Towards a Semantics-Based, End-User-Centered Information Visualization Process

Martin Voigt, Stefan Pietschmann, and Klaus Meißner

Technische Universität Dresden,
01062 Dresden, Germany
`{martin.voigt,stefan.pietschmann,klaus.meissner}@tu-dresden.de`

Abstract. Using and understanding Semantic Web data is almost impossible for lay-users as skills in Semantic Web technologies are required. The information visualization (InfoVis) of this data is one possible approach to address this problem. However, convenient solutions are missing as existing tools like Tableau do not support Semantic Web data or users necessitate programming and visualization skills. In this paper, we propose a novel approach towards a generic InfoVis workbench called VizBoard which enables users to visualize arbitrary Semantic Web data without expert skills in Semantic Web Technologies, programming, and visualization. More precisely, we define a semantics-based,user-centered InfoVis workflow and present a corresponding workbench architecture based on the mashup paradigm, which actively supports lay-users in gaining insights from Semantic Web data, thus proving the practicability and validity of our approach.

Keywords: information visualization, Semantic Web data, semantics, workflow, architecture

1 Introduction

With the advent of the Semantic Web technologies like RDF, RDFS, and OWL, more and more organizations publish their information as so-called *Linked Open Data* in the form of open semantic knowledgebases. Unfortunately, gaining insights from this data is mainly reserved to tech-savvy users [5]. To enable end-users to work and profit from Semantic Web data, various browsers have been proposed, including different ontology visualization methods. However, they are usually limited to graph-based data representations and thus do not exploit the functional capabilities of current visual analytic systems, e. g., the support of generic charts, multiple coordinated views, iterative mapping refinement, or the recommendation of appropriate visualizations.

The visualization of Semantic Web data is still in its infancy for two reasons. First, well-established generic InfoVis tools like Tableau[1] do not support Semantic Web data and extensions towards this direction are currently not observable.

[1] `http://www.tableausoftware.com/`

Second, existing concepts, e. g., for SPARQL result set visualization within the *Data-Gov* project [6], require expert users with Semantic Web, programming and visualization skills to visualize the data.

Even without these problems, interpreting and finding the right visualization for a certain data set and goal is a challenging task for lay-users [7], because they lack the necessary visualization knowledge. Thus, knowledge-assisted visualization [4] tries to fill this gap by using formalized expert knowledge and reasoning. While existing concepts are innovative, the solutions presented, e. g., in [10, 20], are domain-specific, self-contained, and not applicable for Semantic Web data.

In this paper, we propose a novel concept for a user-centered InfoVis workflow geared towards lay-users, which allows for the context-aware mapping of arbitrary Semantic Web data to appropriate visualization components. Further, we present our work in progress of its implementation in a visualization workbench called VizBoard. The basic idea of our approach lies in using a formal, yet generic visualization vocabulary to connect all parts of the visualization process. This conceptual foundation is given by the modular VISO ontology (cf. Fig. 1-1) [19], which provides concepts and relations about data, graphics, human activity, user and system context.

Based on these entities, factual expert knowledge can be formalized (cf. Fig. 1-2), e. g., using *position* instead of *color coding* to visualize *quantitative* data, which is used to rank different mapping alternatives. Equally, the user's input data (cf. Fig. 1-3) can be annotated with visualization semantics, e. g., an RDF property *price* may have a *quantitative* scale of measurement and an assigned domain *UnitPriceSpecification* of the GoodRelations ontology[2].

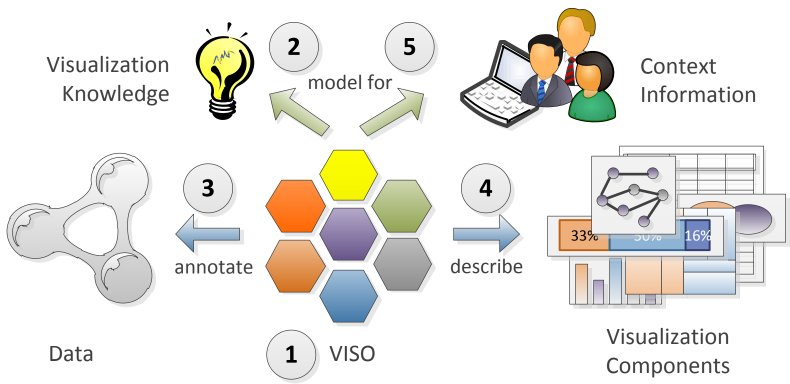

Fig. 1. Generic visualization ontology (VISO) as conceptual foundation of our visualization workflow

[2] http://purl.org/goodrelations/v1

Visualization components (cf. Fig. 1-4) are also described regarding their data interface, kind of graphic representation (map, scatter plot), visual complexity (high, low), and interaction potential (zoom, filter), etc. Finally, the user and system context (cf. Fig. 1-5) are represented, e. g., by *preferences* and user *skills*, the *display size* and the available *software* infrastructure. By using a common vocabulary, all stakeholders and the context of the visualization process can become part of one knowledgebase, which facilitates the context-aware recommendation of visualization components.

Enabling information visualization of Semantic Web data for lay-users using VISO poses a number of challenges. For one, a user-centered process to support lay-users in getting insights from their Semantic Web data sets is required. In this regard, the complexity of the data as well as the tasks of selecting, integrating, and combining visualization components need to be reduced or hidden as much as possible. To enable intelligent system-side assistance in this process, knowledge about the data, visualization, and context needs to be integrated. Furthermore, this knowledge should be continuously improved by reasoning from user interaction. Finally, the process must be implemented in a software architecture which supports each step accordingly, while hiding the complexity of the formal models and their usage, yet being customizable and extensible for different use cases.

Our paper is structured as follows: The next section gives a brief overview of the related work. In Sect. 3, we present our user-centered, semantics-based InfoVis workflow which shows how shared semantics can assist the visualization process. Sect. 4 presents a corresponding workbench architecture, which realizes the process based on the mashup platform CRUISe [15] to illustrate and evaluate its applicability. Finally, we discuss our findings and point out future work in Sect. 5.

2 Related Work

2.1 Understanding and Supporting the Visualization Process

Our vision of a semantics-based InfoVis for lay-users requires a formal knowledge model as mentioned in Sect. 1, but also a structured process defining how to bridge the gap from raw data to an appropriate graphical representation to gain insights. Thus, different visualization-specific process models and work addressing lay-users in InfoVis are discussed briefly.

The pipeline model is most commonly used to describe visualization as a process. In its elementary version it defines a sequence in which raw data is filtered and enriched, mapped to an abstract visualization specification, and finally rendered to a displayable image [8]. This model has been successively enhanced, e. g., to include the user and his tasks [3] or to allow for the coordination of independent views [2]. In contrast to our work, the pipeline model focuses on system-side functionalities and not on the (lay-)user and how he can efficiently reach to insights from his data. Further, it does not support Semantic Web data.

Within the area of knowledge-assisted visualization, several authors have proposed concepts which foster the visualization process by making use of formalized knowledge. Wang et al. [20] give valuable insights of how knowledge "moves" through the visualization process by defining different conversion processes, e. g., externalization of tacit user knowledge to explicit system knowledge. Yet, information on how to employ these processes in generic InfoVis systems to assist users in visualizing semantic models, is missing. Chen et al. [4] sketch a high-level knowledge-based infrastructure in parallel to the visualization system, which extracts information from data and uses it together with predefined expert knowledge to adapt the visualization process. Despite the similar goals, user's interaction steps to gain insights from data and the integration of the formal knowledge in every stage is missing. Thus, we can define semantic-based user support more in detail.

Both, the pipeline model and knowledge-assisted visualization are primarily focusing on how a system can create appropriate visualizations. As our work addresses lay-users, an important orthogonal aspect to consider is user support. The first notable guidelines for supporting InfoVis novices are given by Heer et al. [9]. They advocate easy data input, user assistance in selecting graphical representations, and the use of default mappings from data to visual variables. These are underpinned by a user study in [7]. Therein, Grammel et al. suggest additional guidelines for designing InfoVis tools for lay-users, e. g., to provide (semantics-based) search facilities to narrow the data set, to support the iterative nature of the visualization process, and to deal with partial and uncertain input specifications of lay-users. Finally, Shneidermans mantra [17] defines the most fundamental design guideline for all systems addressing information search: "*Overview first, zoom and filter, then details-on-demand*". This is especially true for lay-users, who need a lightweight overview of the (Semantic Web) data before they dive into details in an iterative way afterwards. In summary, all approaches share our goal of actively supporting users during the InfoVis process by providing valuable advices. However, only Grammel et al. emphasize the power of semantics. More importantly, they all lack a formal process model like the mentioned pipeline.

2.2 Information Visualization of Semantic Web Data

With the growing amount of Semantic Web data sets, more and more methods [11] and tools [5] for their visualization have been proposed focusing mostly on graph-based visualization. Of those, we discuss the few addressing InfoVis by common charts.

And increasing number of US governmental data is made accessible using RDF/XML [6] by the Open Government Directive. In addition, tutorials on their visualization using popular APIs and widget libraries are published[3]. Those imply, that every user has the freedom of building his InfoVis of choice. Unfortunately, the textual tutorial and a proxy for data transformation are little help for lay-users, as Semantic Web, programming, and visualization skills are needed for their use.

[3] `http://data-gov.tw.rpi.edu/wiki/How_to_use_Google_Visualization_API`

The UISPIN framework can be employed to describe user interfaces for rendering Semantic Web data. On top of this, the UISPIN chart library[4] enables the visualization of Semantic Web data using widgets from Google Chart tools and describes these widgets semantically. The library is embedded in Semantic Web tools, such as the TopBraid Composer[5]. Here, a user can include charts without programming skills, but he still needs to define SPARQL queries to specify the data to be visualized. Further assistance, like suggesting appropriate components or an iterative refinement of the results, is missing.

Leida et al. [12] present a solution for annotating SPARQL queries using a shared vocabulary with visualization-specific concepts to (semi-) automatically map RDF data to graphic representations. This promising approach focuses on the mapping. However, a concrete semantic model defining visualization-specific knowledge or a user-centered workflow specifying how a lay-user reaches to his InfoVis is missing.

All in all, current solutions from this field solely focus on the visualization of SPARQL query results. Their common limitation on SELECT statements implies, that graph-based visualizations are usually excluded, even though they are rather suitable and commonly used for Semantic Web data. We share with these concepts the idea of combining arbitrary data sources with existing, web-based widgets from different libraries, following the mashup paradigm. However, and most importantly, prevalent solutions do not support lay-users adequately. Thus, in the next section, we present our solutions towards a user-centered visualization process, which provides the conceptual foundation for our visualization workbench geared towards lay-users.

3 Semantics-based Information Visualization Workflow

To address the problems lined out in Sect. 1 and the deficiencies of related works mentioned in Sect. 2.1, we propose an interactive, user-driven InfoVis workflow (Fig. 2) which builds on the common semantic vocabulary provided by VISO. The latter is independent from concrete data models and architectures.

The workflow design is inspired by the way of how (lay-)users naturally interact when analyzing data and consists of five stages a user needs to pass: choosing or uploading a data set (Fig. 2-1), getting an overview of the data and choosing a subset (Fig. 2-3), selecting relevant data variables and suitable visualization components (Fig. 2-5), configuring them (Fig. 2-7) and, finally, interacting with the rendered data and understanding it (Fig. 2-9). Due to the interactive nature of the visualization process, a user can sequentially pass through, but may also move backwards. For instance, the configuration step can be skipped by using default mappings. Furthermore, he may continue by searching and including alternative visualizations and thus benefit from multiple views of his data after completing a single workflow.

[4] `http://uispin.org/charts.html`
[5] `http://www.topbraidcomposer.com`

This user-driven process is supported by five system-side activities which make use of the VISO (the lower rectangles in Fig. 2). Elementary functionalities like storing, querying, and supplying the data, graphic representations or knowledge are omitted from the figure for the purpose of simplification. In the following, every step of the workflow is discussed in more detail. Additionally, we point to essential requirements for an architecture.

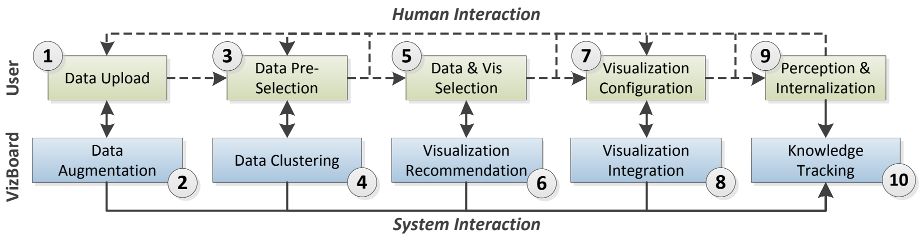

Fig. 2. Overview of the workflow supported by VizBoard

The starting point of every visualization process is the **Data Upload** (Fig. 2-1). That means the user selects a set of raw data and submits it to the system. As we focus on Semantic Web data, the user may upload RDF or OWL files or point to them via their addresses. But it is also possible to allow for other data formats like tabular ones, e. g., a comma-separated file or spreadsheet, or relational data like, e. g., a MySQL database which need to be transformed to RDF by a convenient connector API of the workbench. In the end, a homogeneous data layer, e. g., a RDF triple store, is required to allow for a workbench-wide uniform data access.

After the data is available it needs to be augmented by adding visualization-specific knowledge like the kind of *scale of measurement*, e. g., nominal, ordinal, or quantitative, from VISO. At this, the benefits of the Semantic Web appear as data could be easily annotated with new concepts. This **Data Augmentation** (Fig. 2-2) is the foundation for nearly all following system-side tasks, e. g., the recommendation of appropriate visualization components or their coordination support. This process step is split in two parts: First, an automatic, information-retrieval-driven part adds evident visualization knowledge about the data, e. g., the number of distinct entries of a data variable, and detects also potential enrichments, e. g., domain concepts. Second, the latter are proven and added or declined by the user. As we focus on lay-users, the user part should be optional or guided by the system. To allow for the augmentation, the workbench needs to support from simple quantitative analysis tasks up to named-entity recognition algorithms.

Based on Shneidermans mantra, the purpose of the **Data Pre-Selection** stage (Fig. 2-3) is to give a high-level view of the data structure. Through interactions like zooming, panning, keyword-based searching, or filtering the user is able to find interesting subsets of his data which were selected for an in depth information visualization through suitable components afterwards. A key requirement for the workbench is the intelligent representation of and the convenient interaction with the overview of maybe 3 or 3 million entities so that also a lay-user could efficiently pass this step. Further, it needs to cope with the inherited characteristics of the data structure, e. g., the graph-based nature of RDF.

While working with big data sets its advantageous to assist the user by reducing the number of visual elements. In addition to the interactive reduction of the data during the **Pre-Selection**, we employ a **Clustering** (Fig. 2-4) of the data based on their inherited and enriched semantics as an additional approach. Thus, the workbench needs to employ different algorithms. A common case for Semantic Web data is to subsume entities based on quantitative measures, e. g., the same number of children, instances, or properties. If domain concepts are added by means of the **Data Augmentation** step, entities could be subsumed by their parents, e. g., Toyota and VW are car manufactures. Further, its possible to extract key concepts as proposed in [14].

After the user has narrowed the data set to a region of interest, in the next more detailed **Selection** step (Fig. 2-5) he needs to explore and select interesting data variables from this set. Then, the workbench recommends appropriate visualization components from which he could choose one or more, like the proposed *Show Me* functionality of Tableau does [13]. But we argue that this visualization exploration and selection process needs to be extended for our workbench. First, the user may also search by facets of a graphic representation, e. g., supported visual attributes (color, position), actions (show trend, compare), or the kind of representation (table, map). Second, especially lay-users feel mostly not confident which data variables to select in combination so that a suitable representation could be found (cf. Sect. 2.1). Therefore, a weight assigned to a selection is a proper solution to define what is more or less important for him. Third, the **Data and Visualization Selection** needs to be as dynamic and flexible as possible so that every choice according to the data variables or visualization-specific facets directly influences the list of possible components.

The **Visualization Recommendation** stage (Fig. 2-6) is tightly coupled to users exploration and selection of data and visualizations. Based on his inputs, context information and the semantic descriptions of components, the workbench needs an algorithm and reasoner to calculate a list of visualization components that exactly match the requirements. If the selected data matches a component at the semantically but not at the syntactical level, a mediation rule based on the annotated knowledge is generated. As two or more compatible candidates remain, they need to be rated and ranked based on users preference, e. g., preferring bar charts instead of scatter plots, common visualization knowledge, e. g., for quantitative data position is better than color, and also the weighting given during users selection. All in all, this step is the core of the process which maps data to graphic representations and is proposed in [18].

Users can optionally employ given **Configuration** possibilities (Fig. 2-7), e. g., define filters on instance data or setup the coloring. If different mappings from the data variables to the visualization component are calculated during

Recommendation, the user can concretely choose one of the mappings, e. g., switch the data mapping between the axes of a scatter plot. Further, if one or more components are already integrated, the user has the ability to link and thus, to coordinate them, e. g., selected data in one view is highlighted in others. This interactive behaviour strongly improves the understanding of the displayed information. A key requirement for the workbench is to establish a helpful coordination between a set of independent visualization components.

Before the user can work with the chosen and configured visualization components, they need to be **integrated** (Fig. 2-8). Therefore, the workbench needs to fulfil the following requirements. Components need to be loaded from the provider, configured by the setup properties from the recommendation respectively users configuration, and finally, initialized by passing the selected data. If the data does not match at the syntactical level, the workbench needs to employ the mediation rules to allow for conversion. This mediation is also required for the communication between the components on runtime.

After **Integration**, the user can use and **perceive** the visualization of the data and increase his knowledge, called **internalization** [20] (Fig. 2-9). As stated by van Wijk [21], the amount of the gained knowledge depends on the visual representation, users prior knowledge as well as his perceptional and cognitive capabilities. To adapt the internalization process, we need to differentiate between the functionality of the workbench and the used components. Thus, the appropriateness for internalization depends partially on the components as they may or may not allow for to zoom, filter, or highlight data. The workbench could only supply applicable components due to the semantics used via the recommendation step.

The **Knowledge Tracking** (Fig. 2-10) works mostly behind the scene during the complete visualization process. Its purpose is the extraction of visualization knowledge for its usage in following workflows. The workbench needs tracking “sensors” to collect information about the user, e. g., preferred representation or often used data, his hard- and software context but also the visualization process itself, e. g., often used data-visualization combinations. Besides the automated tracking the user could give explicit feedback about his satisfaction of the result of the process.

4 Architecture of VizBoard

In the following, we present the service-oriented architecture of our workbench specified based on the requirements which come along with our user-centered information visualization workflow. Like the discussed approaches in Sect. 2.2, we build on the mashup paradigm which strives for combining existing web resources to create a new value [1]. According to this, we combine Semantic Web data with visualization components from various libraries to assist the process of insight discovery respectively internalization. As it is not our goal to create yet another mashup platform we build on CRUISe [15]. Thus, we benefit from a homogeneous component model and description, a component repository, a model to compose an application, and a client-side runtime environment which are all extended according to our requirements. Fig. 3 gives an overview of the architecture of VizBoard. It comprises five primary parts which are discussed in more detail in the following.

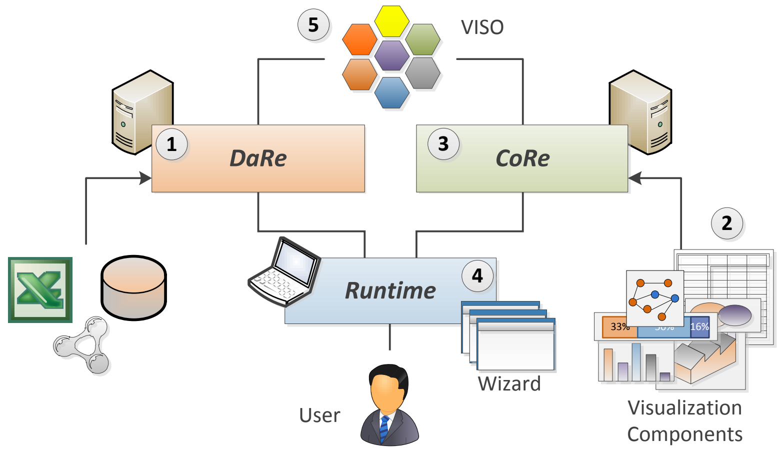

Fig. 3. Overview of the architecture of our workbench.

The **Data Repository** (DaRe) (Fig. 3-1) adds a common data layer to the CRUISe platform. It mainly follows the requirements of the first four process steps (Fig. 2). First, all uploaded data is organized in a user-specific namespace and could be queried as a SPARQL endpoint via SOAP or REST. As users may also wish to visualize data from spreadsheets or databases, we add an adapter API to allow for new data formats which are internally converted to RDF. Another key functionality is the mentioned semi-automatic **Data Augmentation** with VISO vocabulary using quantitative analysis methods as well as named-entity recognition algorithms. To reduce the number of visual elements during the **Data Pre-Selection**, the **DaRe** makes use of existing clustering methods.

Visualization Components (Fig. 3-2) are specialized user interface components focusing on presentation but not on the "*CRUD*" functions. Thus, we can rely on the component model of CRUISe without any difficulties. We only extend the *Semantic Mashup Component Description* (SMCD) [16] to describe visualization features with VISO concepts, e. g., the kind of graphic representation, and to specify the underlying data model, e. g., RDF, XML, or JSON. As the parameters of operations and events are already semantically typed, we can simply point to their generic, semantics-based description of the data structure (more information in [18]).

Further, we use and extend the **Component Repository** (CoRe) (Fig. 3-3) to allow for the semantics-based management of visualization components based on their SMCDs. Here, the recommendation for appropriate components

(Fig. 2-6) is implemented as a multi-level process comprising the discovery and ranking [18]. Furthermore, we added the **Knowledge Tracking** (Fig. 2-10) to the **CoRe** as its results are mostly used during the recommendation. To this, we added service methods to consume implicit and explicit ratings tracked by the **Runtime** (Fig. 3-4).

The **Runtime** (Fig. 3-4) provides the interface between the user, **DaRe**, and **CoRe**. Therefore, the complete interactive workflow of the user (the green rectangles in Fig. 2) is implemented as wizard comprising specialized and thus, efficient user interface components for each stage. An example is a faceted browser for the **Data and Visualization Selection** (Fig. 2-5). As already part of CRUISe, the **Runtime** provides the component integration, the event-based communication between components as well as the context tracking. To enable lay-users to create, edit, and delete communication connections between visualization components, we added a more abstract coordination layer and a helpful meta visualization to show existing communication relations. Further, the runtime is extended to handle RDF data received from the **DaRe** as shared data layer for all components.

As stated in our conceptual foundation (cf. Sect. 1), the visualization-specific vocabulary defined by the **VISO** (Fig. 3-5) is the glue between the data and visualization part. To this, the **DaRe** with the data and the **CoRe** with its components use its concepts to describe their managed entities as well as to discover and rank appropriate mapping between both.

5 Discussion and Future Work

Gaining insights from the growing amount of available Semantic Web data has become seemingly impossible for lay-users. To address this need for user-centered information visualization, we have propose three ingredients: (1) a semantic model formalizing visualization knowledge, (2) a user-centered visualization workflow utilizing the shared visualization model, and (3) a corresponding system architecture implementing the workflow. While the model was published in [18], this paper has focused on the workflow and its application.

In contrast to existing workflows, e. g, the pipeline model, the visualization process presented actively guides lay-users from a given set of semantic input data to suitable visualization components using the shared visualization knowledge. Even though the steps are described as detailed as possible, they are generic enough to be implemented by other tools and frameworks. It should be noted, that the process remains independent from the underlying data models and can thus be employed for arbitrary, not necessarily Semantic Web data. This facilitates evaluation and refinement through implementations by the community.

As a manifestation of the concepts, we have presented an architecture which implements the workflow and utilizes VISO as the semantic model. To this end, we employed the mashup paradigm whose goal is the combination of existing web resources – in our case RDF data and InfoVis widgets – to create an added value for the user. The architecture is easily extensible with new visualization components but also new data connectors. As it is web-based and includes context knowledge in the composition process, it can be utilized on different devices, such as desktops, tablets, and smartphones, independent of location and time.

As mentioned before, this paper provides an overview of our work towards a user-centered InfoVis workflow and its implementation in an extensible, open workbench. While concept's core – the *recommendation* of suitable visualization components – has already been validated [18], and large parts of the workbench have been realized based on the CRUISe platform [15], a few things remain to be done. Currently, we are specifying and implementing the *Data Augmentation*, the *Data and Visualization Selection*, the *Configuration* with regard to coordinating multiple views within mashups, and the *Knowledge Tracking* of our workflow in more detail. Future work is targeted towards concepts for the *Data Pre-Selection* in combination with the *Clustering* for Semantic Web data. Further, we are planning to conduct a user study for the evaluation of the overall workflow with a real-world scenario.

Acknowledgements

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 Annett Mitschick for helpful editing suggestions.

References

1. Benslimane, D., Dustdar, S., Sheth, A.: Services mashups: The new generation of web applications. Internet Computing, IEEE 12(5), 13 –15 (2008)
2. Boukhelifa, N., Roberts, J., Rodgers, P.: A coordination model for exploratory multiview visualization. In: Coordinated and Multiple Views in Exploratory Visualization. pp. 76–85 (2003)
3. Card, S.K., Mackinlay, J.D., Shneiderman, B.: Readings in information visualization: using vision to think. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA (1999)
4. Chen, M., Ebert, D., Hagen, H., Laramee, R., van Liere, R., Ma, K.L., Ribarsky, W., Scheuermann, G., Silver, D.: Data, information, and knowledge in visualization. Computer Graphics and Applications, IEEE 29(1), 12 –19 (2009)
5. Dadzie, A.S., Rowe, M.: Approaches to visualising linked data: A survey. Semantic Web 2(1) (Jun 2011)
6. Ding, L., DiFranzo, D., Graves, A., Michaelis, J., Li, X., McGuinness, D.L., Hendler, J.A.: Twc data-gov corpus: incrementally generating linked government data from data.gov. In: WWW'10. pp. 1383–1386 (2010)
7. Grammel, L., Tory, M., Storey, M.A.: How information visualization novices construct visualizations. IEEE Trans. on Visualization and Computer Graphics 16, 943–952 (November 2010)
8. Haber, R., McNabb, D.A.: Visualization idioms: A conceptual model for scientific visualization systems. Visualization in Scientific Computing pp. 74–93 (1990)

9. Heer, J., van Ham, F., Carpendale, S., Weaver, C., Isenberg, P.: Creation and Collaboration: Engaging New Audiences for Information Visualization, pp. 92–133. Springer-Verlag, Berlin, Heidelberg (2008)
10. Kadlec, B., Tufo, H., Dorn, G.: Knowledge-assisted visualization and segmentation of geologic features. Computer Graphics and Applications, IEEE 30(1), 30 –39 (jan-feb 2010)
11. Katifori, A., Halatsis, C., Lepouras, G., Vassilakis, C., Giannopoulou, E.: Ontology visualization methods—a survey. ACM Comput. Surv. 39(4), 10 (2007)
12. Leida, M., Afzal, A., Majeed, B.: Outlines for dynamic visualization of semantic web data. In: On the Move to Meaningful Internet Systems: OTM 2010 Workshops, vol. 6428, pp. 170–179. Springer Berlin / Heidelberg (2010)
13. Mackinlay, J., Hanrahan, P., Stolte, C.: Show me: Automatic presentation for visual analysis. IEEE Transactions on Visualization and Computer Graphics 13(6), 1137–1144 (Nov 2007)
14. Peroni, S., Motta, E., dAquin, M.: Identifying key concepts in an ontology, through the integration of cognitive principles with statistical and topological measures. In: The Semantic Web, vol. 5367, pp. 242–256. Springer Berlin / Heidelberg (2008)
15. Pietschmann, S.: A model-driven development process and runtime platform for adaptive composite web applications. Int. Journal on Advances in Internet Technology 4(1), 277–288 (2009)
16. Pietschmann, S., Radeck, C., Meißner, K.: Semantics-based discovery, selection and mediation for presentation-oriented mashups. In: Proceedings of the 5th International Workshop on Web APIs and Service Mashups. ACM ICPS, ACM (Sep 2011)
17. Shneiderman, B.: The eyes have it: a task by data type taxonomy for information visualizations. In: Proc. of IEEE Symp. on Visual Languages. pp. 336–343 (1996)
18. Voigt, M., Pietschmann, S., Grammel, L., Meißner, K.: Context-aware recommendation of visualization components. In: Proc. of the 4th International Conference on Information, Process, and Knowledge Management (eKNOW 2012). XPS (Feb 2012)
19. Voigt, M., Polowinski, J.: Towards a unifying visualization ontology. Tech. Report TUD-FI11-01, TU Dresden, Dresden, Germany (Mar 2011), ISSN: 1430-211X
20. Wang, X., Jeong, D.H., Dou, W., Lee, S.W., Ribarsky, W., Chang, R.: Defining and applying knowledge conversion processes to a visual analytics system. Computers & Graphics 33(5), 616 – 623 (2009)
21. van Wijk, J.J.: The value of visualization. In: Proceedings of IEEE Visualization. pp. 79–86. Minneapolis, MN (October 2005)