Widgets for Faceted Browsing

Jan Polowinski

Technische Universität Dresden, Nöthnitzerstr. 46, 01187 Dresden, Germany
jan.polowinski@tu-dresden.de

Abstract. Faceted Browsing, also known as Faceted Search, recently became a popular interface paradigm and is used as a visual selection mechanism to browse data collections. It is particularly suitable for structured, but heterogeneous data with explicit semantics. As the amount of this data will grow in respect to a rising Semantic Web, it is worth looking at the current state of Faceted Browsers. In this paper we extract characteristics of several browsers from various domains. We then propose the use of advanced widgets for Faceted Browsers and how to select appropriate widgets.

Keywords: Faceted Browsing, Faceted Search, Dynamic Taxonomies, User Interface, Semantic Web, Facet Widgets

1 Introduction

A Faceted Browser can be used to explore structured data sources such as media collections with annotated metadata, data stored as part of a Semantic Web [1] or data from Semantic Desktops [8]. Faceted Browsing can also be seen as a means to construct complex queries without writing them by hand [17]. In contrast to visual query languages [2], Faceted Browsing does not return the constructed query explicitly. Instead it performs an incremental refinement of a set of results by selecting values of the data's facets that are turned into restrictions on the data set. The choices for creating new restrictions are adjusted, so that, by construction, only queries leading to a non-empty result set can be created. The paradigm is based on the librarian principle of Faceted Classification [4] which is difficult to implement in the physical world, but easy to do in a virtual world. Therefore it became popular the last years although often being implemented in a relaxed version (Sect. 2). In advantage to a fix predetermined taxonomic order [7] multiple navigations path exist and allow the user to browse task-oriented. Another point is that the user does not need to know what exactly she is looking for and recalling the exact terms for searching is not required. Thereby Faceted Classification is well suited for a process of exploration. Recently many Faceted Browsers have been implemented that are based on the idea of Faceted Classification, some of them following the early and famous Flamenco [12] browser. Additionally pragmatic facet-like systems have wide spread in the field of e-commerce. Now it is time to compare those approaches in order to find possibilities for further improvement of these browsers.

The rest of the paper is structured as follows: In Sect. 2 we will first describe commonalities of Faceted Browsers. In Sect. 3 we will then extract characteristic features of existing Faceted Browsers, divided into the fields of filtering and displaying results. In Sect. 4 we will propose a reflection of the facets structure to achieve better widgets.

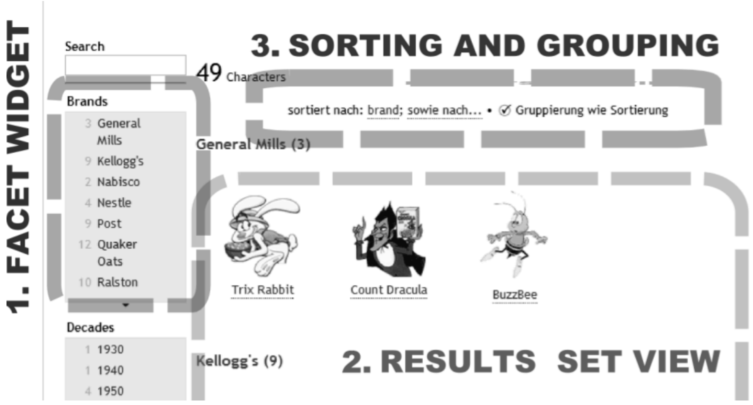


Fig. 1. GUI Elements of Faceted Browsers

2 Commonalities

In the context of Faceted Browsing we distinguish between *facets*, *facet values*, *restrictions* and *categories*. Every property of the artifacts in focus is seen as a facet in a Faceted Browser. For all items in the system this property may have different facet values. Priss mentions that according to Ranganathans original definition of Faceted Classification [4] an item can only have one value per facet. This is often not the case for many facet-like systems in computer science [5]. Additionally the properties are not even required to be fully orthogonal to be called a facet [16].

Putting a restriction to a facet *A* can be seen as a logical expression, reducing the set of items to those that have special facet values for facet *A*. Finally, the term *category* is used in the following sections for a special facet that is usually hierarchically structured by a *subClassOf* relationship and serves as a hierarchical backbone that already classifies a domain in one possible way.

GUI Elements Faceted Browsers offer both, a visualization of the selection mechanism and a means to visualize the data itself. Therefore the user interface of Faceted Browsers (Fig. 1) consists of: 1) widgets for each facet to select the facets values, 2) a view of the currently selected result set, 3) means to sort and possibly group the results. Additionally many browsers visualize the applied restrictions, i.e., the constructed query, as "breadcrumbs".

Instant Update of Facets and Results Common to the majority of all Faceted Browsers is the fact that the facet values the user has to choose from are instantly restricted to the possibilities which do not lead to an empty result set. This guarantees that a search always returns at least one instance. While most browsers completely remove facet values that lead to an empty result set, another approach is to only gray out those values. This allows the user to get an impression of items she might be able to indirectly navigate to [18].

3 Characteristics

We extracted a set of characteristics to compare Faceted Browsers. These can be divided into general characteristics referring to a browser as a whole (Sect. 3.1), characteristics regarding the filtering process with the facet widgets (Sect. 3.2) and the presentation of the results (Sect. 3.3).

3.1 General Characteristics

Domain Independence (without Configuration) Faceted Browsers can be divided into the group of domain-independent general approaches and domain-specific ones, which are usually existing applications that have been extended with Faceted Browsing capabilities.

Facet Set Context Depending Especially if the domain is very broad and many classes are included, it is beneficial to show only a subset of facets. Very specific ones, applying to only a few items, can be shown only after certain other conditions are met. This might be, for instance, when a certain class in a hierarchically structured class-facet is selected. To give a concrete example: Amazon [26] displays the facet *resolution* only if the class *camera* was chosen, not right from the beginning. Foundations for selecting a facet set automatically based on facet ranking and metrics are discussed by Oren et. al. [3]. Hearst describes the selection of important facets based on query logs and click logs [16]. Context-dependency cannot only be applied to facets but also to facet values (e.g., eBay [27] does this) and can not only regard the selection of a subset to be displayed, but also the order. When a subset of possibilities is presented it is a visual convention to insert a link labeled *more…* [15].

Arbitrary Restriction Order Restrictions can usually be added (zoom-in) and cleared again (zoom-out) in any order for all facets. This allows for exploring the data set on multiple paths, depending on the users intention. Other navigation modes such as Shift and Pivot are described in detail in [17].

Facet Set Configurable by the User This characteristic regards the flexibility of the user interface in terms of changing the set of facets and changing their order. Letting the user choose a set of appropriate facets and put those facets into a convenient order adds to the personalization of the navigation path.

3.2 Facet Widgets

Result Cardinalities as Feedback The vast majority of Faceted Browsers shows the number of items with a certain value for every value of each facet in the current result set. Consequently, this is also the number of items that will remain in the result set if this value is turned into a new restriction. This way, the cardinality can be seen as a preview and warning that anticipates the next potential restriction step. Stefaner and Müller [10] use color in addition to numbers to visualize the cardinality.

Value Range Selection Choosing a single value is often undesirable. For some facets, such as time or price, the user often wants to define a range of values between a (often vague) maximum and a minimum. Three different approaches can be found for the selection of ranges: The simplest one is the possibility to select a range by clicking on a start and an end value while pressing a key such as *Ctrl* or *Shift*, or by selecting multiple discrete values with the mouse. However, this approach is not suitable for continuous values. A second possibility is the predefinition of ranges, e.g., price from *10,00 - 20,00 EUR*; *21,00 - 30,00 EUR*. Finally, the most elegant way is the selection of special widgets using sliders, for instance.

Selection of Continuous Values The ability to define arbitrary values for continuous facets can be realized by text input fields, selection wheels or sliders. Since entering one continuous value alone will most likely not hit any item, the input of continuous values will usually be combined with the definition of ranges. While sliders and input fields in particular are a convenient way to edit restrictions, they make it harder to give feedback on remaining possible facet values.

Multiple Selection (MS) An alternative to the selection of ranges is the possibility of selecting multiple values of a facet. However, this has its limitations for facets with a large number of values and is only possible for discrete values. MS is possible, for example, in option lists, lists of checkboxes and could also be implemented for Tree Maps [20]. Sacco [7] stresses the importance of building disjunctions of multiple facet values to allow the user to form custom groups spontaneously without needing to select the whole super class[1]. MS is an important feature to allow the construction of complex queries with the help of Faceted Browsing. However, please note that by MS alone, the expressiveness of the constructed queries is still limited to conjunctions of disjunctions such like *(facetA=ValueX OR facetA=ValueY) AND (facetB=ValueZ)*[2]. Since Faceted Classification in computer science usually allows the assignment of multiple values per facet, it would be useful to allow building queries that can represent conjunctions of values within one facet, for instance: “All countries being part of Europe and Asia at the same time.” An intuitive graphical representation for this definition could be challenging however. Related is the problem of including objects with no value assigned for a given facet. EBay, has introduced an additional label *not specified* that can be additionally selected to give users the possibility of keeping those objects in the result set.

Free Input In contrast to the selection of predefined values, the free input of discrete, nominal values (by their string identifier) requires recall. Therefore free input is better suited to the selection of continuous (i.e., alphanumerical) values or

[1] Selecting a super class requires that a hierarchical widget is employed (Sect. 4).

[2] Parallax [18] (submitted) is a very recent approach that allows for construction of more complex queries, i.e., related sets of instances.

ranges. However, advanced auto-completion functions could be a considerable solution to tackle the problem of recall.

Integration of Free String Search Most browsers allow the combination of faceted search with a search based on string comparison. This can be realized in several flavors. First we can distinguish whether the items (the content) or the facet values (the metadata) are filtered[3]. Here, a second criterion is the scope of search. There are browsers offering only a single search field for all facets together and there are others using one search field for each facet. In the letter case only the facet values that are left for further restriction of this facet are compared to the string.

Integration of Standard Classification In contrast to an explicit query, all browsers require the use of some predefined static classification, because otherwise, starting with the top concept *Thing*, the set of facets becomes too large for the user interface to be efficiently used. However, multiple variants for this exist: i) The trivial case is when a single class is chosen by the author and all facets apply to this class (not considered in Table 1). This principle is used in domain specific browsers for media collections, for instance, where all objects are media-items. ii) A list of important (domain dependent) classes aka *Startingpoints*, *Primaries*[4] or *Focal Points* [6] is offered as a list or tree, this is an approach some generic browsers take. We can further distinguish when the user selects the class. Either j) she does this on one or more separate screens in the beginning of the exploration process (e.g., e-commerce solutions), or jj) the fixed classification is seen as one facet among others.

3.3 Presentation of the Results

Display Paradigm The way the result set is represented varies between the browsers and often multiple views are offered. The most common paradigm is a list representation. However, others can be found such as table, map, timeline, grid, thumbnail view.

Sorting of Results Configurable While nearly all browsers allow for sorting the result set by some facet, not all of them allow the user, or at least the admin, to define the facet for sorting. Another criterion is the possibility to sort by multiple facets with different priority. E.g., it can be filtered first by country then by company. (This is something that most sortable table implementations do not make explicit.)

Grouping of Results The (sorted) result set can often be additionally grouped into items with the same value for a given facet. When icons are grouped by multiple facets this results in a tree that can either be displayed explicitly or, for instance, by nested boxes. The facet for grouping can be chosen by the user in only a few browsers. Whenever sorting, grouping and a modification of the facet set were configurable this could be done separately for sorting, grouping and filtering.

Search for Similar Resources When a specific item is displayed in the result view, some browsers allow for searching for similar items directly. This can be done by linking other items with the same values for most of the facets (such as the product

[3] There are cases where we have no annotated main content, such as knowledge bases.

[4] http://www.w3.org/2005/04/fresnel-info/manual/#primary

recommendations in e-commerce applications). Similar manual navigation modes are referred to as (extensional and conceptual) shift navigation in [17].

4 Advanced Widgets for Faceted Browsing

For filtering, suitable widgets to choose the facet values have to be offered. We propose a selection of different widgets based on the characteristics of the facet to be filtered. In contrast, most of the current Faceted Browsers use the same widgets for all facet types and rarely has been paid attention to the fact that the facets have a structure besides alphabetical order, which can helpfully be reflected by the widget. The standard widget for selecting facet values is a list where the single values can be selected. However, for some data types and purposes other widgets are much more useful. E.g., when supporting the selection of value ranges some (but few) Faceted Browsers offer specific widgets such as sliders or two free input fields for a minimum and a maximum value. Specific widgets could either be chosen automatically, depending on the facets data type, or be defined manually for selected facets.

In the field of statistics, values are categorized into nominal, ordinal and quantitative data representing different scales of information on the relationship between the values. Depending on the category to which the facet values belong to, different widgets are appropriate to reflect this information.

Nominal Data If there is no relationship between the facet values, except for the fact that they are all different and can be distinguished by their name, the data is (only) nominal and theoretically could be represented as an unordered set of choices. A trivial sorting however can be performed using the alphabetical order of the labels.

Ordinal Data If there is an order relation between the values, the data can be referred to as ordinal. The topology build by the order relation might be a sequence, a tree or a DAG. As an example, for a facet that has a tree structure, take the facet *takesPlaceIn* defined as an RDF[5] property with a domain *Event*. It could have the class *GeographicalRegion* as a range. For the moment, assume that geographical regions are related to each other via an exclusive *partOf* relationship. Other *partOf* relationships can be found between body parts or composed components. The *subClassOf* relation which is typically used within taxonomies but also others such as *knows* can build a tree structure. The relationship between the facet values that is reflected by the widget does not necessarily have to be the facet relationship itself. A facet widget can be structured by any other relation that fulfils the condition of not leaving the facets range. Consequently, the range and the domain of a potential alternative structuring relation have to be the same and have to equal the facets range (Fig. 2b). The special case of a widget that is structured by the facet itself is shown in Fig. 2a. Sequences can be displayed by slider and option list widgets. Facets that have a hierarchical topology can be displayed by the classical directory tree as already seen in Fig. 2a but could also be displayed by a Tree Map widget as shown in Fig. 2d.

[5] The examples in this section use the Resource Description Framework (RDF, http://www.w3.org/RDF/) as a data representation format.

There are a few other existing widgets specialized to hierarchical facets: FacetZoom [21] is a novel zooming (facet) widget specialized to the visualization of tree structures and lends itself to be integrated with Faceted Browsers, too. There is a variety of hierarchical facets that could be efficiently displayed with the widget, including *partOf* relationships (e.g., for time spans, geographic regions or body parts). Also, in the code repository of Exhibit [11], some specific widgets such as hierarchical facet widgets (and also sliders for range selection) can be found which have, however, not been released so far. Another distinction in the context of tree navigation is whether the selection of a certain level can be performed with one click, because the whole hierarchy is explicitly shown in an expanded state, or if the selection is performed by an incremental refinement, level by level. With an explicit representation of the facets hierarchy the user can easily restrict a facet on several levels of concreteness. mSpace [20] uses the second approach when it displays two facet widgets, one for genre and another for subgenre. DAGs can be displayed, either as a graph, or by transformation to a tree by copying values.

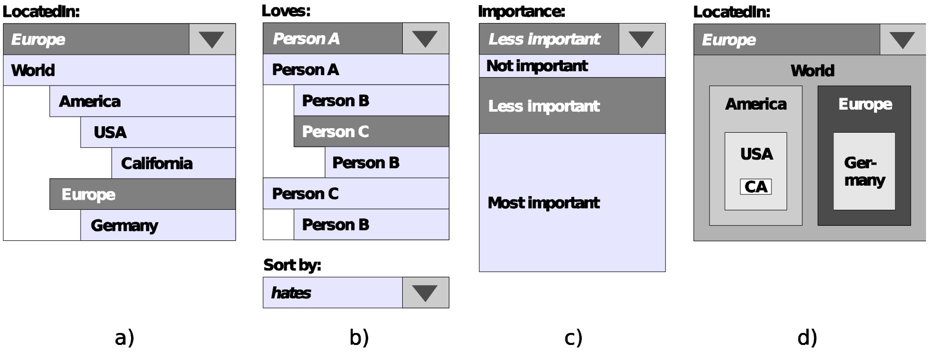
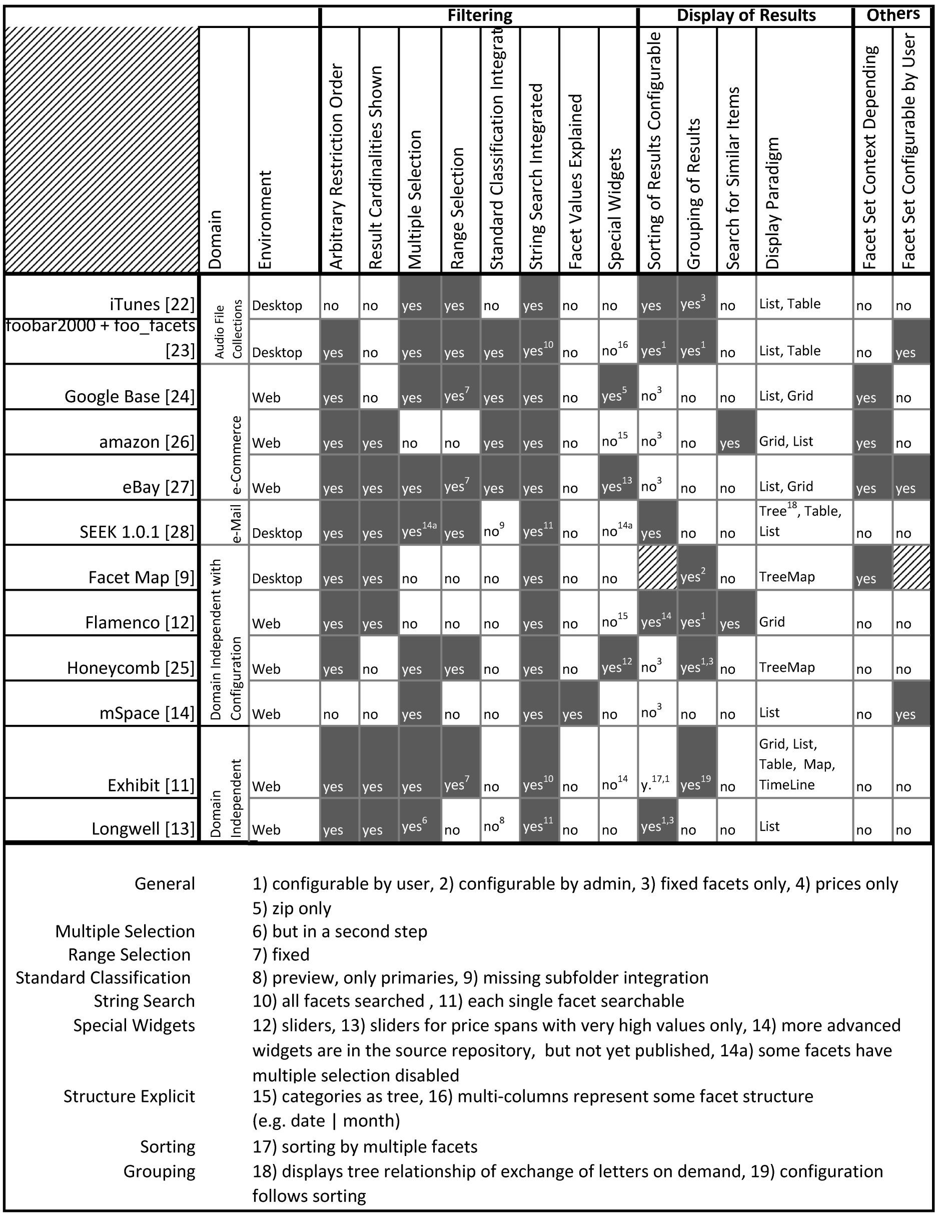


Fig. 2. Selection Widgets for Tree Facets (a), Tree Facets Structured by a Differing Relationship (b), Quantitative Data (c) and Tree Map Facet Widget (d)

Although hierarchical facet navigation already existed in Flamenco [15] and some widgets exist, we argue that widgets depending on the facets structure have not been consequently implemented in many later approaches. Often only typical taxonomical *subClassOf* relations are in the focus (for example in the context of dynamic taxonomies). Data from the semantic web offers explicit semantics and various relations. In [19] this knowledge is exploited for additional navigation modes. But it can also be used to construct advanced widgets representing this structure which might be flat (ordered or unordered) and also a hierarchy or a DAG.

Quantitative Data provides information on the ratio of values or at least on the ratio of differences between pairs of values (depending on the fact if *Null* was chosen in a meaningful way, or not). This quantitative information can be reflected by the area that is consumed by the entry for the value in the list, for example (Fig. 2c). A Tree Map is able to provide quantitative information by the proportions of areas and also a Pie Chart could be used as a widget. The Option List Widget can also be extended to represent quantitative information by using different row heights for each



			Filtering								Display of Results				Others	
▨	Domain	Environment	Arbitrary Restriction Order	Result Cardinalities Shown	Multiple Selection	Range Selection	Standard Classification Integrat	String Search Integrated	Facet Values Explained	Special Widgets	Sorting of Results Configurable	Grouping of Results	Search for Similar Items	Display Paradigm	Facet Set Context Depending	Facet Set Configurable by User
iTunes [22]	Audio File Collections	Desktop	no	no	yes	yes	no	yes	no	no	yes	yes[3]	no	List, Table	no	no
foobar2000 + foo_facets [23]		Desktop	yes	no	yes	yes	yes	yes[10]	no	no[16]	yes[1]	yes[1]	no	List, Table	no	yes
Google Base [24]	e-Commerce	Web	yes	no	yes	yes[7]	yes	yes	no	yes[5]	no[3]	no	no	List, Grid	yes	no
amazon [26]		Web	yes	yes	no	no	yes	yes	no	no[15]	no[3]	no	yes	Grid, List	yes	no
eBay [27]		Web	yes	yes	yes	yes[7]	yes	yes	no	yes[13]	no[3]	no	no	List, Grid	yes	yes
SEEK 1.0.1 [28]	e-Mail	Desktop	yes	yes	yes[14a]	yes	no[9]	yes[11]	no	no[14a]	yes	no	no	Tree[18], Table, List	no	no
Facet Map [9]	Domain Independent with Configuration	Desktop	yes	yes	no	no	no	yes	no	no	▨	yes[2]	no	TreeMap	yes	▨
Flamenco [12]		Web	yes	yes	no	no	no	yes	no	no[15]	yes[14]	yes[1]	yes	Grid	no	no
Honeycomb [25]		Web	yes	no	yes	yes	no	yes	no	yes[12]	no[3]	yes[1,3]	no	TreeMap	no	no
mSpace [14]		Web	no	no	yes	no	no	yes	yes	no	no[3]	no	no	List	no	yes
Exhibit [11]	Domain Independent	Web	yes	yes	yes	yes[7]	no	yes[10]	no	no[14]	y.[17,1]	yes[19]	no	Grid, List, Table, Map, TimeLine	no	no
Longwell [13]		Web	yes	yes	yes[6]	no	no[8]	yes[11]	no	no	yes[1,3]	no	no	List	no	no

General	1) configurable by user, 2) configurable by admin, 3) fixed facets only, 4) prices only 5) zip only
Multiple Selection	6) but in a second step
Range Selection	7) fixed
Standard Classification	8) preview, only primaries, 9) missing subfolder integration
String Search	10) all facets searched , 11) each single facet searchable
Special Widgets	12) sliders, 13) sliders for price spans with very high values only, 14) more advanced widgets are in the source repository, but not yet published, 14a) some facets have multiple selection disabled
Structure Explicit	15) categories as tree, 16) multi-columns represent some facet structure (e.g. date | month)
Sorting	17) sorting by multiple facets
Grouping	18) displays tree relationship of exchange of letters on demand, 19) configuration follows sorting

Table 1. Comparison of some Existing Faceted Browsing Applications

value to reflect the ratio to other values (Fig. 2d). Stefaner and Müller [10] have developed a widget that they call Elastic Lists. Here the widget already reflects the proportional value of one facet value in respect to all other values in the same facet. E.g., it does not simply draw equal rows for each facet value, but maps cardinality to the rows height. Likewise brightness stands for characteristicness and unusualness of one facet value among all others. Quantitative data always provides sequence, so it is always possible to handle only the ordinal data aspect. All numeral simple data types such as the XML-Schema data types *xsd:float* and *xsd:integer* offer quantitative data.

5 Conclusion

In this paper we identified characteristics of existing Faceted Browsing applications. We conclude that, although Faceted Browsing becomes more and more frequently integrated into user interfaces for several domains and although some domain independent solutions exist, the browsers support different useful features and could benefit from adopting each other’s principles. While there is an increasing number of different views for result presentation such as maps, timelines and tables, this does not apply equally to facet widgets. Especially the generic approaches are often limited to simple unified facet widgets that could better be replaced by facet-depending widgets. A basis for creating such specific widgets for the case of data with explicit semantics has been introduced. Based on the extracted characteristics, a detailed tabular comparison of existing Faceted Browsers, including both generic approaches and Faceted Browsing extensions of existing domain specific applications, has been done (Table 1). However, the details cannot be published here due to space limitations and will be available at http://www.polowinski.de/work/facetedbrowsing.

6 References

[1]Berners-Lee, T., Hendler, J., Lassila, O.: The Semantic Web. Scientific American, 284(5), pp. 28–37, 2001.

[2]Fadhil, A., Haarslev, V.: GLOO: A Graphical Query Language for OWL Ontologies. OWL: Experience, Directions, pp. 215–260, 2006.

[3]Oren, E., Delbru, R., Decker, S.: Extending faceted navigation for RDF data. ISWC 2006, 5th International Semantic Web Conference, Athens, Georgia, USA, November 5-9, 2006, Proceedings, 2006.

[4]Priss, U.: Faceted Information Representation. Working with Conceptual Structures, Contributions to Proceedings of the 8th International Conference on Conceptual Structures (ICCS 2000). Shaker Verlag, Achene, pp. 84–94, 2000.

[5]Priss, U.: Facet-like Structures in Computer Science. Axiomathes, 18, pp. 243–255, 2008.

[6]Rutledge, L., Hardman, L.: Making RDF presentable: integrated global and local semantic Web browsing. Proceedings of the 14th international conference on World Wide Web, pp. 199–206, 2005.

[7]Sacco, G.: Some Research Results in Dynamic Taxonomy and Faceted Search Systems. SIGIR 2006 Workshop on Faceted Search, 2006.

[8]Sauermann, L., Bernardi, A., Dengel, A.: Overview and outlook on the semantic desktop. Proceedings of the 1st Workshop on the Semantic Desktop at the ISWC 2005 Conference, 2005.

[9]Smith, G., Czerwinski, M., Meyers, B., Robbins, D., Robertson, G., Tan, D.S.: FacetMap: A Scalable Search and Browse Visualization. IEEE Transactions on Visualization and Computer Graphics, 12(5), pp. 797–804, 2006.

[10]Stefaner, M., Müller, B.: Elastic lists for facet browsers. Database and Expert Systems Applications, 2007. DEXA '07. 18th International Conference on, pp. 217–221, 2007.

[11]Exhibit, http://simile.mit.edu/wiki/Exhibit

[12]Flamenco, http://flamenco.berkeley.edu/

[13]Longwell, http://simile.mit.edu/wiki/Longwell

[14]mSpace, http://demo.mspace.fm

[15]Hearst, M.: Design Recommendations for Hierarchical Faceted Search Interfaces, ACM SIGIR Workshop on Faceted Search, 2006.

[16]Hearst, M.: UIs for Faceted Navigation: Recent Advances and Remaining Open Problems, HCIR08 Second Workshop on Human-Computer Interaction and Information Retrieval, 2008.

[17]Ferré, S.: Agile Browsing of a Document Collection with Dynamic Taxonomies, Database and Expert Systems Application, 2008. DEXA'08., 377—381, 2008.

[18]Huynh, D. F.: Parallax and Companion: Set-based Browsing for the Data Web, WWW 2009, 2009. (submitted, to be published)

[19]Allard P., Ferré S.: Dynamic Taxonomies for the Semantic Web, Database and Expert Systems Application, 2008. DEXA'08, 2008.

[20]Johnson, B., Shneiderman, B.: TreeMaps: A space-filling approach to the visualization of hierarchical information structures. Proc. Visualization ’91, 284—291, 1991.

[21]Dachselt, R., Frisch, M., Weiland, M.: FacetZoom: A Continuous Multi-Scale Widget for Navigating Hierarchical Metadata, ACM Press, Florence, Italy, 2008.

[22]iTunes, http://www.apple.com/de/itunes/

[23]foobar2000 + foo facets, http://foobar2000.audiohq.de/foo_facets/

[24]Google Base, http://base.google.com

[25]Honeycomb, http://www.hivegroup.com/

[26]Amazon, http://www.amazon.com

[27]eBay, http://www.ebay.com

[28]Seek, http://simile.mit.edu/wiki/Seek