity for future traceability and communication.
- 4) Metamodeling provides scalability for the VA system. Scalability can be achieved by adding more domain elements to the decision-making catalog or adding more visual specifications to the visualization catalog.

In the first layer of the approach, the domain-specific concepts provide design logic to the next layer to guide VA design modeling. For example, if a specific decision area that the VA aims to support is a key concept, it can be incorporated into the VA design model. The VA design model provides guidance for design choices, such as selecting a bar chart for the dashboard, and it is implemented in the final layer using a specific programming language. By employing this multi-layered approach, we can develop VA systems that are customized to meet the specific requirements of the domain, leading to improved effectiveness and efficiency in system design. Fig. 2 depicts the adapted multi-layer approach. Fig. 3 demonstrates the process of VA dashboard development using the MDE.

First, the metamodeling technique is used to characterize the domain elements, and a DSML is defined to provide the

corresponding model notations. Domain analysis is the first step in defining the abstract syntax of DSML [33]. On the other hand, a graphical modeling workbench was created based on visual modeling to simplify the interaction with the metamodel, serving as the concrete syntax of the DSML. The workbench simplifies the interaction with the metamodel for creating a Domain-Specific Model (DSM). Once a model is created, a model-to-text transformer is developed based on a code template to transform the DSMs into the source code of a dashboard. Finally, source codes were deployed using a deployment environment. In the following section, we demonstrate the implementation of this approach by creating a prototype version of a VA system.

IV. DESIGN OF MODELVIZ

In this section, we present an implementation of the above approach using a prototype called ModelViz. The prototype was used to evaluate the feasibility of the approach in a case study. Based on the presented method, the prototype includes the following four parts: 1. Metamodel, 2. Graphical Modeling workbench, 3. Model-to-text transformers and 4. Deployer. The source code for ModelViz is available in the GitHub repository [34]. Next, we describe the implementation of the components.

A. METAMODEL

ModelViz contains a metamodel that is created based on captured concepts of the domain. The metamodel is created using class diagrams following the Ecore format of the Eclipse Modeling Framework (EMF) [35]. Fowler [36] discussed the importance of perspectives when designing a model. He describes three modeling perspectives. Conceptual perspective that represents concepts within the domain. A specification perspective considers the type or interface of a class, whereas an implementation perspective manifests the implementation of a class. In what follows, we described the structure of the ModelViz metamodel considering these perspectives. Fig. 4 shows the proposed metamodel for a supply chain VA application.

Each class of the metamodel has a data-type catalog that provides a perspective on the specifications. Based on the previously described framework, the meta-model conceptually defines a dashboard based on three main classes: 1) A “visualization class” that describes the types of visual encodings and their specifications, such as mark and channel (or encoding) types, 2) A “decision area class” that describes the decision-making elements, including the decision area that the VA seeks to support, the visualization goal that the VA seeks to achieve, and the corresponding VA task that should be carried out in that regard, and 3) A “data class” that identifies the required data sources.

Every *DecisionArea* class can contain only one visualization ([1..1] bounds) that tends to achieve 1 to many ([1..*] bounds) visualization goals. That is, achieving different visualization goals through one visualization and supporting a single decision area. Following that, every visualization goal

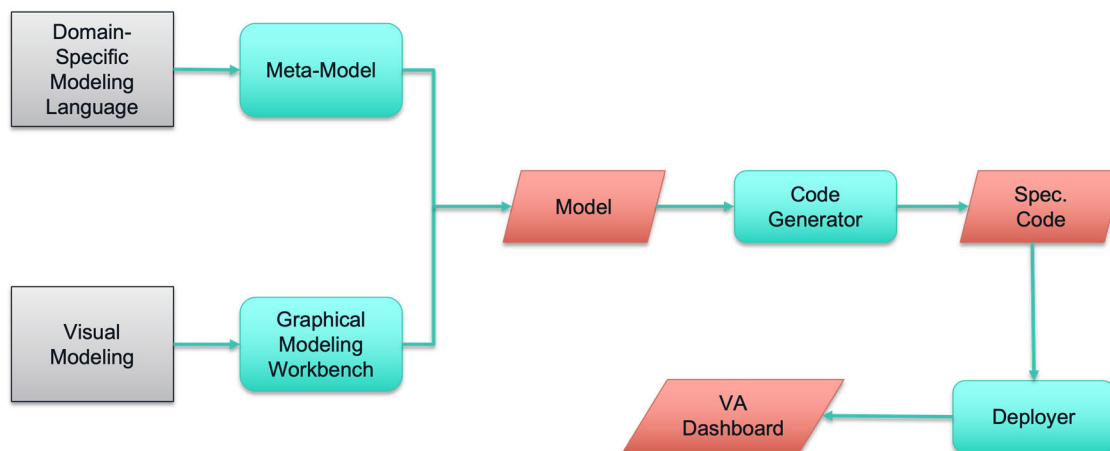


FIGURE 3. Process of VA dashboard development using MDE.

can contain 1 to many *VATask*, depicting that multiple VA tasks may need to be performed to achieve a single goal. The relation between the visualization class and data class is 1 to many, which provides the ability to have different data sources for different visualizations in a single dashboard.

The visualization specification catalogs, including *Visual-FormatList*, *ChannelTypeList*, *FilterTypeList*, *VariableTypeList*, and *AggregateTypeList* are based on the Vega-lite visualization library. Vega-Lite provides a simple visualization grammar that simplifies the process of generating and deploying target dashboard source code [5]. The *Mark* class has a catalog of visual types based on various visual formats implemented in Vega-Lite, such as bars, lines, and circle charts. The *Encoding* class is an instantiation of the visual properties of the marks through the channel class. Every mark class contains 2 to 4 encodings, given that each mark should have a minimum of two encoding channels, such as x and y, and a maximum of four encoding channels, including color and size. The *Channel* class defines the specifications of a channel, including the axis definitions, variable types, and variable aggregations and filterings.

Furthermore, the decision-making specification catalogs are based on a previous study in the supply chain VA domain [10]. Various decision areas within supply chain activities fulfill the user variability requirements of a supply chain VA system. For example, sales management decision areas can provide insights into customer behavior that are useful for marketing departments. Similarly, demand management decision areas can help anticipate changes in demand patterns in the manufacturing planning department. Similarly, network integration and visibility decision areas can be utilized to offer real-time information on the status of products and shipments throughout the supply chain for the logistics department. A VA goal-type list defines various types of VA goals, such as identifying underlying patterns and exploring dimensions, to ensure a goal-oriented analysis. A VA task-type list provides various visualization tasks to assist with the exploratory analysis of unknown VA tasks. The decision-making catalog presented in [10] provides different

goalTypes based on the decision area, and different taskTypes based on the goalType. Based on this, the four lists regarding the decision-making process and data collection, including *DecisionAreaList*, *VAGoalTypeList*, *VATaskTypeList*, and *DataTypeList*, are created. The authors in [10] conducted a study regarding the use of VA in supply chain analytics and provided different VA goals and tasks based on different decision areas of the supply chain environments.

These catalogs are not exhaustive and can be expanded. For the purpose of the prototype, we included a representative set of options only.

B. GRAPHICAL MODELING WORKBENCH

To simplify the modeling activity and enhance the interaction with the metamodel, we utilized Eclipse Sirius [37] and create a graphical DSL. Sirius is an open-source platform used to create a modeling workbench through which users can generate various VA models based on the metamodel. Fig. 5 shows the workbench. The workbench can be used to interact with the DSML to create a model. The workbench has four sections, numbered in Fig. 4, and is explained as follows.

- 1) The canvas allows users to easily drag and drop model entities from a palette located on the right side of the workbench. The palette also contained different sections for the tools.
- 2) A section for creating dashboards where users can define the visualization frames, select a specific chart type, specify corresponding encodings, and select data sources.
- 3) The decision support tool offers a method for defining the decision-making process. It allows users to choose a decision area based on which the tool suggests a visualization goal and provides guidance for the visualization task. The tool uses a conditional branching logic based on the decision-making catalog to show the relevant goalTypes based on the user selection of decision area, and the corresponding taskTypes based on the selected goalType. The choices can be made in

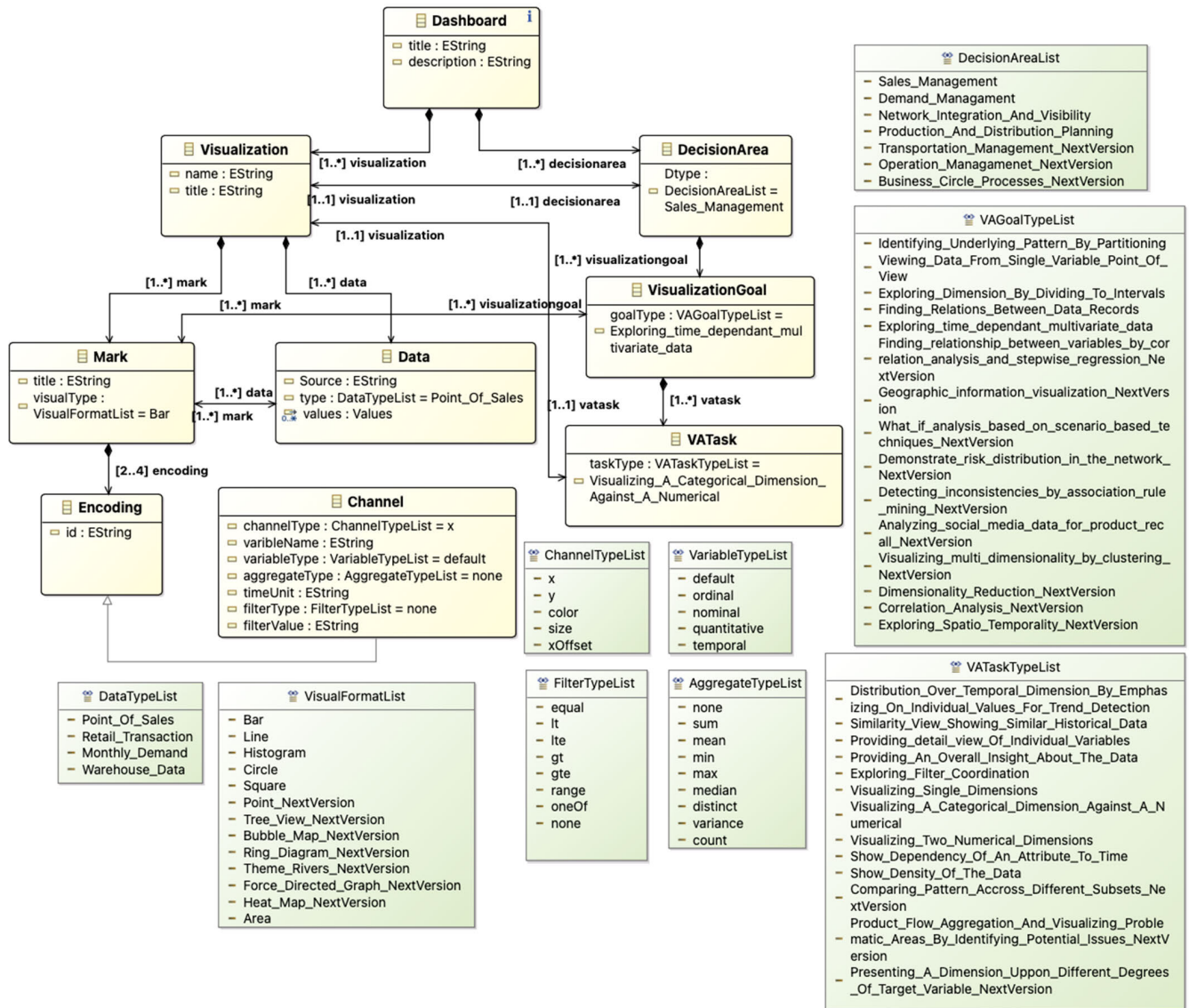


FIGURE 4. Supply chain VA metamodel.

the property panel located under the canvas that will pop-up immediately after an element is created.

- 4) Finally, a link tool helps specify the relationship between different model entities, such as the connection between a data source and a chart.

The tool provides a validation feature, where the user can check the conformance of the created model with the metamodel, and a small red cross mark by each of the model elements shows the errors. Additionally, there are well-formedness constraints that do not allow the user to create wrong models. For example, if the user wants to connect a database to a visualization frame instead of its corresponding visual encoding, the tool does not allow it.

Additionally, a brief description of each tool element is provided along with the tool in the documentation section of the workbench. The user sees the description of each

component by hovering the mouse over their notations, which guides the user to work with the tool.

C. MODEL TO TEXT TRANSFORMER

After creating the model, users can simply right-click on the model and choose the “Generate Dashboard Code” button to create an HTML file containing a Vega-Lite specification for the target dashboard source code. The file can then be opened and viewed on a browser. Vega-Lite provides a high-level grammar that enables a specification for interactive data visualization. Vega-Lite grammar includes a unit specification that outlines an individual Cartesian plot, encompassing a corresponding dataset, a specified mark type, and a collection of one or more encoding definitions for visual channels like position (x, y), color, size, etc. [5].

To create the automatic code generation, a code template is created using the Eclipse Acceleo [38] code generator to con-

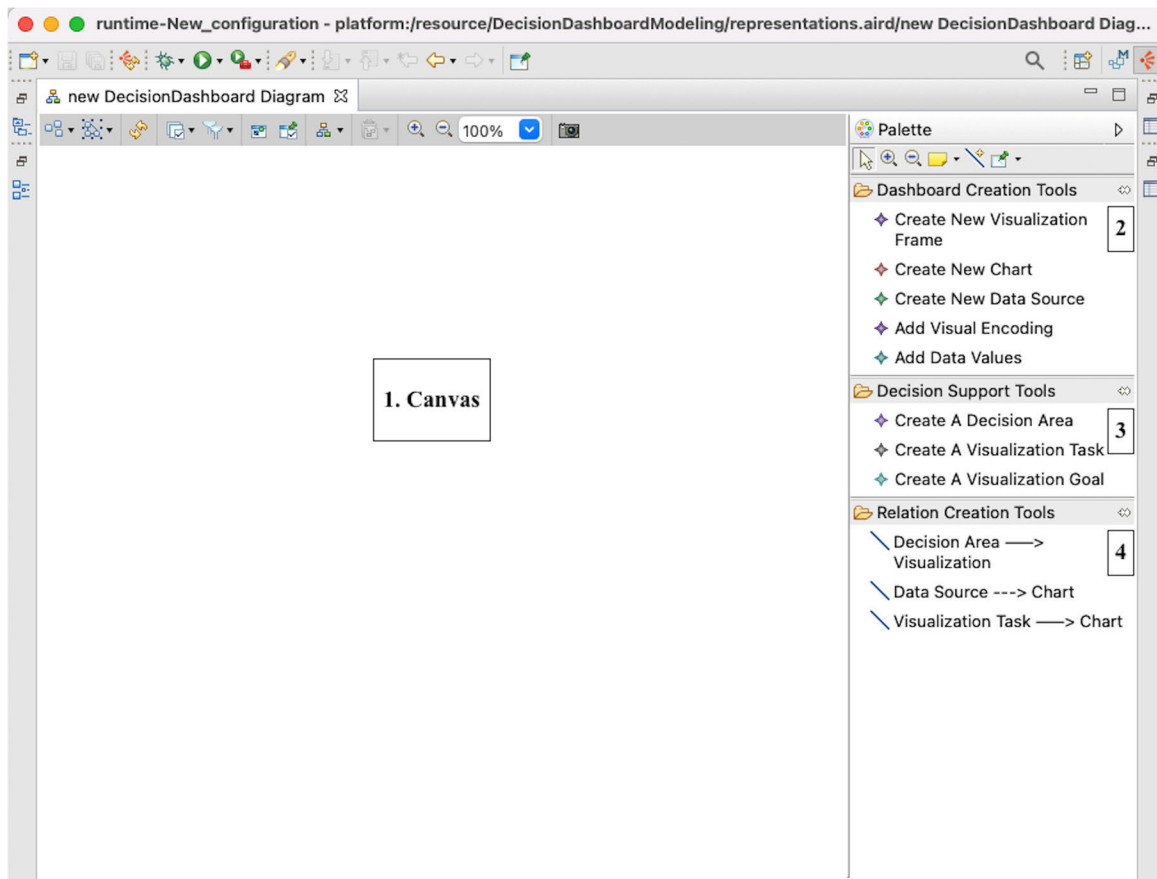


FIGURE 5. Sirius workbench for the graphical domain-specific language.

```

1  [template public model2JSONSpecification(c:Visualization, M:Mark)]
2
3  "title": "[M.title/]"[' /]
4  "data": {
5  "url": "[M.data.Source/]"
6  } [' /]
7  "mark": {"type": "[M.visualType.toString().toLowerCase/]"[' /]
8  "tooltip": true}[' /]
9      "encoding": {
10         [for (E : channelType | M.encoding)separator (',')]
11             "[E.channelType/]": {"timeUnit": "[E.timeUnit/]"[' /]
12         "aggregate": "[E.aggregateType/]"[' /] "field": "[E.variableName/]"[' /]
13         "type": "[E.variableType/]"
14         [/for]
15 [/template]

```

FIGURE 6. Main function of the model to JSON code template.

vert the model into source code. Aceleo is an open-source code generator development framework from the Eclipse Foundation that utilizes Object Constraint Language (OCL) to generate code templates. Fig. 6 shows a part of the code template function that generated visual marks JSON specification for the dashboard.

D. DEPLOYER

The generated code is an HTML file containing the Vega-Lite [5] JSON specification. Vega-Lite has a library called Vega-Embed, which makes it extremely simple to load Vega-Lite specifications into web-pages. For our prototype, we utilized this library to display our output dashboard on

TABLE 1. Aggregated data attributes from different datasets.

Dataset	Attribute	Description
Product Information data	ProductID*	Product Identification number
	ProductUnitPrice	Unit price for the product
	ShopProfileName	Name of the retail chain
Retail store Information data	StoreName*	Name of the Store
	PostCode	Postal Code
	AvgTurnover	Average turnover per year
	StoreName*	Name of the Store
Point-of-Sales data	MunicipalityNo	National municipality identification number
	CountyNo	National county identification number
	Date	Date of sale
	ProductID*	Product identification Number
	TotalSold	Number of total sale for each product

“*” denotes the join key attributes used to merge different tables.

a web browser. The Sirius Workbench also has a built-in browser that can be used to view the dashboard alongside the created model.

V. DEMONSTRATION OF MODELVIZ: A CASE STUDY

In order to demonstrate the working and feasibility of ModelViz, we design a couple of case studies with a real-world company and the data from its business processes. The intended users of ModelViz for the following case studies are the domain expert analysts from the supply chain environment who have the knowledge of their domain data attributes and want to create VA for their decision support applications. In what follows, we refer to them simply as “analysts.”

A. CASE STUDY DESIGN AND DATA

For the case study, we utilized the ModelViz prototype to provide decision support for a confectionery production company based in Norway. The company is part of a food supply chain that has collaboration with a number of retail stores across Norway. One of their most important datasets is Point-of-Sales (PoS) data that are collected from their partner retail chains. Additionally, they store their monthly sales to different stores along with information about each store in their business information system. We aggregated, cleaned and preprocessed these datasets in order to have one structured tabular data to use for visualization. The final dataset consist of more than 1.4 million rows corresponding to 3 years of product sales. Table 1 presents the aggregated attributes from different datasets along with their descriptions. To preserve confidentiality, we anonymized the name of the company and the partner retail stores from the data.

The case study consisted of two cases that were considered to demonstrate the feasibility of ModelViz. First, a case regarding sales management supports decisions related to

sales, for example, during negotiations between a company’s sales representatives and retail stores. Second is a production planning case that supports production decisions, such as prioritizing production orders based on customer demand volume. In the following section, we describe both cases in detail.

B. CASE 1: SALES MANAGEMENT

The company is interested in analyzing the data to understand its sales trends in different locations and periods. For this purpose, we used the prototype to design a dashboard to visualize sales data and conduct sales analytics to investigate 1) sales by store and 2) sales by region.

- 1) *Sales by retail store*: Sales by store refers to the number of sales made by each retail store that sells the company’s product. Obtaining insights into this metric facilitates distinguishing key retailers with the highest sales within the market and can be used in negotiations with stores. In Norway, the company sells its products through various retail chains. The analyst must know the sales figures for each retail chain’s product within a specific time frame. This information can be visualized in different time units. In this regard, we aim to visualize the data for retail sales performance on a monthly and weekday basis.
- 2) *Sales by region*: Sales by region indicate the number of sales in various geographical locations. The data include sales at both municipality and county levels. These analytics can help the company determine where its product is selling best and plan its marketing efforts accordingly. The analyst uses county and municipality identification numbers to investigate the performance of the company in different counties and municipalities.

For this purpose, the user interacts with the designed meta-model and instantiates it with the help of the modeling workbench to create a dashboard model. Table 2 presents the instantiation of the metamodel for each case.

Based on these requirements, as depicted in Fig. 7, the decision area selected for the analysis is sales management (The green box). Next, the user must identify the goal of the analysis. Since the attempt is to analyze the correlation between the total amount of sales and various variables in the dataset (The green data source icon), among the various options provided in the properties section of the tool, user selects the VA goal (The blue box) of “*Finding the relation between data records.*” Number of sales is a numerical variable, whereas retail chain names are categorical variables. Therefore, the appropriate VA task (The red box) would be “*Visualizing a categorical dimension against a numerical dimension.*” Based on these selections, the tool suggests three different chart types: “*bar,*” “*line,*” and “*force-directed.*” The user decides to utilize stacked bar charts for sales by store per month and a multi-series colored line chart to show the changes in sales during a weekday for various stores (The chart icons). Moreover, two stacked bar charts are used to

TABLE 2. Instantiation of the metamodel for the case study.

#	Domain Goal	Visualization	Mark	channelType: variableName
1	Sales by store	Monthly sales by store	Stacked Bar Chart	X: Month, Y: Total Sale Color: Shop Profile Name
		Weekday sales by store	Multi-series Colored Line Chart	X: Week day, Y: Total Sale Color: Shop Profile Name
2	Sales by region	Monthly sales by county	Filtered Stacked Bar Chart	X: Month, Y: Total Sale above 1000 units Color: County No.
		Monthly sales by municipality	Filtered Stacked Bar Chart	X: Month, Y: Total Sale above 500 units Color: Municipal No.

visualize sales in each municipality and county. The analyst decides to filter the data to show the sum of total sales above 500 units in each municipality and 1000 units in each county to identify significant sales locations. The last column in Table 2 displays the channels (The yellow boxes) used for various dataset variables.

The model depicted in Fig. 7 was created based on the above instantiation. The model serves as a documentation of the entire design process. Once the model is created, we generate the target dashboard source code using the code generator command found in the menu items of the tool. The generated code can then be opened in a browser that displays the output dashboard. Fig. 8 shows the dashboard generated for the sales analytics model depicted in Fig. 7.

C. CASE 2: PRODUCTION PLANNING

As mentioned, the company sells its products through retail stores in Norway. Three different analyses were performed for this purpose. 1) top-selling customer segment, 2) individual product performance, and 3) correlation between sales and product unit prices.

- 1) *Top-selling customer segment*: The company aims to analyze the demands of its top-selling customer segment and plan its production accordingly. The goal of the analyst is to analyze sales data by customer segment over time. This is known as time-dependent multivariate analysis and involves investigating the relationship between multiple variables that change

over time. Additionally, identifying weekdays with the highest sales for high-selling products helps the production department plan their daily production.

- 2) *Individual product performance*: Individual product performance can indicate best-selling products and prioritize resource allocation for production. The analyst uses the filter feature of the tool to identify products with sales above a certain limit, and demonstrates their sales performance.
- 3) *Correlation between sales and product unit price*: The goal is to identify whether there is a correlation between total sales and the product unit price, such as products with higher unit prices that have higher total sales. For this purpose, the analyst uses the aggregate feature of the tool to calculate the average unit price for each product and observe the total sales of such products.

We want to demonstrate how ModelViz can be used to design a dashboard for these objectives. The process starts by selecting the decision area and obtaining suggestions for visualization goals and tasks. In addition, suitable chart types are suggested to the analysts. Finally, the target dashboard source code is created automatically and can be opened in a browser.

Following the Fig. 10 (Analysis 1), to identify the top-selling customer segment, analyst:

- 1) Creates an instance of a decision area (The green box) and selects “production planning and distribution” as the decision area that needs to be supported.
- 2) Uses the toolbox to add an instance of the visualization goal (The blue box) to the decision area and selects “Exploring time-dependent multivariate data” in the properties panel.
- 3) Within the decision-making process, adds a visualization task (The red box) that suggests “showing the dependency of an attribute on time.”
- 4) In the next stage, the analyst creates a new visualization frame (The orange box).
- 5) Adds the data source (The green data source icon) and selects the sales data as the data source.
- 6) Associate it with the link to ingest the data (The dotted arrow).
- 7) Eventually, a new chart is added to the frame (The chart icon). The tool suggests choosing from a variety of potentially useful chart types. One way to visualize time-dependent multivariate data is through stacked area charts. This type of chart allows for the visualization of total sales over time, broken down by different customer segments. The analyst then selects a stacked area chart. In addition, the analyst creates a stacked bar chart to show weekdays with the highest sales for product sales higher than 2000 units per weekday.
- 8) Adds necessary channel encodings (The yellow boxes). For the stacked area chart, the x-axis representing “month,” the y-axis representing “total sales,” and different retail store names represented by color

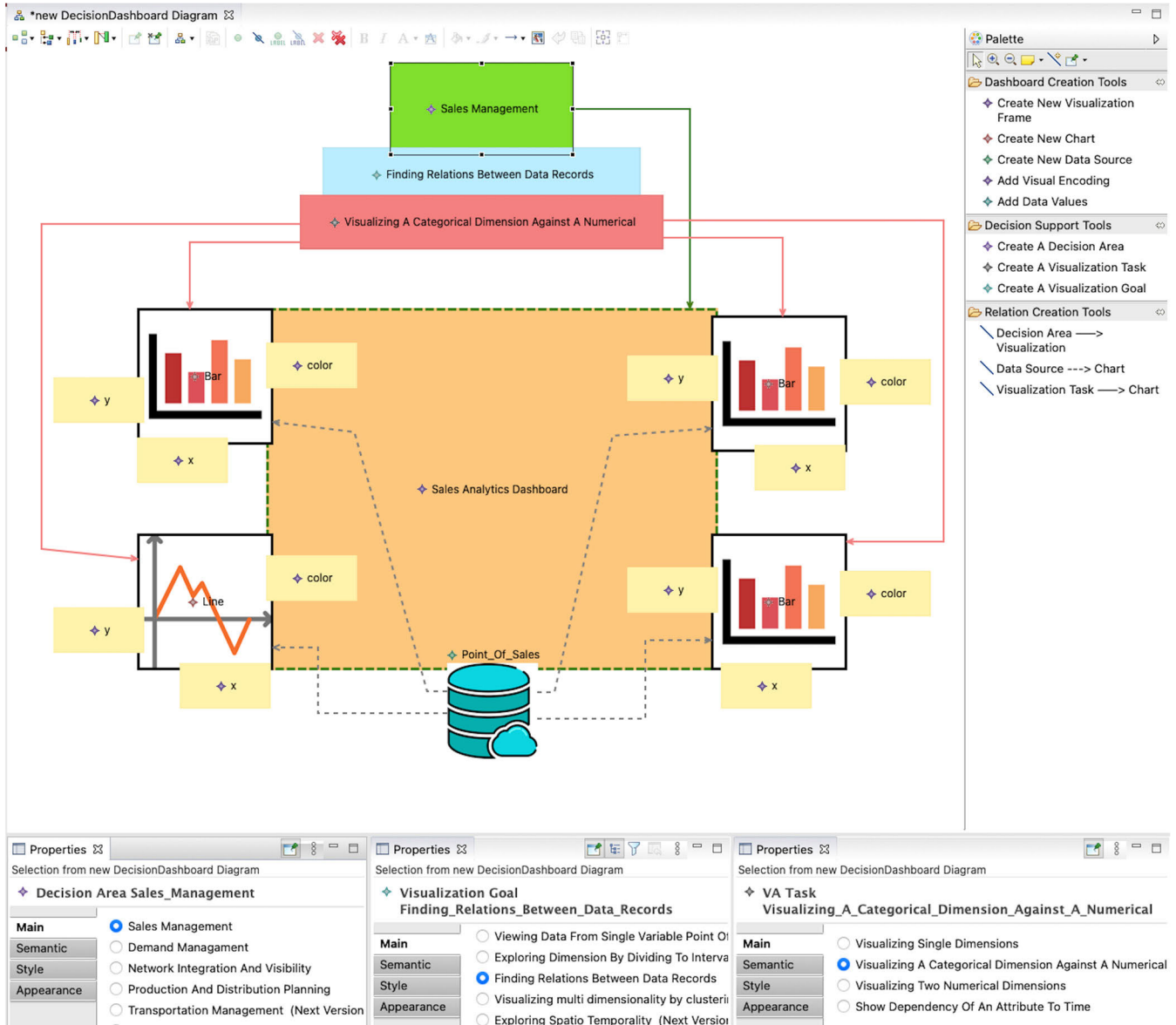


FIGURE 7. Sales VA dashboard model within the workbench.

encoding will be added. For the bar chart, the x-axis represents weekdays, the y-axis represents the total sales of products selling higher than 2000 units, and the color encoding will be different product identification numbers.

Following Fig. 10 (Analysis 2), the steps to observe the individual product performance will be as follows:

- 1) Creating a new instance of a decision area (The green Box) and selecting “*production planning and distribution*” as the decision area that needs to be supported.
- 2) Choosing “*Finding relations between data records*” for the visualization goal (The blue box).
- 3) Adding a visualization task (The red box) that suggests “*Visualizing a categorical dimension against a numerical.*”

- 4) Adding a new chart (The chart icon) to the frame and selecting the bar charts based on the suggested charts.
- 5) Associating the chart with a link (The dotted arrow) to the data source to ingest the data.
- 6) Adding the necessary channel encodings (The yellow boxes). The x-axis represents “*ProductID*” and the y-axis represents “*total sales.*”

Finally, following Fig. 10 (Analysis 3), to identify the correlation between sales and product unit price the analyst,

- 1) Creates a new instance of a decision area (The green Box) and selects “*production planning and distribution*” as the decision area that needs to be supported.
- 2) Chooses “*Finding relations between data records*” for the visualization goal (The blue box).
- 3) Adds a visualization task (The red box) that suggests “*Visualizing two numerical dimensions.*”

Sales Analytics Dashboard

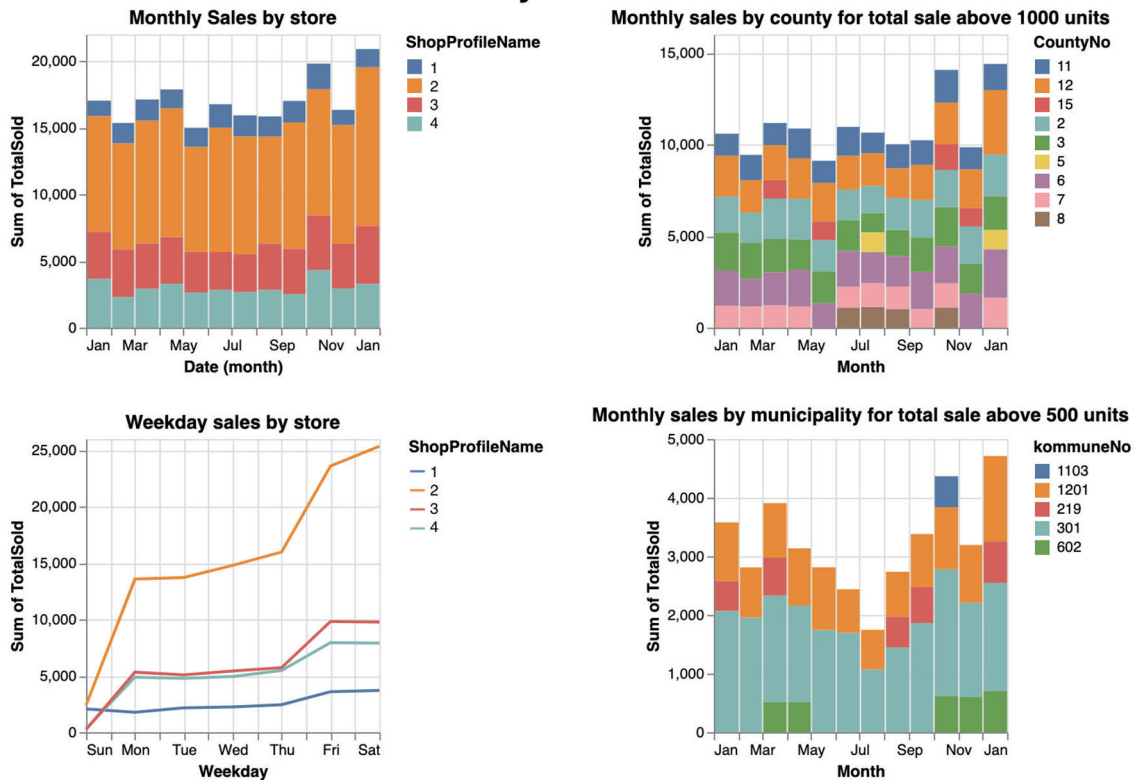


FIGURE 8. Generated sales management dashboard from the model in Fig. 7.

- 4) Adds a new chart (The chart icon) to the frame, and selects the circle chart from the suggested charts.
- 5) Associates the chart with a link (The dotted line) to the data source.
- 6) Adds the necessary channel encodings (The yellow boxes). The x-axis represents the “sum of total sales” for the products with sales higher than 5000 units, the y-axis represents “average product unit price,” and the color encoding represents different products.

Fig. 9 shows the final model created as a result of these steps in a screenshot of ModelViz. Finally, the analyst simply right-clicks on the model and chooses the generate code option that produces the visualization and can be opened in the browser, as shown in Fig. 10.

Overall, ModelViz supported the decision-making process in the VA. As mentioned earlier, the aim was to support the user in the analytical process and to record this process for future reference. Next, we discuss how this approach satisfies the desired requirements.

VI. EVALUATION OF MODELVIZ

This study aimed to demonstrate the feasibility of the framework. Therefore, we developed a prototype to validate the framework by verifying whether the proposal meets the desired requirements. Prototyping is a well-known method for representing a system and validating it according to

predefined requirements. Therefore, we begin by discussing the extent to which these requirements are met. Moreover, we conducted a comparative analysis of the developed prototype with some similar research prototypes from the literature.

A. DISCUSSION OVER REQUIREMENTS VALIDATION

At the beginning of this study, we identified four requirements that must be satisfied for the development of a VA system. Those requirements formed the basis for the three views of the VA development framework presented in [29] and Fig. 1. In order to support the three views, we proposed and developed a graphical DSML for designing VA systems using the metamodeling technique based on the requirements. Therefore, the metamodel consists of three main classes: 1) a visualization class that handles the visual encodings along with their relevant descriptions, such as marks and channels; 2) a decision class that supports the integration of the decision-making process into the VA activity, which consists of decision areas, VA goals, and VA tasks; and 3) a data class that handles data ingestion. We also conducted a case study with a confectionery manufacturing company. This case study demonstrates how sales and demand data can be utilized to create VA dashboards for sales management and production planning. The proposed system satisfies the design requirements by offering the following capabilities:

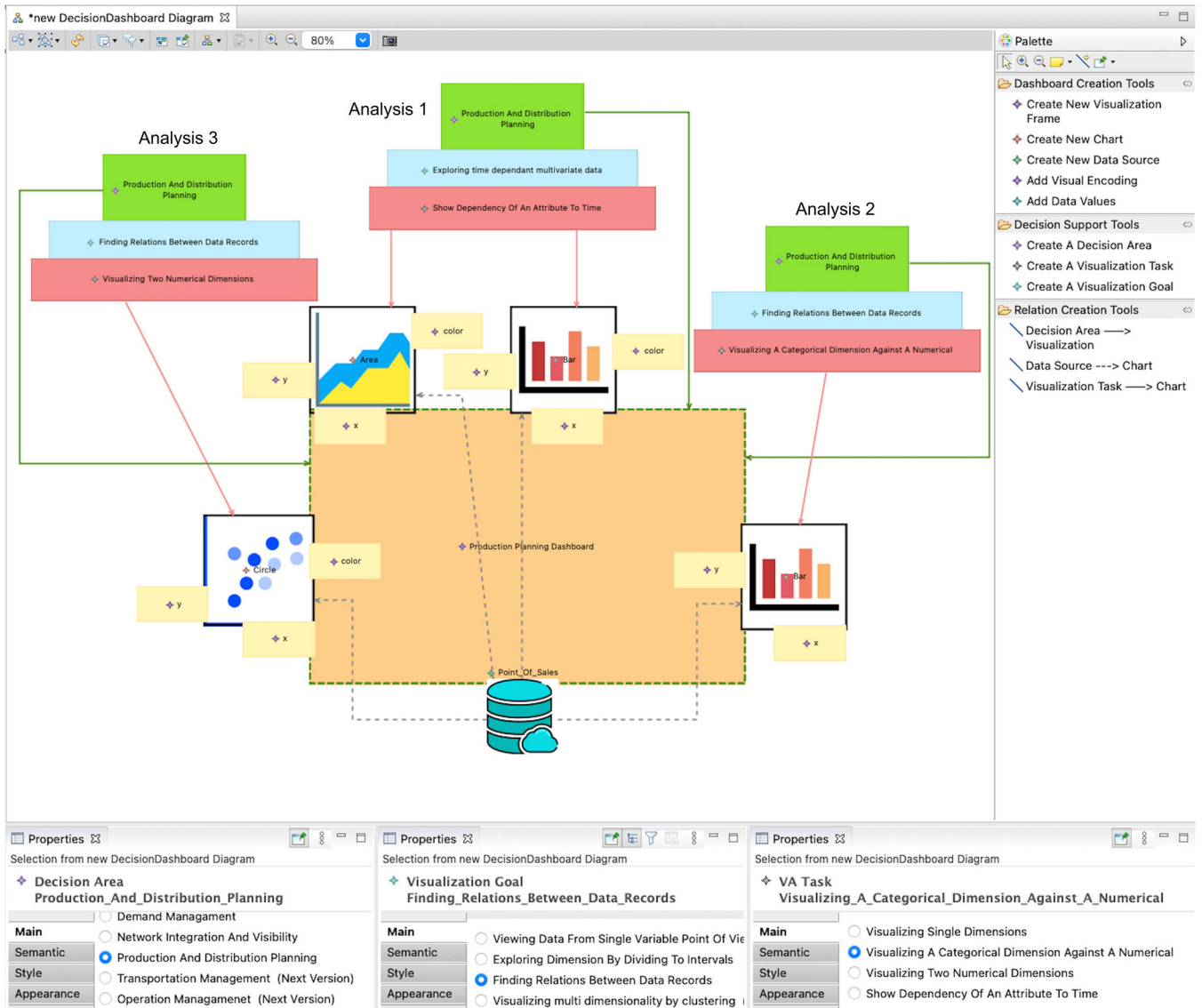


FIGURE 9. Product planning model in ModelViz.

- 1) *Streamlining exploratory analysis*: The system aids users in exploratory analysis by guiding them in choosing the right VA goals and tasks based on their desired decision area. The process of making visualizations can then be recorded for future traceability and reproducibility.
- 2) *Consolidating data from scattered sources*: Users can ingest data from different sources for each of the visuals separately which ensures data availability for different analytics tasks.
- 3) *Assisting with visual encodings*: The generation of an optimal visualization dashboard occurs automatically based on the user's goals and tasks, ensuring that users can create effective visualizations quickly and easily. This feature streamlines the generation of dashboards by eliminating the need for users to write code or manually create visualizations.
- 4) *Integration with decision-making activities*: The decision-making process is integrated into the VA process, ensuring that the data analytics is aligned with the business objectives.

To conclude, the case study with the confectionery manufacturing company demonstrates how the system meets these requirements. The company utilized sales and demand data to generate tailored visualizations that effectively conveyed insights, aiding sales management and production planning decisions. Additionally, the modeling approach has been shown to be feasible in defining decision areas, VA goals, and VA tasks, ensuring that VA activity is aligned with the company's business goals. Overall, the case study demonstrates how the system's capabilities can be applied to real-world supply chain applications, streamlining the exploratory analysis process, and enabling informed decision-making.

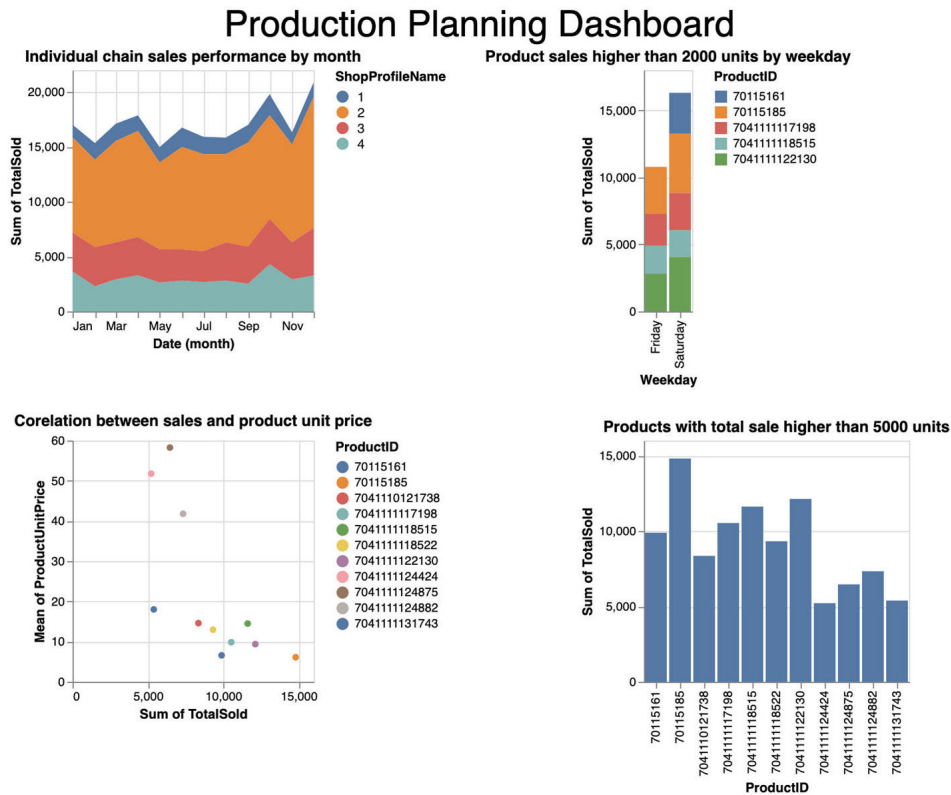


FIGURE 10. Generated production planning dashboard from the model in Fig. 9.

B. QUANTITATIVE COMPARATIVE EVALUATION

ModelViz was developed using a DSL based on the meta-modeling technique. To be efficient and useful, DSLs should be comprehensive and correlated with the corresponding domains. In this context, Guizzardi et al. [39] proposed a framework for evaluating the design of modeling languages by comparing their metamodel components with another reference language. The authors proposed four metrics for comparison of the metamodels. They argued that DSLs could be compared by their metamodels to check their truthfulness to the domain, as well as their pragmatic efficiency to support communication, understanding, and reasoning in the domain. The four metrics are defined as follows:

- 1) *Lucidity*: A modeling language is lucid compared with a reference language if and only if there is a mapping between each concept of the model and at most one of the concepts within the other language. Lucidity was used to evaluate the clarity of the modeling language. Lucidity can be calculated as the number of concepts from the proposed DSL with only one equivalent concept in the reference language divided by the total number of concepts from the proposed DSL.
- 2) *Soundness*: A modeling language is sound compared with a reference language if and only if there is a mapping between every concept of the model and at least one of the concepts within the other language. Soundness checks the degree of correlation between the

modeling language and the reference model. Soundness can be calculated as the number of concepts from the proposed DSL with at least one equivalent concept in the reference language divided by the total number of concepts from the proposed DSL.

- 3) *Laconicity*: A modeling language is laconic compared with a reference language if and only if there is a mapping between each concept of the reference model and at most one of the concepts of the modeling language. Laconicity checks how concise or redundant concepts are defined in the modeling language. Laconicity can be calculated as the number of concepts from the reference DSL with only one equivalent concept in the proposed language divided by the total number of concepts from the reference DSL.
- 4) *Completeness*: A modeling language is complete compared to a reference language if and only if there is a mapping between each concept of the reference language and at least one of the concepts of the modeling language. Completeness validates the expressiveness of the modeling language. Completeness can be calculated as the number of concepts from the reference DSL with at least one equivalent concept in the proposed language divided by the total number of concepts from the reference DSL.

From the literature, there are three similar modeling languages, namely MetaViz [40], ReqViz [9], and CitiesBoard [20], which can be used as reference languages. All

TABLE 3. Results of the quantitative comparison.

Metrics	MetaViz	ReqViz	Cities-Board
Lucidity	5/9 (56%)	6/9 (67%)	7/9 (78%)
Soundness	8/9 (89%)	9/9 (100%)	5/9 (56%)
Laconicity	25/25 (100%)	29/30 (97%)	10/10 (100%)
Completeness	13/25 (52%)	14/30 (47%)	5/10 (50%)

three are modeling languages for VA, although they have different application domains. MetaViz is a generic modeling language used to create visualization dashboards. ReqViz is a modeling language for requirement-driven visualization, and the Cities-board is a modeling language for creating visualization dashboards for smart cities. In order to calculate the corresponding values for the metrics, we have reviewed all three DSLs along with the ModelViz and identified all of the concepts defined in each. The concepts were listed in an Excel document, and the values were calculated by counting the concepts based on the provided formulas for each metric definition. We first considered the main classes of the metamodels, and if a main class was not present, we also excluded all child classes. Table 3 presents the results of the analysis.

Although the proposal showed high soundness and laconicity, its lucidity and completeness were relatively low. The reason for this is the difference between the domains of the modeling languages. While MetaViz and ReqViz are used for generic VA, ModelViz is a prototype for the supply chain domain. Therefore, there is less focus on visualization-specific concepts, such as the positioning of graphs on the screen or legends of axes.

VII. CONCLUSION AND FUTURE WORK

Data-driven decision-making involves the challenging task of conducting an exploratory visual analysis of the data generated and collected from various sources. For example, supply chain management encompasses a range of business activities and decision-making processes throughout the various stages of the chain, including procurement, manufacturing, warehousing, distribution, logistics, and retail. The amount of data generated and collected at each stage supports data-driven decision-making. The diversity of business activities within the supply chain makes VA a challenging task. To effectively support this task, VA systems should satisfy specific requirements. These requirements include the ability to conduct exploratory analysis, consolidate data from disparate sources, create visual encodings that suit different users, and integrate seamlessly with decision-making activities to achieve business objectives.

In this study, we proposed a graphical DSML, named ModelViz, to address the outlined requirements. Within the software engineering discipline, DSMLs utilize abstract domain concepts to develop software systems. We harness this potential to utilize supply chain domain concepts in the

development of VA systems. In this way, abstract domain aspects simplify the design of a VA that aligns with business objectives and can be used by different users across a supply chain. Simultaneously, simplifying the creation of visual encodings for non-programmer domain experts can be achieved by treating the generation of VA dashboards as a modeling activity. Moreover, consolidation of data can be integrated into the VA design process by incorporating it as a component of the model.

In this regard, a metamodel for VA development includes classes related to both automated dashboard creation and decision-making. The metamodel is then used as an abstract syntax for DSML. Furthermore, a graphical workbench tool simplifies the interaction with the metamodel and facilitates the creation of dashboards through graphical modeling. We utilized various Eclipse software tools, including Eclore, Sirius, and Acceleo, at different stages of the process, which facilitated the implementation of DSML. Consequently, users can use the metamodel concept to generate VA dashboards.

To present our proposal better, we conducted a case study involving the development of a dashboard for sales management and production planning in a confectionery production company. The focal company is part of a Norwegian fast-moving consumer goods supply chain and owns the necessary data for our analysis. As a result, the tool simplifies the interaction with the metamodel and uses various domain concepts to create a goal-oriented VA dashboard.

Finally, after validating ModelViz for the desired requirements, we evaluated the proposed DSML by comparing it with three similar modeling languages from the literature. These evaluation results can be used to improve ModelViz in later versions. Based on the evaluation results presented in Section VI-B., although the proposed DSL demonstrates *soundness* and *laconicity*, it requires improvements regarding its completeness and lucidity. We argued that since ModelViz is a prototype for the supply chain domain, we had less focus on the design of visual encodings. Therefore, future versions of ModelViz can be improved in terms of these concepts, such as the positioning of graphs on the screen or customizing the legends based on user needs. Additionally, to improve ModelViz in terms of laconicity or the clarity of the DSL, we can create new classes from the attributes of some of the defined classes. For example, the attribute “filterType” from the “channelType” class can be separated and form a new class. This will add to the clarity of the ModelViz in later versions.

For future work, we plan to conduct a validation study with the help of experts to verify the applicability and validity of both the metamodel and tool. The tool was developed as a research prototype and may not include all the necessary functionalities for usability testing. However, validating the metamodel helps to verify the coherence between the metamodel and the graphical tool. Additionally, future work can include a long-term user study to explore user experience and performance evaluation of the method.

Moreover, although the objective of the ModelViz prototype was to demonstrate the applicability of the proposed framework, future work can explore adding new features regarding its analytics capabilities. In this regard, machine learning based predictive analytics can provide the support for advanced analytics.

ModelViz was developed based on a domain-specific approach focused on the supply chain domain. Nevertheless, the approach can be used for different application domains as an area for future work to examine its generalizability. Finally, the system should be employed in a real-world setting to inspect its ability to integrate with other business intelligence systems. We believe that this method can be further developed into a fully functional application that will be useful for supply chain partners, facilitating their decision-making activities with the help of VA.

REFERENCES

- [1] M. Bostock and J. Heer, "Protovis: A graphical toolkit for visualization," *IEEE Trans. Vis. Comput. Graphics*, vol. 15, no. 6, pp. 1121–1128, Nov. 2009, doi: [10.1109/TVCG.2009.174](https://doi.org/10.1109/TVCG.2009.174).
- [2] J. Heer, S. K. Card, and J. A. Landay, "Prefuse: A toolkit for interactive information visualization," in *Proc. SIGCHI Conf. Human Factors Comput. Syst.* New York, NY, USA: Association for Computing Machinery, Apr. 2005, pp. 421–430, doi: [10.1145/1054972.1055031](https://doi.org/10.1145/1054972.1055031).
- [3] L. Nair, S. Shetty, and S. Shetty, "Interactive visual analytics on big data: Tableau vs D3.js," *J. e-Learn. Knowl. Soc.*, vol. 12, no. 4, pp. 139–150, Sep. 2016. Accessed: Feb. 16, 2020. [Online]. Available: <https://www.learnlib.org/p/173675/>
- [4] D. Orlando, J. Ormachea, V. Soliani, and A. A. Vaisman, "TGV: A visualization tool for temporal property graph databases," *Inf. Syst. Frontiers*, pp. 1–22, Aug. 2023, doi: [10.1007/s10796-023-10426-1](https://doi.org/10.1007/s10796-023-10426-1).
- [5] A. Satyanarayan, D. Moritz, K. Wongsuphasawat, and J. Heer, "Vega-lite: A grammar of interactive graphics," *IEEE Trans. Vis. Comput. Graphics*, vol. 23, no. 1, pp. 341–350, Jan. 2017, doi: [10.1109/TVCG.2016.2599030](https://doi.org/10.1109/TVCG.2016.2599030).
- [6] R. Morgan, G. Grossmann, and M. Stumptner, "VizDSL: Towards a graphical visualisation language for enterprise systems interoperability," in *Proc. Int. Symp. Big Data Vis. Analytics (BDVA)*, Nov. 2017, pp. 1–8, doi: [10.1109/BDVA.2017.8114629](https://doi.org/10.1109/BDVA.2017.8114629).
- [7] O. Marjanovic, G. Patmore, and N. Balnave, "Visual analytics: Transferring, translating and transforming knowledge from analytics experts to non-technical domain experts in multidisciplinary teams," *Inf. Syst. Frontiers*, vol. 25, no. 4, pp. 1571–1588, Aug. 2023, doi: [10.1007/s10796-022-10310-4](https://doi.org/10.1007/s10796-022-10310-4).
- [8] S. Nalchigar and E. Yu, "Business-driven data analytics: A conceptual modeling framework," *Data Knowl. Eng.*, vol. 117, pp. 359–372, Sep. 2018, doi: [10.1016/j.datak.2018.04.006](https://doi.org/10.1016/j.datak.2018.04.006).
- [9] A. Lavallo, A. Maté, and J. Trujillo, "Requirements-driven visualizations for big data analytics: A model-driven approach," in *Conceptual Modeling (Lecture Notes in Computer Science)*, A. H. F. Laender, B. Pernici, E.-P. Lim, and J. P. M. de Oliveira, Eds. Salvador, Brazil: Springer, 2019, pp. 78–92, doi: [10.1007/978-3-030-33223-5_8](https://doi.org/10.1007/978-3-030-33223-5_8).
- [10] A. Khakpour, R. Colomo-Palacios, and A. Martini, "Visual analytics for decision support: A supply chain perspective," *IEEE Access*, vol. 9, pp. 81326–81344, 2021, doi: [10.1109/ACCESS.2021.3085496](https://doi.org/10.1109/ACCESS.2021.3085496).
- [11] S. Alsbaugh, N. Zokaie, A. Liu, C. Jin, and M. A. Hearst, "Futzing and moseying: Interviews with professional data analysts on exploration practices," *IEEE Trans. Vis. Comput. Graphics*, vol. 25, no. 1, pp. 22–31, Jan. 2019, doi: [10.1109/TVCG.2018.2865040](https://doi.org/10.1109/TVCG.2018.2865040).
- [12] S. Carpendale, "Evaluating information visualizations," in *Information Visualization: Human-Centered Issues and Perspectives (Lecture Notes in Computer Science)*, A. Kerren, J. T. Stasko, J.-D. Fekete, and C. North, Eds. Berlin, Germany: Springer, 2008, pp. 19–45, doi: [10.1007/978-3-540-70956-5_2](https://doi.org/10.1007/978-3-540-70956-5_2).
- [13] L. G. Nonato and M. Aupetit, "Multidimensional projection for visual analytics: Linking techniques with distortions, tasks, and layout enrichment," *IEEE Trans. Vis. Comput. Graphics*, vol. 25, no. 8, pp. 2650–2673, Aug. 2019, doi: [10.1109/TVCG.2018.2846735](https://doi.org/10.1109/TVCG.2018.2846735).
- [14] S. Ko, R. Maciejewski, Y. Jang, and D. S. Ebert, "MarketAnalyzer: An interactive visual analytics system for analyzing competitive advantage using point of sale data," *Comput. Graph. Forum*, vol. 31, no. 3, pp. 1245–1254, Jun. 2012, doi: [10.1111/j.1467-8659.2012.03117.x](https://doi.org/10.1111/j.1467-8659.2012.03117.x).
- [15] M. Wasesa, M. Mashuri, P. Handayani, and U. S. Putro, "Open data visual analytics to support decisions on physical investments," *Proc. Comput. Sci.*, vol. 161, pp. 797–804, Jan. 2019, doi: [10.1016/j.procs.2019.11.185](https://doi.org/10.1016/j.procs.2019.11.185).
- [16] A. Vázquez-Ingelmo, F. J. García-Peñalvo, R. Therón, and M. Á. Conde, "Representing data visualization goals and tasks through meta-modeling to tailor information dashboards," *Appl. Sci.*, vol. 10, no. 7, p. 2306, Mar. 2020, doi: [10.3390/app10072306](https://doi.org/10.3390/app10072306).
- [17] A. González-Torres, F. J. García-Peñalvo, R. Therón-Sánchez, and R. Colomo-Palacios, "Knowledge discovery in software teams by means of evolutionary visual software analytics," *Sci. Comput. Program.*, vol. 121, pp. 55–74, Jun. 2016, doi: [10.1016/j.scico.2015.09.005](https://doi.org/10.1016/j.scico.2015.09.005).
- [18] I. Logre and A.-M. Déry-Pinna, "MDE in support of visualization systems design: A multi-staged approach tailored for multiple roles," *Proc. ACM Hum.-Comput. Interact.*, vol. 2, pp. 1–17, Jun. 2018, doi: [10.1145/3229096](https://doi.org/10.1145/3229096).
- [19] C. Pettit, S. N. Lieske, and M. Jamal, "CityDash: Visualising a changing city using open data," in *Planning Support Science for Smarter Urban Futures (Lecture Notes in Geoinformation and Cartography)*. Cham, Switzerland: Springer, 2017, pp. 337–353, doi: [10.1007/978-3-319-57819-4_19](https://doi.org/10.1007/978-3-319-57819-4_19).
- [20] E. Rojas, V. Bastidas, and C. Cabrera, "Cities-board: A framework to automate the development of smart cities dashboards," *IEEE Internet Things J.*, vol. 7, no. 10, pp. 10128–10136, Oct. 2020, doi: [10.1109/JIOT.2020.3002581](https://doi.org/10.1109/JIOT.2020.3002581).
- [21] I. Logre, S. Mosser, P. Collet, and M. Riveill, "Sensor data visualisation: A composition-based approach to support domain variability," in *Proc. 10th Eur. Conf. Model. Found. Appl.*, vol. 8569. Berlin, Germany: Springer-Verlag, Jul. 2014, pp. 101–116, doi: [10.1007/978-3-319-09195-2_7](https://doi.org/10.1007/978-3-319-09195-2_7).
- [22] H. Park, M. A. Bellamy, and R. C. Basole, "Visual analytics for supply network management: System design and evaluation," *Decis. Support Syst.*, vol. 91, pp. 89–102, Nov. 2016, doi: [10.1016/j.dss.2016.08.003](https://doi.org/10.1016/j.dss.2016.08.003).
- [23] Q. Li, Q. Q. Liu, C. F. Tang, Z. W. Li, S. C. Wei, X. R. Peng, M. H. Zheng, T. J. Chen, and Q. Yang, "Warehouse vis: A visual analytics approach to facilitating warehouse location selection for business districts," *Comput. Graph. Forum*, vol. 39, no. 3, pp. 483–495, Jun. 2020, doi: [10.1111/cgf.13996](https://doi.org/10.1111/cgf.13996).
- [24] M. P. Brundage, W. Z. Bernstein, K. C. Morris, and J. A. Horst, "Using graph-based visualizations to explore key performance indicator relationships for manufacturing production systems," *Proc. CIRP*, vol. 61, pp. 451–456, Jan. 2017, doi: [10.1016/j.procir.2016.11.176](https://doi.org/10.1016/j.procir.2016.11.176).
- [25] M. Adnan and R. A. Ruddle, "A set-based visual analytics approach to analyze retail data," in *Proc. EuroVis Workshop Visual Anal. (EuroVA)*, C. Tominski and T. von Landesberger, Eds. Eindhoven, The Netherlands: Eurographics Association, 2018, pp. 37–41, doi: [10.2312/eurova.20181110](https://doi.org/10.2312/eurova.20181110).
- [26] M. A. Teruel, A. Maté, E. Navarro, P. González, and J. C. Trujillo, "The new era of business intelligence applications: Building from a collaborative point of view," *Bus. Inf. Syst. Eng.*, vol. 61, no. 5, pp. 615–634, Oct. 2019, doi: [10.1007/s12599-019-00578-3](https://doi.org/10.1007/s12599-019-00578-3).
- [27] P. M. Booth, "Analytical behaviour: A study of decision-making facilitated by visual analytics," *Fac. Eng., Sci. Math., School Electron. Comput. Sci., Univ. Southampton, Southampton, U.K.*, 2019. Mar. 26, 2020. [Online]. Available: <https://eprints.soton.ac.uk/438599/>
- [28] T. Ruppert, "Visual analytics to support evidence-based decision making," PhD. dissertation, Dept. Faculty Comput. Sci., Technischen Universität Darmstadt, Darmstadt, Germany, 2018.
- [29] A. Khakpour, R. Colomo-Palacios, and A. Martini, "Towards a framework for developing visual analytics in supply chain environments," *Int. J. Inf. Syst. Project Manage.*, vol. 11, no. 1, pp. 52–71, Apr. 2023, doi: [10.12821/ijispm110103](https://doi.org/10.12821/ijispm110103).
- [30] C. Atkinson and T. Kuhne, "Model-driven development: A metamodeling foundation," *IEEE Softw.*, vol. 20, no. 5, pp. 36–41, Sep. 2003, doi: [10.1109/MS.2003.1231149](https://doi.org/10.1109/MS.2003.1231149).
- [31] N. Vasic, H.-G. Fill, R. A. Buchmann, and D. Karagiannis, "A domain-specific language for modeling method definition: From requirements to grammar," in *Proc. IEEE 9th Int. Conf. Res. Challenges Inf. Sci. (RCIS)*, May 2015, pp. 286–297, doi: [10.1109/RCIS.2015.7128889](https://doi.org/10.1109/RCIS.2015.7128889).

- [32] B. H. C. Cheng, B. Combemale, R. B. France, J.-M. Jézéquel, and B. Rumpe, "On the globalization of domain-specific languages," in *Globalizing Domain-Specific Languages: International Dagstuhl Seminar, Dagstuhl Castle, Germany, October 5–10, 2014, Revised Papers*, B. Combemale, B. H. C. Cheng, R. B. France, J.-M. Jézéquel, and B. Rumpe, Eds. Cham, Switzerland: Springer, 2015, pp. 1–6, doi: [10.1007/978-3-319-26172-0_1](https://doi.org/10.1007/978-3-319-26172-0_1).
- [33] A. Khakpour, R. Colomo-Palacios, A. Martini, and M. Sánchez-Gordón, "The use of domain-specific languages for visual analytics: A systematic literature review," *Technologies*, vol. 11, no. 2, p. 37, Mar. 2023, doi: [10.3390/technologies11020037](https://doi.org/10.3390/technologies11020037).
- [34] A. Khakpour. (Oct. 2023). *ModelViz*. Accessed: Mar. 16, 2024. [Online]. Available: <https://github.com/alirezakhpy/ModelViz>
- [35] R. Gronback. *Eclipse Modeling Project | The Eclipse Foundation*. Accessed: Aug. 25, 2023. [Online]. Available: <https://eclipse.dev/modeling/emf/>
- [36] M. Fowler, *UML Distilled: A Brief Guide to the Standard Object Modeling Language*. Reading, MA, USA: Addison-Wesley, 2004.
- [37] Eclipse Foundation. *The Easiest Way to Get Your Own Modeling Tool*. Accessed: Mar. 16, 2024. [Online]. Available: <https://eclipse.dev/sirius/>
- [38] Eclipse Foundation. *Generate Anything From any EMF Model*. Aceleo Home. Accessed: Mar. 16, 2024. [Online]. Available: <https://www.eclipse.org/aceleo/>
- [39] G. Guizzardi, L. F. Pires, and M. van Sinderen, "An ontology-based approach for evaluating the domain appropriateness and comprehensibility appropriateness of modeling languages," in *Model Driven Engineering Languages and Systems* (Lecture Notes in Computer Science), L. Briand and C. Williams, Eds. Berlin, Germany: Springer, 2005, pp. 691–705, doi: [10.1007/11557432_51](https://doi.org/10.1007/11557432_51).
- [40] A. Vázquez-Ingelmo, F. J. García-Peñalvo, and R. Therón, "MetaViz—A graphical meta-model instantiator for generating information dashboards and visualizations," *J. King Saud Univ. Comput. Inf. Sci.*, vol. 34, no. 10, pp. 9977–9990, Nov. 2022, doi: [10.1016/j.jksuci.2022.09.015](https://doi.org/10.1016/j.jksuci.2022.09.015).



ALIREZA KHAPOUR received the bachelor's degree in information systems and engineering from MSRIT, the first master's degree in software engineering from Azad University, and the second master's degree in applied computer science from Østfold University College. He is currently pursuing the Ph.D. degree with the University of Oslo, with a focus on visual analytics modeling for decision-support systems.

He is a Ph.D. Research Fellow with Østfold University College. As a part of the Ph.D. studies, he is working on a project called DigiMat that aims to make the Norwegian food supply chain smart, transparent, and sustainable. His main research interests include machine learning, visual analytics, and information systems.



ANDREA VÁZQUEZ-INGELMO received the bachelor's, master's, and Ph.D. degrees in computer science from the University of Salamanca (USAL), Spain, in 2016, 2018, and 2022, respectively.

Since 2016, she has been a member of the Research Group of Interaction and eLearning (GRIAL), USAL. Her research interests include human-computer interaction, software engineering, data visualization, and machine learning applications.



RICARDO COLOMO-PALACIOS received the M.B.A. degree from the Instituto de Empresa, in 2002, and the Ph.D. degree in computer science from Universidad Politécnica de Madrid, Spain, in 2005.

He was with Østfold University College, Norway, and Universidad Carlos III de Madrid, Spain. He has been a Software Engineer, a Project Manager, and a Software Engineering Consultant at several companies, including the Spanish IT Leader INDRA. He is currently a Full Professor with Escuela Técnica Superior de Ingenieros Informáticos, Universidad Politécnica de Madrid. His research interests include software governance, management information systems, software project management, people in software projects, software and service process improvement, and management information systems.

Prof. Colomo-Palacios is an Editorial Board Member and an Associate Editor for several international journals, such as *IEEE SOFTWARE*, *Computer Standards and Interfaces*, and *IET Software*.



FRANCISCO J. GARCÍA-PEÑALVO (Member, IEEE) received the degree in computing from the University of Salamanca, the degree in computing from the University of Valladolid, Spain, and the Ph.D. degree from the University of Salamanca (USAL).

From 2007 to 2009, he was the Vice Chancellor of Innovation with USAL. He is currently the Head of the Research Group Interaction and eLearning and a Full Professor with the Computer Science Department, USAL. He has led and participated in more than 50 research and innovation projects. He has authored more than 300 articles published in international journals and conferences. His main research interests include e-learning, computers and education, adaptive systems, web engineering, semantic webs, and software reuse.

Prof. García-Peñalvo is a member of the Program Committee of several international conferences and a reviewer of several international journals. He was a Guest Editor of several special issues in international journals, such as the *Online Information Review*, *Computers in Human Behavior*, and *Interactive Learning Environments*. He is currently the Editor-in-Chief of the *International Journal of Information Technology Research and Education in the Knowledge Society* journal. He is also a coordinator of a multidisciplinary Ph.D. Program on Education in the Knowledge Society.



ANTONIO MARTINI received the Ph.D. degree from the Chalmers University of Technology, Gothenburg, Sweden, in 2015.

He is currently a Professor with the University of Oslo, Norway. He has also worked as a software developer and independent consultant. He has led a ten-year project focused on managing technical debt, collaborating with several large companies in Northern and Central Europe. He has received several funds for the commercialization transfer to the practice of research results, including a Marie Curie EU project related to the TecnioSpring+ Program, and an Innovation Project for the Industrial Sector from the Research Council of Norway. He has published more than 50 contributions to the top software engineering conferences and journals. His research interests include technical debt, software architecture, technical leadership, and agile software development. His experience covers software engineering and management in several contexts, such as large, embedded software companies, small, web companies, business-to-business companies, and startups.

Prof. Martini is active in the research community as an editor and a reviewer for top IEEE software engineering conferences and journals, such as *IEEE SOFTWARE* and *IEEE TRANSACTIONS ON SOFTWARE ENGINEERING*. He is involved in the organization of IEEE conferences, such as TechDebt@ICSE, ICSA, and SEAA Euromicro.

...