Context-aware Recommendation of Visualization Components

Martin Voigt*, Stefan Pietschmann*, Lars Grammel†, and Klaus Meißner*
*Technische Universität Dresden, Chair of Multimedia Technology,
Dresden, Germany
Email: martin.voigt, stefan.pietschmann, klaus.meissner@tu-dresden.de
†University of Victoria, Computer Human Interaction and Software Engineering Lab
Victoria, British Columbia, Canada
Email: lars.grammel@gmail.com

***Abstract*—Although many valuable visualizations have been developed to gain insights from large data sets, selecting an appropriate visualization for a specific data set and goal remains challenging for non-experts. In this paper, we propose a novel approach for knowledge-assisted, context-aware visualization recommendation. Both semantic web data and visualization components are annotated with formalized visualization knowledge from an ontology. We present a recommendation algorithm that leverages those annotations to provide visualization components that support the users' data and task. We successfully proved the practicability of our approach by integrating it into two research prototypes.**

***Keywords*-recommendation, visualization, ontology, mashup**

I. INTRODUCTION

Visualization is a powerful way of gaining insight into large data sets. Therefore, a myriad of visualizations have been developed in recent decades. To bridge the gap between data and an appropriate visual representation, models like the visualization pipeline [1] have been developed and implemented in numerous tools. As one part of this process, the mapping of data to a graphic representation is critical because only small subsets of existing visualization techniques are expressive and effective for the selected data in a specific context. Generally, domain-specific data can be visualized either using tools which were developed specifically for that domain and use case, or using generic visualization systems. The development of the former requires extensive knowledge by visualization and domain experts, and is usually costly and time-consuming. Thus, in many cases generic visualization tools are preferable, because they are quickly available and reusable in different contexts. Using such tools, domain experts can directly get the information they need out of their data. However, these tools typically require them to select the visualization type and to specify the visual mappings, which can be difficult because they often lack the necessary visualization knowledge [2]. Knowledge-assisted visualization can address this problem by representing and leveraging formalized visualization knowledge to support the user [3]. Suggesting automatically generated visualizations to the user is one promising approach to aid domain experts in constructing visualizations [2], [4].

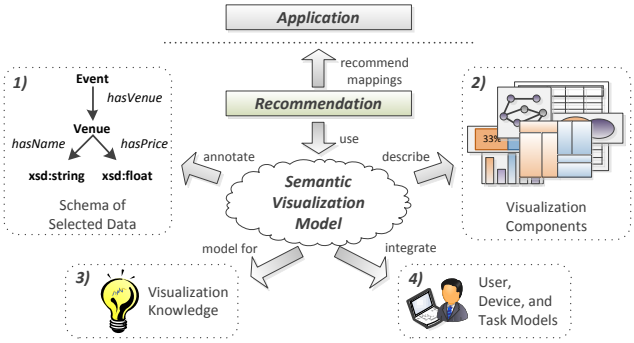

Figure 1. Overview of our goal to recommend mappings of a data source (1) to a visualization component (2) based on a semantic model utilizing visualization knowledge (3) and context information (4).

Consider for example a semantic web data set comprising a list of events hosted at different venues with varying fees. A business user with less visualization experience wants to get an overview of how expensive the events are using his laptop. Thus, he selects a subgraph from a semantic data set as shown in Fig. 1-1 containing two classes (EVENT, VENUE) linked by a Property (*hasVenue*) as well as two Data Properties (*hasName*, *hasPrice*). To map this data to a compatible visualization component (Fig. 1-2), a user needs visualization knowledge (Fig. 1-3). Context information (Fig. 1-4) about the user (knowledge, skills), his device (hard- /software capabilities) and his task (get overview) must also be considered to create a successful mapping. We strive for a generic recommendation approach utilizing and understanding these different ingredients based on a common semantic knowledge model to facilitate the automated visualization process for different tools.

Our goal of creating a knowledge-assisted, context-aware system which recommends visualization components involves a number of challenges, which are addressed by this paper. First, a formalized vocabulary for the interdisciplinary visualization domain is needed. To this end, we have developed a modular visualization ontology called VISO. Second, means to semantically describe visualization characteristics of both data sources and visualization components must be

 ISBN: 978-1-61208-181-6

provided. Therefore, we propose the linking and annotation of semantic web data and component descriptors with concepts of VISO. Third, appropriate visualization components must be discovered for a certain set of data, which includes the deduction of applicable mappings between data and graphic representations. We present a discovery algorithm which takes the aforementioned formalized visualization knowledge and given user requirements into account to search for compatible visualization components. Finally, component candidates need to be ranked with regard to the user, usage and device context, e. g., to consider user language, screen real estate, and available plugins on the device. We have developed a corresponding ranking algorithm for the mappings, i. e., component candidates resulting from the discovery. It explicitly takes into account the visualization knowledge, assigned domain concepts, the user and device context as well as optional criteria from the end user to achieve a context- and task-aware rating. We show the feasibility and practicability of our approach with two prototype implementations: an Eclipse-based visualization tool for semantic multimedia data and a mashup-based visualization workbench called *VizBoard*.

The remainder of this paper is structured as follows. First, we discuss related work in the fields of automated visualization, semantic models for visualization, and semantic-based component recommendation in Section II. Then, Section III introduces our visualization ontology VISO in detail and clarifies how it is applied to describe visualization components and data sources. Afterwards, we present the corresponding recommendation algorithm separated into discovery and ranking in Section IV. Section V gives a brief overview of the prototype implementations and discusses our findings. Finally, in Section VI we conclude the paper and outline future work.

II. Related Work

The recommendation algorithm presented in this paper builds on previous research in the three different research areas (1) automated visualization, (2) semantic visualization models, (3) mechanisms for semantics-based component discovery and ranking. We will now discuss the state of the art in those three areas.

A. Automated Visualization

Several automatic visualization systems have been developed to help users to create visualizations. They produce visualization specifications based on user-selected data and implicitly or explicitly represented visualization knowledge. We distinguish between data-driven, task-driven, and interaction-driven approaches.

Data-driven approaches analyze the meta-model of the data and potentially instance data to generate visualization specifications. Mackinlay addressed the problem of how to automatically generate static 2D visualizations of relational information in his APT system [5]. It searches the design space of all possible visualizations using expressiveness criteria and then ranks them using effectiveness criteria. Gilson et al. developed an algorithm that maps data represented in a domain ontology to visual representation ontologies [6]. Their visual representation ontologies describe single visualization components, e. g., tree maps. A semantic bridging ontology is used to specify the appropriateness of the different mappings. Our automated visualization approach is similar to the one by Gilson et al. in that both data and visualization components are described using ontologies. The main limitation of data-driven approaches is that they do not take other information such as the user's task, preferences or device into account. Task-driven and interaction-driven approaches usually build on the data analysis ideas present in data-driven approaches, but go beyond them.

The effectiveness of a visualization depends on how well it supports the user's task by making it easy to perceive important information. This is addressed by **task-driven approaches**. Casner's BOZ system analyzes task descriptions to generate corresponding visualizations [7]. However, BOZ requires detailed task descriptions formulated in a structured language and is limited to relational data. The SAGE system by Roth and Mattis extends APT to consider the user's goals [8]. It first selects visual techniques based on their expressiveness, then ranks them according to their effectiveness, refines them by adding additional layout constraints (e.g., sorting), and finally integrates multiple visualization techniques if necessary. In constrast to SAGE and BOZ, our algorithm is ontology-based to allow for reasoning and it leverages device and user preference information.

Visual data analysis is an iterative and interactive process in which many visualizations are created, modified and analyzed [2]. **Interaction-driven approaches** consider either the user interaction history or the current visualization state to generate visualizations that support this process. Mackinlay et al. have developed heuristics that use the current visualization state and the data attribute selection to update the current visualization or to show alternative visualizations [9]. Behavior-driven visualization recommendation monitors users' interactions with visualizations, detects patterns in the interaction sequences, and infers visual tasks based on repeated patterns [10]. The current visualization state and the inferred visual task are then used to recommend more suitable visualizations. Interaction-driven approaches leverage implicit state information such as the interaction history, but they consider neither task information that is explicitly expressed by the user, nor user preferences or device constraints.

In summary, while our work builds on many ideas from automated visualization approaches, in particular the work by Gilson et al. [6], it is extensible in terms of visualization components, and it considers task, user preferences and device capabilities. In contrast to generative approaches [5],

 ISBN: 978-1-61208-181-6

[7]–[9], the strength of using visualization components is that such components are optimized for the visual metaphor they represent.

B. Formalizations of Visualization Knowledge

As shown in the previous section, automated visualization requires one or more models to bridge the gap between data and suitable graphic representations. In this regard, prevalent approaches use different concepts, such as rules [8], heuristics [9], and semantic models [6]. We share the view of Gilson et al. [6] that semantic technologies are the methods of choice today. They allow for capturing and formalizing expert knowledge in a readable and understandable manner for humans as well as machines. Therefore, they provide an effective solution for automated recommendation. Further, the current technologies facilitate an easy and dynamic reuse of existing semantic models in new scenarios.

Actually, only few academic works have explored semantic web technologies as means to capture visualization knowledge for describing and recommending resources. Duke et al. [11] were the first proposing the need for a visualization ontology. Their promising approach captures an initial set of concepts and relations of the domain comprising data, visualization techniques, and tasks. Potter and Wright [12] combine formal taxonomies for hard-/software capabilities, sensory experience as well as human actions to characterize a visualization resource. Similarly, Shu et al. [13] use a visualization ontology to annotate and query for visualization web services, with regard to their (1) underlying data model and (2) visualization technique. While the former is a taxonomy comprising various kinds of multidimensional data sets, the latter builds on the data module to classify the graphic representations. For our work, their data taxonomy is not flexible enough as we need to support graph-based data structures for example. Gilson et al. [6] employ three dedicated ontologies to allow for automatic visualization: The first one captures domain semantics and instance data to visualize; the second one describes a particular graphic representation; the final ontology contains expert knowledge to foster the mapping from domain to visualization concepts. In contrast, we allow for a more flexible and generic linking of both sides by annotating each with VISO concepts instead of the explicit, manual creation of an additional ontology. Rhodes et al. [14] aimed to categorize, store and query information about software visualization systems using a visualization ontology as the underlying model. Their approach facilitates methods for specifying data, graphic representation, or the skill of users.

In summary, we share the goal of the works presented above: defining a formalized vocabulary to describe and recommend visualization resources. However, as we strive for a context-aware recommendation we need a more comprehensive and detailed model that covers not only data and graphical aspects, but also represent the user, his activity, and device.

C. Semantics-Based Component Discovery and Ranking

When it comes to finding and binding adequate services for a desired goal, such as visualizing semantic data as we are, *Semantic Web Services* (SWS) tackle a very similar problem. SWS research provides solutions for finding a service or service composition that fulfills a goal or user task based on certain instance data. Therefore, they employ a formal representation of the services' functional and non-function semantics – usually based on description logics – to facilitate reasoning. Based on this, they strive for the automation of the service life-cycle including the discovery, ranking, composition, and execution of services through proper composition environments.

The discovery of suitable semantic services employs either complete semantic service models, e. g., in OWL-S [15] and WSMO [16], or semantic extensions to existing description formats, as proposed by SAWSDL [17] and WSMO-Lite [18]. The former *top-down* approaches are usually very expressive, but descriptions are complex and time-consuming to build. The latter *bottom-up* approaches add semantic annotations, i. e., references to concepts in external ontologies, to WSDL. Even though the above-mentioned solutions cannot be directly applied to our problems, e. g., due to their limitation to web services formats and design principles (stateless), we follow the idea by extending a mashup component description language with semantic references. Thereby, visualization components can be described regarding their data, functional and non-functional semantics, including references to formalized visualization knowledge.

In SWS discovery, suitable services are searched based on a formalized goal or task definition, which is usually a template of an SWS description. Thus, the desired data and functional interface is matched with actual service models. The corresponding algorithms either use measures like text and graph similarities, which restricts the applicability to design-time, or determine the matching degree of services, operations, etc., using logic relationships between annotated concepts as in [19]. In contrast to SWS, we follow a data-driven approach, in which semantically annotated data forms the input for the discovery of suitable candidates. The direct generation of SWS goals from a selected data set is not feasible. Therefore, we individually match data types, functional interface and hard-/software requirements with and between data and visualization components based on shared conceptualizations. Based on this measure, compatible visualization components can be found.

Ranking of service candidates in SWS bears a number of similarities with ranking visualization components for a certain data set. It is usually based on non-functional properties, such as QoS and context information (user profile, device

 ISBN: 978-1-61208-181-6

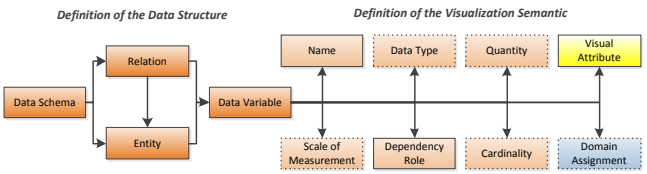

Figure 2. Overview of the VISO data module

capabilities). To this end, a number of sophisticated concepts exist, e. g., for multi-criteria ranking based on semantic descriptions of non-functional service properties [16] and for context sensitive ranking [20]. Since these algorithms are rather generic and work on a semantic, non-functional level, they likewise apply to our concept space.

In summary, the discovery and ranking of candidate services for a predefined goal in SWS research follows a similar principle as our work. Yet, its solutions can not be directly applied to our problems. For one, there is a difference in component models, e. g., with regard to statefulness of visualization components. Furthermore, the discovery of visualization components can not be based on predefined, formalized goal descriptions, as it basically depends on semantic data which is annotated with visualization knowledge. For the annotation of visualization components with semantic concepts though, we can apply the ideas of SAWSDL and WSMO-Lite to the component descriptions. To *link* semantic data with visualization components, a shared conceptualization of visualization knowledge is needed. Therefore, the next section presents VISO.

III. VISO: A Modular Visualization Ontology

The foundation of our visualization recommendation approach is a formalized, modular visualization ontology called VISO [21]. It provides a RDF-S/OWL vocabulary for annotating data sources and visualization components, contains factual knowledge of the visualization domain, and serves as a semantic framework for storing contextual information. Altogether, it serves as a *bridging ontology* between semantic data and visualization components by offering shared conceptualizations for all four mapping ingredients shown in Fig. 1. Details of VISO and its development are described in [21]. The seven VISO modules (data, graphic, activity, user, system, domain, and facts) represent different facets of data visualization domain. They refer to each other and to existing ontologies as needed. VISO modules can be extended to accommodate new concepts.

1) Data: Fig. 2 shows the data module which contains concepts for describing data variables and structures for visualization purposes. While all concepts are employed to describe visualization components, those with dotted lines are also used to annotated semantic data. The vocabulary is especially need at component-side to describe possible input data in a generic manner as the most of visualizations allow for representing domain independent data. For example, a simple table may visualize data about hotels, cars, or humans. Using this vocabulary, we specify only the data structure and characteristics. As can be seen, a DATA SCHEMA consists of ENTITY and RELATION concepts. The latter represent links between ENTITY concepts like an OWL Object Property. Both ENTITIES and RELATIONS can contain DATA VARIABLE concepts, whose equivalent in OWL space is a Data Property. For example, the semantic data model of a table visualization component would be represented as one ENTITY concept with several DATA VARIABLES for every column. Further semantics, e. g., the SCALE OF MEASUREMENT and CARDINALITIES – specified using built-in OWL constraints – can be defined on the DATA VARIABLE concepts (cf. Fig. 2) to constrain its, e. g., its scale. By linking the concepts from the data module to the VISUAL ATTRIBUTE concepts from the graphic module, we bridge the gap between data attribute and visual elements and properties.

2) Graphics: The graphics module conceptualizes the semantics of GRAPHICAL REPRESENTATIONS and their parts, e. g., their VISUAL ATTRIBUTES. Concrete graphical representations, e. g., *scatter plot* and *treemaps*, and concrete visual attributes such as *hue* or *shape* are contained as instance data. The concepts from the graphics module are used to semantically annotate visualization components and to define visualization knowledge in the facts module.

3) Activity: The activity module models user activity in a visualization context. It builds on the ontology-based task model by Tietz et al. [22], which distinguishes betweens high-level, domain specific TASKS and low-level, generic ACTIONS, similar to the distinction made by Gotz and Zhou [23]. We have extended the action taxonomy of Tietz's task model by separating data- and UI-driven ACTIONS, and by formalizing ACTIONS from the visualization literature such as *zoom* and *filter*. This enables the fine-grained annotation of interaction functionality in visualization components.

4) User: The user module formalizes user PREFERENCES and KNOWLEDGE. Users can, for example, have PREFERENCES for different GRAPHICAL REPRESENTATIONS, and their *visual literacy* can differ. As manifold context models for users, their characteristics and preferences, already exist those can be seamlessly integrated and used here.

5) System: The system module facilitates the description of the device context, e. g., installed PLUG-INS or SCREEN SIZE. It also allows us to annotate a visualization component with its system requirements. Again, sophisticated models for device characteristics and context exist, which were reused or integrated in this module. As an example, we borrow concepts from the *CroCo* ontology [24], which combines user, usage, system, and situational context from different existing works developed by academia.

6) Domain: Many visualizations are domain-specific, and thus it is important to consider the domain context during

 ISBN: 978-1-61208-181-6

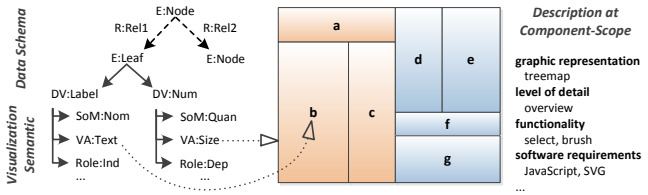

Figure 3. Description of a treemap visualization in VISO.

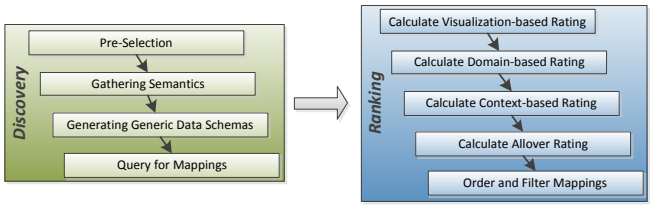

Figure 4. Overview of the recommendation algorithm.

visualization recommendation. However, it is not feasible to model all possible visualization domains. Instead, we support linking to existing domain ontologies. A DOMAIN ASSIGNMENT links VISO concepts, e. g., a DATA VARIABLE (cf. Fig. 2), to concepts from specific domain ontologies. As this assignment is usually created automatically during data analysis, it can be qualified with a probability value reflecting its accuracy. Thus, the analysis of a data source with ambiguous Properties, such as *typeOfJaguar* and *typeofApple*, will result in multiple domain assignments with probabilities below 1. In contrast, a Data Property *hasPrice* from our motivating example could be annotated with *price* and a probability of 1. A visualization component supporting DATA VARIABLE annotated with the more general concept *value* could be inferred as a possible mapping.

7) Facts: The visualization recommendation also depends on factual visualization knowledge to select suitable visualizations. Thus, we formalized knowledge from the information visualization community, e. g., verified statements such as "position is more accurate to visualize quantitative data than color" [25], to make it machine-processable. These rankings and constraints are formalized in rules in the Facts module. These rules use of the vocabulary of the other VISO modules in their conditions part, e. g., SCALE OF MEASUREMENT (*quantitative*) and the VISUAL ATTRIBUTE (*position, color*) for the mentioned example. If the conditions are matched, a rating is assigned to the corresponding visualization component description.

To give a more practical insight, the following example explains how a *treemap* visualization is described using VISO (see Fig. 3). First, the hierarchical data structure of the *treemap* is specified. At the top level, a *Node* ENTITY represents the whole *treemap*. It can contain *Leaf* ENTITIES and *Node* ENTITIES. The label and size variables of *Leafs* can be configured. They are annotated with visualization semantics, e. g., the SCALE OF MEASUREMENT for the label variable is *nominal* and the ROLE of the size variable is *dependent*. Further domain semantics could be added to the variables, e. g., WordNet (http://wordnet.princeton.edu/) concepts such as *value*. In addition to the data structure and the variables, more general semantics such as the kind of GRAPHIC REPRESENTATION (*treemap*), the LEVEL OF DETAIL (*overview*) and possible ACTIONS (*select*, *brush*) are defined for the entire visualization component.

In order to facilitate the construction of visual mappings, VISO is used to annotate visualization components and semantic web data. In the latter case, we annotate only RDF Properties on a schema level. RDF Properties hold the data that will be visualized, e. g., literals and relations, whereas RDF classes assemble such properties and do not provide additional information that would be relevant for visualization. Similarly, annotations are made on the schema level, because instance data annotation would be redundant. Consider our motivating example (see Fig. 1-1), comprising the Property *hasPrice*. Because the Property has the RDFS Range *xsd:float*, the required DATA TYPE is already defined and the SCALE OF MEASUREMENT is *quantitative*. The number of distinct values (CARDINALITY) and the overall number of values (QUANTITY) can be extracted from the instance data. While a DOMAIN ASSIGNMENT is not mandatory, it could be applied, e. g., to *price* from the WordNet vocabulary.

In summary, VISO models the concepts required for data visualization. It is used to annotate data, to describe visualization components, to represent context and factual knowledge. Together, these different pieces are the foundation of our visualization recommendation algorithm.

IV. VISUALIZATION RECOMMENDATION ALGORITHM

The visualization recommendation algorithm creates an ordered list of mappings to visualizations components for the selected data (see Fig. 1-1). It considers contextual information (e.g., device, user model) as well as knowledge about the full data source. While the user model and device are mandatory inputs, visualization specific information like the required LEVEL OF DETAIL or the requested kind of GRAPHICAL REPRESENTATION can be provided as optional constraints.

The algorithm consists of two separate steps: discovery and ranking (see Fig. 4). Both steps leverage semantic knowledge formulated as VISO concepts (see Section III). In the discovery step, potential mappings between data and widgets are generated based on functional requirements. The resulting visualization set is then sorted in the ranking step using the formalized visualization knowledge and domain concepts, as well as by contextual and visualization specific information.

 ISBN: 978-1-61208-181-6

A. Discovery of Mappings

The discovery algorithm generates a set of mappings from the selected data to visualization components (see Fig. 4). First, potentially applicable widgets are identified and non-applicable components are ruled out (**pre-selection**), since limiting the set of available visualization components early improves the overall algorithm performance. To be applicable, a widget has to (1) be compatible with the target device (e. g., required PLUGINS must be available), (2) support the number of selected Data Properties, and (3) support visualization and task specific requirements (e. g., showing an *overview*), if specified by the user. As can be seen, these constraints don't relate to data structure or semantics of the data variables, yet. Semantic matching is carried out with the resulting component candidates in the following step.

Second, semantics, e. g., the SCALE OF MEASUREMENT, DATA TYPE, and QUANTITY (Fig. 2) of the selected Properties are fetched (**gathering semantics**). For example, the DATA TYPE *xsd:float* or the SCALE OF MEASUREMENT *quantitative* of the property *hasPrice* (see Fig. 5-3)) would get retrieved. This semantic information about the Properties is used in the next steps.

Third, we **generate generic data schemas**, which are then used to query for mappings. We distinguish between tabular and graph-based DATA SCHEMAS. TABULAR DATA SCHEMAS contain one ENTITY with several DATA VARIABLES (Fig. 5-1). GRAPH-BASED DATA SCHEMAS contain two or more linked ENTITIES, each containing zero or more variables (Fig. 5-2).

If a **single class** has been selected, a TABULAR DATA SCHEMA is chosen and an ENTITY is created for that class. For every selected Data Property of this class, a DATA VARIABLE with the semantic information (that was retrieved in the previous step) is attached to the ENTITY.

If **several classes** have been selected, we generate both a tabular and a graph-based DATA SCHEMA. For the TABULAR DATA SCHEMA, a single ENTITY gets created. For any selected Data Property from those classes, a DATA VARIABLE with the semantic information is attached to the single ENTITY. This reduces the graph-based data structure to a tabular structure. For example, consider the data shown in Fig. 5-3. The algorithm would create one ENTITY with two DATA VARIABLES. The first DATA VARIABLE would represent the semantics of *hasName*, e. g., the *nominal* SCALE OF MEASUREMENT, and the second DATA VARIABLE would represent *hasPrice*. The GRAPH-BASED DATA SCHEMA gets generated as follows: Beginning with a class from the input data, e. g., *Event* in Fig. 5-3, an ENTITY is created. Similar to the other cases, DATA VARIABLES and their semantics are attached to this ENTITY for the selected Data Properties linked to the class. Next, for each Object Property connected with the class, a RELATION gets generated. If the target class for that RELATION has not been processed yet, it is created and processed in a similar way. This depth-first processing continues until the current part of the input graph is completely traversed. If there are multiple unconnected classes in the input, the algorithm continues with those until all graph components are processed. For example, the algorithm would generate the DATA SCHEMA illustrated in Fig. 5-4 by processing the input data structure shown in Fig. 5-3.

Fourth, the mappings are generated by querying the semantic representations of the pre-selected components with the generic DATA SCHEMAS that were computed in the previous step (**query for mappings**). The mappings include permutations of DATA VARIABLES with similar semantics, and thus the number of mappings may be higher than the number of existing components. Using the data structure generated by the algorithm for the example shown in Fig. 5-4, both the *scatter plot* (Fig. 5-1) and the *treemap* (Fig. 5-2) would fit on the level of data structure. However, only the *treemap* is a suitable mapping due to the annotated semantics which are also employed by querying. The *scatter plot* is not suitable because it has two *quantitative* DATA VARIABLES where both a *nominal* and a *quantitative* DATA VARIABLE are required. The generated set of mappings from the selected data to the visualization components is ranked in the next part of the algorithm.

B. Ranking of Mappings

The ranking step of the algorithm sorts the visual mappings that were generated by the previous discovery step. While the discovery step identifies valid mappings and visualization components that satisfy functional criteria, it does not take their effectiveness into account. To sort the mappings by their effectiveness, the ranking step applies factual visualization knowledge, domain assignments and contextual user and device information.

1) Factual Visualization Knowledge: The factual visualization knowledge (see Section III) is defined by a set of rules which consist of a condition and a rating. The conditions are specified using the VISO vocabulary for the visualization components. For each widget, the ratings of all rules that are met are added to its specification. During runtime, the arithmetic mean of all ratings r_{v_i} is calculated for the discovered component of each visual mapping. For example, we formalized rules to rate the appropriateness of visual encodings for quantitative data [25]. The *quantitative* DATA VARIABLE of the *treemap* (Fig. 5-2) is rated with *0.5* as it employs "only" *size* and not *position*.

2) Domain Assignments: Domain concepts from various ontologies are assigned to both the data input and the visualization components with a certainty value (see Section III). For each pair of input Property and DATA VARIABLE of the visualization component, we calculate a semantic similarity rating between 0 and 1 (e. g., using [26]), if they both have a domain concept assigned with a certainty greater

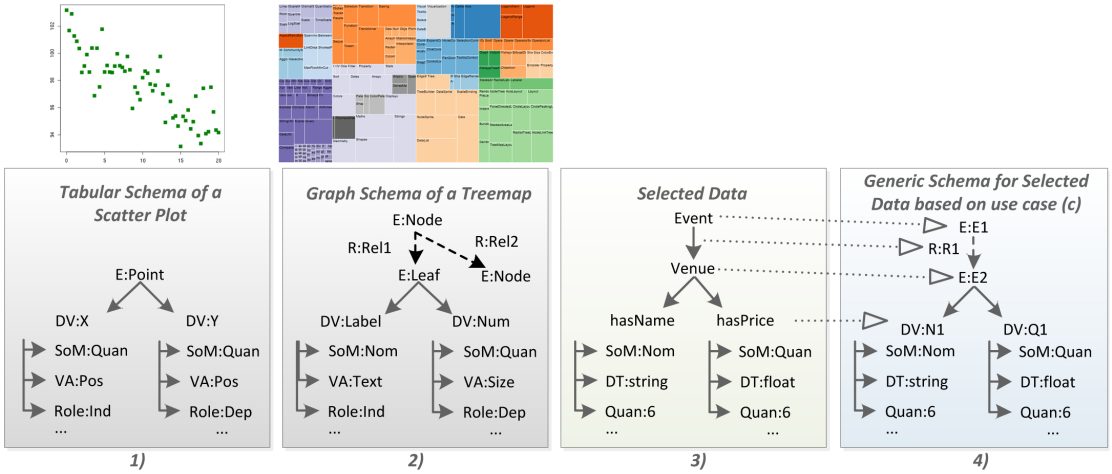

Figure 5. Comparison of the data structure and the annotation between 1) a scatter plot, 2) a treemap, 3) user's selected data, and 4) a generic equivalent of the selected data.

than 0. The final rating r_{d_j} is the product of the semantic similarity and the arithmetic mean of both certainties. In our example (see Sect. III), we used *value* and *price* from WordNet to annotate the *quantitative* DATA VARIABLE of the *treemap* and the Property *hasPrice* from our data set, each with a certainty of *1*. Using [26], we get a rating r_d=0.9094.

3) User and Device Information: The rules for the context-based rating r_c are part of the knowledge base and use the VISO vocabulary, similar to the factual visualization knowledge. The rules are executed during runtime and employ the above mentioned identifiers of users' and their device context models. For example, we construct a SPARQL-based rule that counts the use of different GRAPHIC REPRESENTATIONS, like *treemaps* or *scatter plots*. This rule assigns a rating r_{c_k} between 0 and 1 to the visual mappings.

The three different kinds of rating are combined using an arithmetic mean. The overall rating has a range between 0 and 1. We weight all three rating types equivalently for two reasons. First, the assignment of a (quantitative) rating is often subjective. Second, a profound user study is needed to evaluate the impact of each knowledge base in users visualization selection process what will be future work. As x, y, and z are the number of each kinds of rating, the overall rating R for each mapping is calculated in terms of

$$R = \frac{1}{3}\left(\frac{1}{x}\sum_{i=1}^{x} r_{v_i} + \frac{1}{y}\sum_{j=1}^{y} r_{d_j} + \frac{1}{z}\sum_{k=1}^{z} r_{c_k}\right)$$

The list is ordered based on the combined ratings R for each mappings. This ranking could be used to automatically display the top mapping to the user as a visualization, or, as in our approach, to let the user pick one of the top n ranked visualizations. We next discuss our implementation of the visualization recommendation algorithm in two research projects.

V. IMPLEMENTATION AND DISCUSSION

To realize the concepts discussed above, we first developed VISO as an open, modular ontology (available at http://purl.org/viso/) based on OWL DL. It is comprised of concepts, properties and instance data from the visualization domain, as well as factual knowledge modeled using Jena Rules (http://jena.sf.net/). Details on the design process and decisions can be found at [21].

We then implemented our generic recommendation approach and integrated it with two existing research projects: *KIMM* [27] and *CRUISe* [28]. The algorithms build on Jena to manage all semantic models and employ SPARQL 1.1 to query the knowledge bases and to create the mappings.

Within the frame of *K-IMM*, we enhanced the Eclipse RCP-based application *Sim²* which allows for visualizing RDF-based multimedia data. With a wizard we enable users to select classes and properties from theirdata set for visualization. Since all views are semantically annotated with concepts from VISO, this input can be used for our discovery algorithm, which recommends suitable visualizations for the selected data. Since *Sim²* neither tracks nor uses any context information, this integration was limited to the discovery. The context-aware rating was omitted in this prototype. In this prototype, the discovery mechanism, which relies only on functional and objective matching, was able to identify suitable visualizations.

Our approach is also an integral part of *VizBoard*, an information visualization workbench for semantic data based on the mashup platform *CRUISe*. *CRUISe* facilitates the dynamic, context-aware composition of mashups from distributed web resources. Hence, it builds on a universal component model which includes semantic descriptors. We

 ISBN: 978-1-61208-181-6

encapsulated a number of well-known visualizations, including several from the Protovis library [29], and employed the descriptors to realize the annotation with VISO concepts. The discovery and ranking algorithms were integrated as part of a multi-step wizard, which results in the context-aware recommendation of suitable visualization components as basis for a mashup UI. Using this implementation, we could also evaluate our ranking algorithm. Even though it performs as expected, more work is needed for two reasons. First, we build on visualization knowledge, e. g., the expressiveness of visual attributes for quantitative data, that reflects the current state of knowledge in the field of information visualization. However, due to limited empirical evidence and different expert opinion, some of the current guidelines are not accepted and could change in the future. Second, user studies to identify the impact created by the visualization, domain, and context knowledge in the visualization process could improve the weighting of these three components.

As with many search mechanisms, defining the goal of the visualization selection process is challenging. We argue that the query creation should allow for defining requirements and options, e. g., "I like to visualize A and B, and C if possible", in an uncomplicated way. Hence, a sophisticated user interface should assist the user during the goal definition. Furthermore, our discovery algorithm needs to be extended to support optional input.

A limitation of semantic approaches like ours is the need of descriptions respectively of annotations. Thus, it is up to component authors and data providers to augment their components/data using the VISO concepts correctly. In this regard, adequate tool support would be beneficial.

VI. Conclusion and Further Work

Selecting an appropriate visualization for a specific data set in a specific scenario remains challenging for non-experts. Therefore, we have presented a context-aware and knowledge-assisted approach to recommend suitable visualizations for semantic web data. Its foundation is the modular visualization ontology VISO which provides the vocabulary to annotate both data sources and visualization components. Based on these shared concepts from the visualization domain, our recommendation algorithm covers both discovery and context-aware ranking of suitable graphic representations: First, possible mappings from data to visual encodings are identified using the selected data, its semantics, and other functional information. Then, quantitative ratings for each mapping are calculated with respect to visualization knowledge, domain concept relations and context information.

As our implementations shows that the approach can recommend visualization components based on semantics from different sources, the discussion shows some directions for future work. We will investigate a tool for the semi-automatic annotation of visualization semantics for semantic web data. We are also planning to conduct a user study to identify and model the interdependencies between the knowledge bases employed within the ranking. To enhance users interactive selection of data, task- and visualization-specific input for the algorithm, we are working on a faceted browser which will distinguish between requirements and weighted optional criteria.

Acknowledgment

This work is partly funded by the German Federal Ministry of Education and Research under promotional reference number 01IA09001C.

The authors wish to thank Michael Aleythe for implementation support as well as David Rusk and Patrick Gorman for helpful editing suggestions.

References

[1] R. Haber and D. A. McNabb, "Visualization idioms: A conceptual model for scientific visualization systems," *Visualization in Scientific Computing*, pp. 74–93, 1990.

[2] L. Grammel, M. Tory, and M.-A. Storey, "How information visualization novices construct visualizations," *IEEE Transactions on Visualization and Computer Graphics*, vol. 16, pp. 943–952, November 2010.

[3] M. Chen, D. Ebert, H. Hagen, R. Laramee, R. van Liere, K.-L. Ma, W. Ribarsky, G. Scheuermann, and D. Silver, "Data, information, and knowledge in visualization," *Computer Graphics and Applications, IEEE*, vol. 29, no. 1, pp. 12 –19, Jan. 2009.

[4] J. Heer, F. van Ham, S. Carpendale, C. Weaver, and P. Isenberg, *Creation and Collaboration: Engaging New Audiences for Information Visualization*. Berlin, Heidelberg: Springer-Verlag, 2008, pp. 92–133.

[5] J. Mackinlay, "Automating the design of graphical presentations of relational information," *ACM Trans. Graph.*, vol. 5, no. 2, pp. 110–141, 1986.

[6] O. Gilson, N. Silva, P. Grant, and M. Chen, "From web data to visualization via ontology mapping," in *Computer Graphics Forum*, vol. 27, no. 3. Blackwell Publishing Ltd, Sep 2008, pp. 959–966.

[7] S. M. Casner, "Task-analytic approach to the automated design of graphic presentations," *ACM Trans. Graph.*, vol. 10, pp. 111–151, April 1991.

[8] S. F. Roth and J. Mattis, "Automating the presentation of information," in *Artificial Intelligence Applications, 1991. Proc. of 7th IEEE Conf. on*, vol. 1. IEEE, 1991, pp. 90–97.

[9] J. Mackinlay, P. Hanrahan, and C. Stolte, "Show me: Automatic presentation for visual analysis," *IEEE Transactions on Visualization and Computer Graphics*, vol. 13, no. 6, pp. 1137–1144, Nov. 2007.

[10] D. Gotz and Z. Wen, "Behavior-driven visualization recommendation," in *IUI '09: Proceedings of the 13th international conference on Intelligent user interfaces*. New York, NY, USA: ACM, 2009, pp. 315–324.

 ISBN: 978-1-61208-181-6

[11] D. Duke, K. Brodlie, D. Duce, and I. Herman, "Do you see what i mean? [data visualization]," *Computer Graphics and Applications, IEEE*, vol. 25, no. 3, pp. 6–9, May 2005.

[12] R. Potter and H. Wright, "An ontological approach to visualization resource management," in *DSV-IS*, 2006, pp. 151–156.

[13] G. Shu, N. Avis, and O. Rana, "Bringing semantics to visualization services," *Advances in Engineering Software*, vol. 39, no. 6, pp. 514–520, 2008.

[14] P. Rhodes, E. Kraemer, and B. Reed, "Vision: an interactive visualization ontology," in *ACM-SE 44: Proceedings of the 44th annual Southeast regional conference*. New York, NY, USA: ACM, 2006, pp. 405–410.

[15] D. Martin, M. Burstein, D. McDermott, S. McIlraith, M. Paolucci, K. Sycara, D. McGuinness, E. Sirin, and N. Srinivasan, "Bringing semantics to web services with owl-s," *World Wide Web*, vol. 10, pp. 243–277, Sep. 2007.

[16] D. Fensel, M. Kerrigan, and M. Zaremba, *Implementing Semantic Web Services: The SESA Framework*. Springer, 2008.

[17] J. Farrell and H. Lausen, "Semantic annotations for WSDL and XML Schema," http://www.w3.org/TR/sawsdl/, W3C, Aug. 2007.

[18] J. Kopecký and T. Vitvar, "Wsmo-lite: Lowering the semantic web services barrier with modular and light-weight annotations," in *Proc. of the Intl. Conf. on Semantic Computing*, Aug. 2008, pp. 238–244.

[19] Y. Chabeb, S. Tata, and A. Ozanne, "YASA-M: A semantic web service matchmaker," *Proc. of the Intl. Conf. on Advanced Information Networking and Applications*, pp. 966–973, Apr. 2010.

[20] F. Gilles, V. Hoyer, T. Janner, and K. Stanoevska-Slabeva, "Lightweight composition of ad-hoc enterprise-class applications with context-aware enterprise mashups," in *Proc. of the Intl. Conf. on Service-Oriented Computing*. Springer, 2009, pp. 509–519.

[21] M. Voigt and J. Polowinski, "Towards a unifying visualization ontology," TU Dresden, Institut fuer Software und Multimediatechnik, Dresden, Germany, Technical Report TUD-FI11-01, Mar. 2011, ISSN: 1430-211X.

[22] V. Tietz, G. Blichmann, S. Pietschmann, and K. Meiner, "Task-based recommendation of mashup components," in *Proceedings of the 3rd International Workshop on Lightweight Integration on the Web (ComposableWeb 2011)*. Springer, Jun. 2011.

[23] D. Gotz and M. Zhou, "Characterizing users' visual analytic activity for insight provenance," in *Visual Analytics Science and Technology, 2008. VAST '08. IEEE Symposium on*, Oct. 2008, pp. 123 –130.

[24] A. Mitschick, S. Pietschmann, and K. Meißner, "An ontology-based, cross-application context modeling and management service," *Intl. Journal on Semantic Web and Information Systems (IJSWIS)*, Feb. 2010.

[25] W. S. Cleveland and R. McGill, "Graphical perception: Theory, experimentation, and application to the development of graphical methods," *Journal of the American Statistical Association*, vol. 79, no. 387, pp. 531–554, 1984.

[26] D. Lin, "An information-theoretic definition of similarity," in *Proc. 15th International Conf. on Machine Learning*. Morgan Kaufmann, San Francisco, CA, 1998, pp. 296–304.

[27] A. Mitschick and K. Meissner, "Generation and maintenance of semantic metadata for personal multimedia document management," in *Advances in Multimedia, 2009. MMEDIA '09. First International Conference on*, july 2009, pp. 74 – 79.

[28] S. Pietschmann, "A model-driven development process and runtime platform for adaptive composite web applications," *International Journal on Advances in Internet Technology (IntTech)*, vol. 4, no. 1, pp. 277–288, February 2009.

[29] M. Bostock and J. Heer, "Protovis: A graphical toolkit for visualization," *Visualization and Computer Graphics, IEEE Transactions on*, vol. 15, no. 6, pp. 1121 –1128, nov.-dec. 2009.

 ISBN: 978-1-61208-181-6