Weighted Faceted Browsing for Characteristics-Based Visualization Selection through End Users

Martin Voigt, Artur Werstler, Jan Polowinski, and Klaus Meißner
TU Dresden
01062, Dresden, Germany
{martin.voigt, artur.werstler, jan.polowinski, klaus.meissner}@tu-dresden.de

ABSTRACT
Faceted browsing is a widely spread, intuitive, and interactive search paradigm for information collections based on the metadata of its items. However, it has the problem that every selected criterion is mandatory so that less important ones may reduce the result set and interesting items may be removed unintentionally. On the other hand, choosing only very few facets yields to an unmanageable set of items wherein the best ones do not become obvious. In this paper, we propose *weighted faceted browsing*, which seamlessly extends the existing faceted browsing paradigm. Besides basic filtering capabilities, it provides a sophisticated relevance ranking of the result set based on the distinction between mandatory and weighted optional search criteria. Further, we show its practicability within an information visualization workbench to facilitate the end user's search for visualization components based on their characteristics.

Author Keywords
faceted browsing; query building; weighted search; filtering; visual analysis; end user

ACM Classification Keywords
H.5.2 Information Interface and Presentation: Graphical user interfaces (GUI); H.4.3 Communications Applications: Information browsers

INTRODUCTION
Today, information search is a crucial task in people's life, thus, new search algorithms but also user interfaces (UI) are evolving continuously [6]. In this context, faceted browsing – or faceted search – evolved as a highly usable paradigm to filter and navigate information collections based on the characteristics of its items [22]. For example, e-commerce platforms like amazon or eBay successfully implemented faceted browsing to allow for a flexible and uncomplicated product search also for novices.

Also the procedure of information visualization (InfoVis) is a search process which aims at finding the best presentation for a data set. This is especially challenging for end-users, as their lack of InfoVis knowledge leads to unsatisfying visualizations [5]. In order to assist the visual mapping, systems should suggest appropriate visualizations based on data attributes and other characteristics like the kind of graphic representation or the representation goal. The following two examples which emphasize different kinds of characteristics illustrate users thoughts on finding visual representations: "I'd like to visualize music events – *preferably* taking place in my home town – using a map." or "I'd like to compare interactively the number of performers according their genre, *at best* using a bar chart or – *less preferred* – a scatter plot."

Faceted browsing is a suitable approach to assist end-users in searching for visual representations for two reasons: First, users filter the collection by selecting only from existing characteristics of the remaining items. This is easier to handle than a free, textual query specification but foremost it assures not to deliver an empty result set. Second, InfoVis novices often create partial query specifications [5], e. g., by choosing one data attribute after the other to see the result. Thus, they immediately need feedback to refine their query iteratively.

Within our project VizBoard we strive for a semantics-based, end-user-centered InfoVis process [21]. As a first step, we have already developed and validated an algorithm that recommends appropriate visualization components for arbitrary Semantic Web data [20]. Unfortunately, a suitable, intuitive UI which would allow for an easy creation of search queries to "feed" the algorithm is missing. Hence, the faceted browsing paradigm comes into play.

However, with regard to the example queries mentioned above, there are problems for implementing them with existing faceted browsers. First, it is not possible to assign priorities to facets within a search query. Second, every selected facet is a mandatory search criterion. This is crucial, because facets, which are meant to be optional, strip down the result set to a great extent to maybe less relevant items. On the other hand, specifying only a few criteria causes a broad, confusing result set, where the most interesting items are not obvious to the user. In this case, ranking criteria provided by the user would be helpful but existing approaches are limited to basic sorting like the alphabetical one.

Targeting the outlined problems, the contributions of this paper are twofold: First, we propose a seamless extension of the faceted browsing paradigm, the so-called *weighted faceted browsing* (WFB). It supports the well-known filtering capabilities but adds a sophisticated ranking mechanism using facets in combination with a weight. Further, these concepts are applied to an intuitive UI. Second, we demonstrate the practicability of WFB for visualization selection within an InfoVis workbench which allows for iterative query specification using mandatory and optional weighted search criteria.

RELATED WORK

The weighted faceted browser presented in this paper builds on previous work in the three different research areas (1) faceted and (2) weighted search as well as (3) approaches to search for visualizations. We will now discuss the state of the art in these three areas.

Faceted Browsing

Faceted browsing [22] is a widely applied search paradigm, e. g., in e-commerce (amazon or eBay) or document collections (ACM digital library or IEEE Xplore), based on the characteristics of the items within a collection. In general, a faceted browser consists of three parts: a list of widgets, presenting the facets, a visualization of the result set, and the representation of the query. Existing research in this domain concentrated on providing more sophisticated facet widgets [3, 15] or more efficient browser layouts [19, 8]. However, improvements with regard to the representation of the result set have been neglected so far. Indeed, a user may specify a grouping or sorting for the items. But this is limited to single, basic, and predefined attributes like the alphabetical order. Sorting based on combinations of item characteristics or assigning a relevance to each item is not possible.

Weighted Search

Weighted search is a matured approach [16] especially in the area of multimedia database systems [18] to allow for – in contrast to weights assigned during indexing – a user driven relevance ranking. Besides the explicit specification of quantitative or relative weightings it is possible to define them implicitly [18]. For example, the high-performance search engine Lucene [1] allows for an explicit, quantitative query term weighting using its *boost* factor, e. g., event^5 map^5 place^1. However, it is currently not possible to distinguish between mandatory and optional criteria.

As outlined in the section above, faceted browsing does not support relevance ranking of the result set using the facets. An approach towards this direction illustrates the online radio service Musicovery comprising the mood pad [2]. The latter allows for choosing the mood of songs – from energetic to calm as well as from dark to positive, in a Cartesian coordinate system. By picking a point in the reference system, the user starts a new search and assigns a weight to the two mood scales implicitly. Nevertheless, it is not possible to weight other or additional characteristics of the items.

Visualization Selection for End Users

InfoVis tools provide different levels of user support to find appropriate visualizations [4]. One of the most sophisticated is Tableau, comprising the "Show Me" mechanism which suggests graphic representations for the selected data variables [12]. However, it is not possible to search visualizations based on their characteristics and representation goals or to distinguish between mandatory and optional data variables.

Further, many online galleries like visualcomplexity.com or visual.ly emerged in the last years, providing huge collections of graphic representations. Unfortunately, here the items can only be filtered and sorted using categories or basic attributes like "most viewed". [11] proposes a characteristics-based classification for visualization using the metaphor of the periodic table, but an interactive tool implementing this approach is missing. The gap of interactive filtering visualizations based on their attributes is partly bridged by DelViz [10]. Compared to DelViz, we additionally provide features for relevance ranking as well as an integration within a visualization workbench.

WEIGHTED FACETED BROWSING

In this section, we describe the general functionality of weighted faceted browsing, including the division into mandatory criteria and weighted optional criteria, the relevance ranking based on them, and finally, the interactive UI of the weighted faceted browser.

Mandatory and Weighted Optional Criteria

In order to integrate sophisticated relevance ranking seamlessly with the faceted browsing paradigm, the query which constraints the result set, is still created by incrementally selecting facet values. But now, the user can choose between mandatory and optional criteria. To narrow the results, a facet value needs to be added to the set M where all criteria are linked conjunctively – which is the standard behavior of a faceted browser. In contrast, optional facets, which are combined disjunctively within a set O, do not constrain but rank the results. Thus, the more optional criteria an item satisfies the higher is its rank.

However, if some items meet the same number of optional criteria, their ranking would be the same. Further, our second example stating that "*bar charts* are preferred to *scatter plots*" emphasizes that users also need to define priorities between single facets. To achieve this, two distinct approaches could be used. First, the weight could be set relatively to other facet values, e. g., $f_1 < f_2 \leq f_3$, where the system needs to assign quantitative values implicitly at query interpretation. As it would be complicated – especially in the UI – to define long criteria chains or the distance between two criteria, we applied a second strategy and added an explicit weight to every criterion using a quantitative scale between 1 and 100. We neglect 0 as this could suggest that the facet could be omitted. Further, mandatory criteria are not weighted as they need to be fully satisfied and not only partly.

The complete query can be represented by means of the following formula. Taking into account that the optional criteria should not constrain the result set if no item satisfies any criterion, we add a ε to O so that this set will always be *true*. This behavior could simply be skipped by removing ε again.

$$Q = \bigwedge_{x} \wedge \bigvee_{(y,w)\vee\varepsilon} |\forall x \in M, \forall y \in O, \forall w \in [1, 100]$$

The seamless integration of weighted, optional criteria with faceted browsing requires transforming mandatory criteria to optional ones and vice versa. In the first case, the criterion does not restrict the result set anymore but may affect its order. This navigation step, called a *zoom-out* in regular faceted browsing [17], is especially interesting in combination with a high weight. In this case, the facet value has a high impact on the ranking but does not result in items being hidden that are good candidates with respect to other criteria. For the opposite direction, moving a facet from the set O to M, being a *zoom-in* navigation step, narrows the result set and may outrange selected optional criteria as no item complies with them. As the feature is crucial for a comprehensible interaction, our compromise is to give the user visual feedback whether a facet is supported or not.

Result Ranking

The input for calculating the result set is a query in terms of the formula proposed above. While the mandatory part simply constraints the set by removing all items not supporting a chosen facet value, the relevance ranking using multiple optional facets is more complicated. To solve the multiple-criteria optimization, we can rely on an extensive set of related work, e. g., the survey proposed in [13]. Our ranking problem can be categorized as "a priori" because the user already chose the goals and weights. In this field, we can basically distinguish two opposed strategies to interpret the optional, weighted criteria:

1. The user strives for a result which satisfies all goals as much as possible. Hence, the weighting method (or weighted sum model) is applicable.
2. The criterion with the highest weight is the users primary goal and is to be preferred to others. This strategy is implemented by the *lexicographic ordering*.

We employ *both* strategies, which is a common use case [13], in an iterative way. We are optimizing by using the weighting method at first. This method considers items comprising more but possibly less important criteria and, thus, does not neglect the low weighted facets the user has selected. Afterwards, if some items share the same weight, we apply the lexicographic ordering. Finally, we order elements alphabetically if they still have the same weight.

User Interface for Weighted Faceted Browsing

After having discussed the background concepts of weighted faceted browsing, in the following, we will introduce the UI and interaction concept. Its implementation is illustrated in Fig. 1. Basically, the layout corresponds to classical faceted browsing and is split into three main areas: facet widgets at the top part ① ②, the query visualization, called *querycloud*, in the middle ③, and the results view at the bottom part ④ ⑤. To create a query, the user simply needs to drag a desired facet value and drop it at the querycloud. The result set visualization ④ updates subsequently. By selecting an item from the list, detailed information are displayed in ⑤. In the following, we will outline these parts in more detail.

Facet Widgets

The purpose of the facet widgets is twofold: they represent the characteristics of the items per facet and offer these facet values for selection. For this purpose, the browser can dynamically integrate the widgets, usually list widgets, defined in a declarative configuration file. Further, we distinguish direct and indirect facets, similar to [7]. Whereas the first are directly associated with the items metadata, indirect facets are characteristics which are calculated or referred to based on semantic links. In our scenario of searching appropriate visualization components, the indirect data facet ① is calculated at runtime using the algorithm proposed in [20].

In order to present the semantic data in a appropriate way for users, who may not have prior semantic web knowledge, we rely on the simple but well-known indented list metaphor [9]. Therefore, we display the classes, which can be expanded to show datatype properties, at the top level. These properties are linking literals, e. g., *performer* or *genre label* in Fig. 1, and could be dragged to the querycloud. Object properties are visualized implicitly using hierarchies and its associated class, e. g., an *event* is linked to a *location* and *time*.

The widgets provide different kinds of visual feedback to assist the user. First, a number behind every datatype property and facet value shows the number of items corresponding to this characteristic and is updated after every interaction. Second, after applying a facet value to the querycloud, its color adapts to orange (mandatory) or blue (optional). Third, if a facet value is not applicable due to restrictions on the result set, i. e., its item number is zero, the facet value is removed from the widget to raise the clearness.

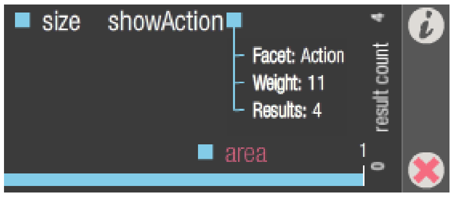
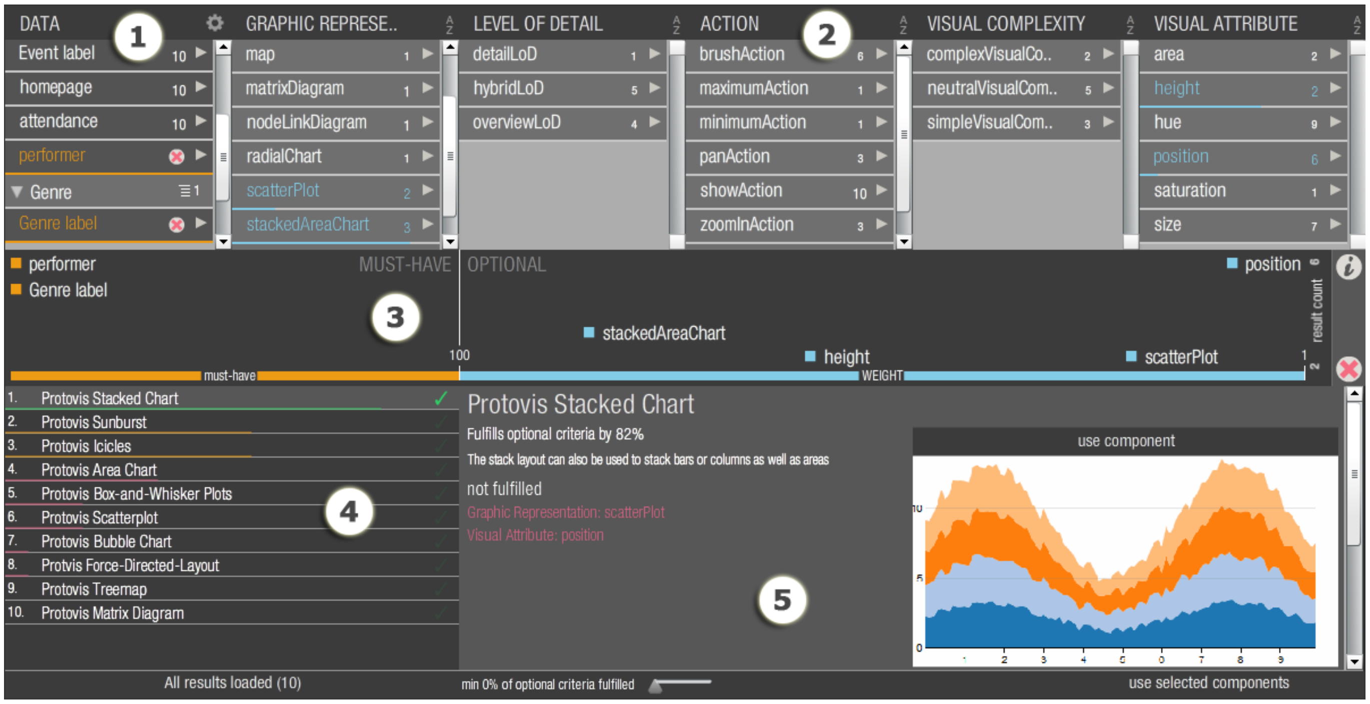


Figure 2. Details of the querycloud

Querycloud

The key capabilities of the querycloud ③ are the query configuration and its visualization. As we distinguish between mandatory and optional criteria, the cloud is split into two areas where the user can drop the chosen facet values. Within the *hot area* (orange), all must-have criteria are simple visualized as a list using the order the user dropped them. The capabilities of the cold area (blue), which represents the optional criteria, are more advanced. While dropping a facet, the horizontal position is used to assign a weight to the facet. Instead of defining a value explicitly, e. g., by using an input field, we consider that this behavior is precise enough



1. Protovis Stacked Chart ✓
2. Protovis Sunburst
3. Protovis Icicles
4. Protovis Area Chart
5. Protovis Box-and-Whisker Plots
6. Protovis Scatterplot
7. Protovis Bubble Chart
8. Protvis Force-Directed-Layout
9. Protovis Treemap
10. Protovis Matrix Diagram

Protovis Stacked Chart

Fulfills optional criteria by 82%

The stack layout can also be used to stack bars or columns as well as areas

not fulfilled

Graphic Representation: scatterPlot

Visual Attribute: position

use component

All results loaded (10)

min 0% of optional criteria fulfilled

use selected components

Figure 1. Overview of the weighted faceted browser

while being more uncomplicated and intuitive. The weight can be adjusted by moving the facet value on the axis. To remove any facet, the user simply needs to drop them out of the querycloud.

The vertical axis in the cold area is used to show the number of items comprising the facet value. The range is represented at the right of the area (Fig. 2). Further, as Fig. 2 illustrates for "showAction", detailed information of a criterion is given on hovering any criterion in the querycloud. This includes its facet, the assigned weight, and the number of items using the characteristic.

As discussed above, the browser needs to support the transformation of mandatory criteria to optional and vice versa. For this reason, the user may also drag and drop values between both areas of the query cloud. On dropping, all facets widgets as well as the position of the optional criteria are updated subsequently. By turning an optional criterion into a mandatory, other optionals may not be comprised by any item of the result set any longer. Hence, these facets are marked red, like the "area" facet in Fig. 2, to call the users attention.

Result Set

The result set visualization at the bottom part is split as well: ④ comprises a ranked list of items whereas ⑤ provides detailed information of the selected item. To visually emphasize the relevance and to allow for a comparison of the items at first sight, the labels are underlined using colored lines. The ranking value is mapped to the color – from green (high) to red (low) – and redundantly to the length of the lines. The representation of the detail view depends at most on the metadata provided by the items. In our case, visualization components metadata like a screenshot and a textual description are shown. Additionally, the view represents the percentage of fulfilling the optional criteria as well as a list of optional facets which the item does not comprise.

VISUALIZATION SELECTION AS EVALUATION DOMAIN

In the following, we outline an implementation of WFB. Its integration into our InfoVis workbench VizBoard is the foundation of a preliminary user study whose results are discussed afterwards.

Implementation

The weighted faceted browsing concept was realized as essential part of the VizBoard InfoVis workbench [21] which relies on the CRUISe mashup platform [14]. Fig. 3 gives a brief architectural overview of its integration. The WFB is divided into a backend service (Fig. 3-4) and a frontend mashup component (Fig. 3-5) to counter performance issues during filtering or facet calculation. The latter, which is implemented using Adobe Flex and JavaScript, is dynamically integrated at runtime and provides the UI introduced above. The facets, its direct or indirect nature, and their appearance and layout within the browser is described declaratively to foster an easy adaptation, e. g., for other domains.

A REST-based web service interface enables the frontend to retrieve the data, e. g., the facets and their values, from the backend as well as to send the query created by the user. The WFB backend, implemented using Java and Jersey for the web service communication, handles a series of tasks. At first, it retrieves the Semantic Web data – narrowed in a forgoing pre-selection step [21] – from the data repository (Fig. 3-1) to construct the indirect data facet (Fig. 1-1). Subsequently, it initially calls the component repository (Fig. 3-2) using the

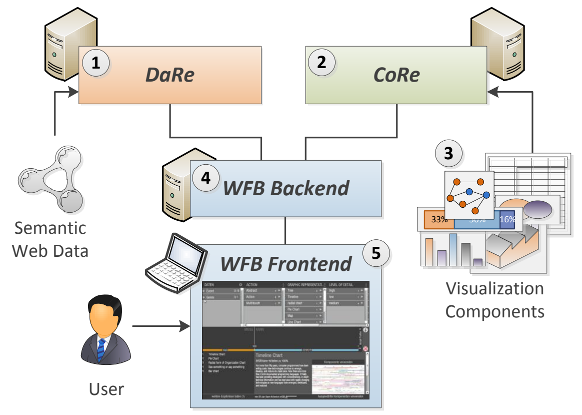


Figure 3. Overview of the software architecture and its integration into VizBoard

data to get a list of suitable visualization components (Fig. 3-3) including their metadata. The recommendation algorithm proposed in [20] determines the components which are able to visualize at least one data variable of the data set. Thereafter, the backend creates the direct facets and its values based on the metadata (Fig. 1-2). After the user has modified the query, the backend recalculates the facets as well as the result set. If the query comprises values from the data facet, again the recommendation algorithm is called to narrow the component list. Otherwise, the backend can directly process the direct facet values.

A screencast as well as a stand-alone live demo is available on `http://cruisedemos.dyndns.org/wfb/`.

Evaluation

We conducted a preliminary user study, consisting of three parts, to prove our novel concept of weighted faceted browsing in the area of visualization selection. Five students, who are familiar with faceted browsing, InfoVis tools like MS Excel or Tableau but never worked with Semantic Web data, participated. In the first part, we familiarized the users with the scenario and WFB by providing a screencast illustrating the main functionalities. Next, they got 2 minutes to explore the browser individually. In the second part, the user had to handle five basic tasks like selecting or removing facets. But, more interestingly, they had to solve five more advanced search tasks – similar to the examples mentioned in the introduction – to find appropriate visualization components. In the third part, we asked some questions related to usability issues.

The subjects answered all questions correctly. Whereas the basic tasks were solved without any help, we needed to give some assistance at solving the advanced issues. This was mostly caused by the missing understanding of the data set and the metadata of the visualization components within the facet widgets. Exemplary questions were: "Why is the *genre* shown twice?" or "What does *neutral visual complexity* mean?". Thus, we need to improve the data widget and should provide additional information for the facet values. Further, the users stated they enjoyed the intuitive approach of the querycloud. Assigning a weight or transforming criteria from mandatory to optional and vice versa seems to be convenient. Two users suggested to give a live preview of the weight while dragging a facet over the "cold" area. Further, the participants comprehended the result view as well. On asking why they choose two particular components, they answered: "They are the first and had a great distance to the next results.".

CONCLUSION AND FURTHER WORK

This paper presents a seamless extension of the well-known faceted browsing paradigm which allows for sophisticated but intuitive relevance ranking of the result set. This is – amongst others – worthwhile if a vast quantity of items in the result set hinders to identify an appropriate one. We achieved this objective by 1) distinguishing between mandatory and optional criteria, 2) assigning priorities to optional criteria, 3) implementing a ranking method, and 4) developing a self-explanatory UI.

Furthermore, we successfully implemented our approach and proved its practicability within the InfoVis workbench VizBoard in order to facilitate end users in searching for visualization components based on their characteristics. A preliminary user study suggests that users are confident with the weighted faceted browsing and are able to express advanced search queries.

As one result of our evaluation, we are currently working on the indirect facet widget to represent the relations of the small Semantic Web dataset in a more comprehensible way. Further, we are currently discussing how to foster the understanding of single facet values by using additional visuals without losing clarity. Finally, we will verify the WFP including the relevance ranking strategy within other domains, e. g., news and media search.

ACKNOWLEDGMENTS

Work on this paper is funded within the Mefisto project by the German Federal Ministry of Education and Research under promotional reference number 01IA09001-C and also received financial support from the European Social Fond / Free State of Saxony, contract no. 80937064.

REFERENCES

1. Apache Software Foundation. Apache Lucene, 2012.
2. Castaignet, V., and Vavrille, F. Musicovery.com, 2012.
3. Dachselt, R., Frisch, M., and Weiland, M. Facetzoom: a continuous multi-scale widget for navigating hierarchical metadata. In *Proc. of the 26th conf. on Human factors in computing systems (CHI'08)* (2008), 1353–1356.
4. Gilson, O., Silva, N., Grant, P., and Chen, M. From web data to visualization via ontology mapping. In *Computer Graphics Forum*, vol. 27 (2008), 959–966.
5. Grammel, L., Tory, M., and Storey, M.-A. How information visualization novices construct visualizations. In *Proc. InfoVis 2010* (2010).

6. Hearst, M. A. *Search User Interfaces*. Cambridge University Press, 2009.

7. Heim, P., Ertl, T., and Ziegler, J. Facet graphs: Complex semantic querying made easy. In *The Semantic Web: Research and Applications*, vol. 6088. 2010, 288–302.

8. Huynh, D., and Karger, D. Parallax and companion: set-based browsing for the dataweb, 2009.

9. Katifori, A., Halatsis, C., Lepouras, G., Vassilakis, C., and Giannopoulou, E. Ontology visualization methods - a survey. *ACM Comput. Surv. 39* (2007), 10.

10. Keck, M., Kammer, D., Iwan, R., Taranko, S., and Groh, R. Delviz: Exploration of tagged information visualizations. In *Informatik 2011 - Interaktion und Visualisierung im Daten-Web* (Berlin, 2011).

11. Lengler, R., and Eppler, M. J. Towards a periodic table of visualization methods for management. In *Proc. of the Conf. on Graphics and Visualization in Engineering (GVE 2007)* (2007), 1–6.

12. Mackinlay, J., Hanrahan, P., and Stolte, C. Show me: Automatic presentation for visual analysis. *IEEE Transactions on Visualization and Computer Graphics 13*, 6 (Nov. 2007), 1137–1144.

13. Miettinen, K. Some methods for nonlinear multi-objective optimization. In *Evolutionary Multi-Criterion Optimization*, vol. 1993. 2001, 1–20.

14. Pietschmann, S., Nestler, T., and Daniel, F. Application composition at the presentation layer: Alternatives and open issues. In *Proc. of the 12th Intl. Conf. on Information Integration and Web-based Applications & Services (iiWAS 2010)*, ACM (2010).

15. Polowinski, J. Widgets for faceted browsing. In *Human Interface and the Management of Information. Designing Information Environments*, vol. 5617. 2009, 601–610.

16. Robertson, S. E., and Jones, K. S. Relevance weighting of search terms. *Journal of the American Society for Information Science 27*, 3 (1976), 129–146.

17. Sacco, G. M., and Tzitzikas, Y. *Dynamic Taxonomies and Faceted Search: Theory, Practice, and Experience.* Springer, New York, 2009.

18. Schmitt, I., Schulz, N., and Saake, G. Multi-level weighting in multimedia retrieval systems. In *XML-Based Data Management and Multimedia Engineering EDBT 2002 Workshops*, vol. 2490. 2002, 524–528.

19. Tvarožek, M., and Bieliková, M. Collaborative multi-paradigm exploratory search. In *Proc. of the Hypertext 2008 workshop on Collaboration and collective intelligence* (2008), 29–33.

20. Voigt, M., Pietschmann, S., Grammel, L., and Meißner, K. Context-aware recommendation of visualization components. In *Proc. of the 4th Intern. Conf. on Information, Process, and Knowledge Management (eKNOW '12)* (2012).

21. Voigt, M., Pietschmann, S., and Meißner, K. Towards a semantics-based, end-user-centered information visualization process. In *Proc. of the 3rd Intern. Workshop on Semantic Models for Adaptive Interacive Systems (SEMAIS '12)* (2012).

22. Yee, K.-P., Swearingen, K., Li, K., and Hearst, M. Faceted metadata for image search and browsing. In *Proc. of the Conf. on Human factors in computing systems (CHI '03)* (2003), 401–408.