gFacet: A Browser for the Web of Data

Philipp Heim, Jürgen Ziegler and Steffen Lohmann

University of Duisburg-Essen
Lotharstr. 65, 47057 Duisburg, Germany
{Philipp.Heim, Juergen.Ziegler, Steffen.Lohmann}@uni-due.de

Abstract. This paper introduces a new approach to browsing the Web of data by combining graph-based visualization with faceted filtering techniques. The graph-based visualization of facets supports the integration of different domains and an efficient exploration of highly structured and interrelated datasets. It allows to access information from distant user-defined perspectives and thereby enables the exploration beyond the borders of Web pages.

Keywords: Web of Data, Graph-based Visualization, RDF Data, Semantic Web, Hierarchical Facets, Faceted Browsing.

1 Introduction

The Web is commonly thought of as a Web of documents. With the steady growth of the Semantic Web, this Web of documents is more and more transformed into a Web of data. Information is no longer stored as monolithic blocks in Web pages, but is fragmented into many pieces that can be assigned to ontological concepts. This goes along with a fragmentation of the linking structure, transforming the untyped links between Web pages into typed relationships between information pieces.

The information pieces along with the concepts linked to them and the typed relationships are stored in RDF [1], leading to large and highly interrelated RDF datasets. Several such datasets for different domains are already available on the Web; examples include DBpedia[1], FOAF profiles[2] or MusicBrainz[3]. The semantic annotation stored in RDF datasets allows Web content to be found, shared and combined more efficiently and hence fosters the integration of data from distributed sources. These developments require a new generation of applications that provide a single access point to information on the Web in order to facilitate the exploration beyond the borders of Web pages.

In this paper we introduce gFacet, a browsing approach that supports the exploration of the Web of data by combining graph-based visualization with faceted filtering functionalities. The graph-based visualization facilitates a comprehensible integration of different domains; the use of facets supports a controlled filtering of information.

[1] http://dbpedia.org
[2] http://foaf-project.org/
[3] http://musicbrainz.org/

With gFacet, users are enabled to browse the Web of data efficiently and to retrieve information from different user-defined perspectives.

The remainder of this paper is structured as follows. In section 2, we give an overview of existing approaches to visualize and browse RDF data. In section 3, we introduce our approach and describe a prototypical implementation. We proceed with an evaluation of the prototype in section 4 and close the paper with a conclusion and outlook on future work in section 5.

2 Overview of Existing Approaches

Tools to visualize and browse RDF data have been developed within several projects. We can classify these tools into two main groups, each using a different approach. The first group uses a graph-based approach that explicitly visualizes the structure of the RDF datasets by nodes and edges. The second group divides the screen into several areas, each containing a subset of the data, allowing for faceted browsing.

2.1 Graph-based Approach

The RDF syntax is based on triples in the form of subject-predicate-object expressions. An RDF graph is a set of such triples with nodes being subjects or objects and labeled edges being predicates [1]. Thus, using a visualization that directly presents this RDF graph structure to the users seems obvious and is suitable to show the structural complexity of the relations within the data. For instance, RDF Gravity [2], Welkin [3], IsaViz [4] and Paged Graph Visualization [5] are such tools that directly visualize the RDF graph.

Since RDF datasets are often large and highly interconnected [6], visualizing all the relations that exist within the graph structure can quickly cause a high number of edge crossings and hence hinder an understandable visualization of the data. In order to prevent the graph visualization from getting over-cluttered, different mechanisms to reduce the displayed information are provided. RDF Gravity, for example, applies local and global filter techniques to hide certain nodes and edges. The Paged Graph Visualization by contrast starts with only a small set of information and let the user gradually expand this set to explore specific parts and fragments of the graph.

Even though these filter mechanisms can help to reduce the displayed information, they are not appropriate to visualize complex interrelations within large datasets. The direct presentation of all the existing relations between entities only scales well to small sets of data with a limited number of interrelations [7]. The visualization of a large dataset with many interrelations on one screen will still tend to result in a complex graph, which is hardly manageable or understandable by the user [8].

2.2 Facet-based Approach

Another way to visualize large and complex RDF datasets is based on the concept of faceted exploration [9]. In faceted exploration, the data gets partitioned using orthogonal conceptual dimensions. One of the dimensions serves as the result set and the others are used as facets to filter the result set by different attributes that can be selected independently from each other.

Visual models of faceted exploration mostly apply an approach where the different facets and the result set filtered by the current facet selections are shown as lists in different screen areas. A popular example is Apple's music and media player iTunes [10] that uses the faceted filtering approach to let the user access music according to different attributes, such as the artist, album, or genre. The relation between the attribute values and the target objects is not directly visible but can only be recognized indirectly based on the filtering behavior exhibited by the system.

The faceted-browsing paradigm is especially applicable to large datasets with many interrelations. Several tools, such as mSpace [11], Longwell [12], /facet [13] or Haystack [14], use this way of browsing for the visualization and exploration of RDF data. The facets are automatically constructed and can be added and removed to the screen allowing the user to select facets that are of specific interest. However, Haystack, mSpace and Longwell are limited only to facets that are directly related to the result set. With the use of hierarchical facets users become empowered to access information from distant perspectives also.

Hierarchical Facets. Hierarchical facets are facets that are indirectly related to the result set. The simplest case of a hierarchical facet is a facet that has a direct relation to a facet that is in turn directly related to the result set. However, hierarchical facets can be indirectly related to the result set by more than one facet and hence allow to access the result set also from distant perspectives. A common way to visualize them is a tree that can be gradually expanded by the user. Both, Tabulator [15] and the Nested Faceted Browser [16] use trees to present hierarchical facets to the users.

If we extend the iTunes example by hierarchical facets, the filtering is no longer restricted to related facets such as artists, albums or genres, but can also include facets of facets such as the nationality of the artists. By providing hierarchical facets, users become enabled to access information from a broad range of different, user-defined perspectives.

3 Our Approach

The basic idea of our approach for the visualization and exploration of the Web of data is to combine the graph-based and the facet-based approach by arranging facets as nodes in a graph structure. With this specific combination of both visualization paradigms, we are able to counterbalance their respective disadvantages and facilitate the exploration of large and highly interrelated RDF datasets by the user. The major aims of our approach are:

1. **Prevention of an over-cluttered graph:** Using a facet-based visualization groups the instances contained in the Web of data into facets according to their conceptual structure. This prevents relations between single instances to be directly visualized by edges in the graph. Relations between these information pieces become only

indirectly visible when certain instances in a facet get selected by the user in order to filter the result set.

2. **Representation of relations between facets:** Using a graph-based visualization for the presentation of facets allows the relations between facets to be visualized as labeled edges. In doing so, the relations become explicit and dependencies as well as the filtering of the result set are more easily recognizable and traceable. This is especially useful for the visualization and understanding of hierarchical facets: Visualizing the dependencies and relations by labeled edges between facets can facilitate the understanding of how hierarchical facets can be used to find certain information.
3. **Single coherent visualization:** Using a graph-based visualization allows the information to be displayed in a single visualization instead of being spread over several screens or windows. Thus, information from distributed sources can be presented at once and hence reduce the cognitive load of the users and prevents them from getting "lost in hyperspace".

3.1 Prototype

We developed a first prototype to evaluate our proposed approach for graph-based faceted exploration of RDF data. The prototype is implemented in Flash and can be accessed via the Internet[4]. The RDF data is requested by the use of SPARQL[5], a protocol and RDF query language. In the following, the design of the visualization is demonstrated by an example from the field of music, where song information is represented as RDF statements.

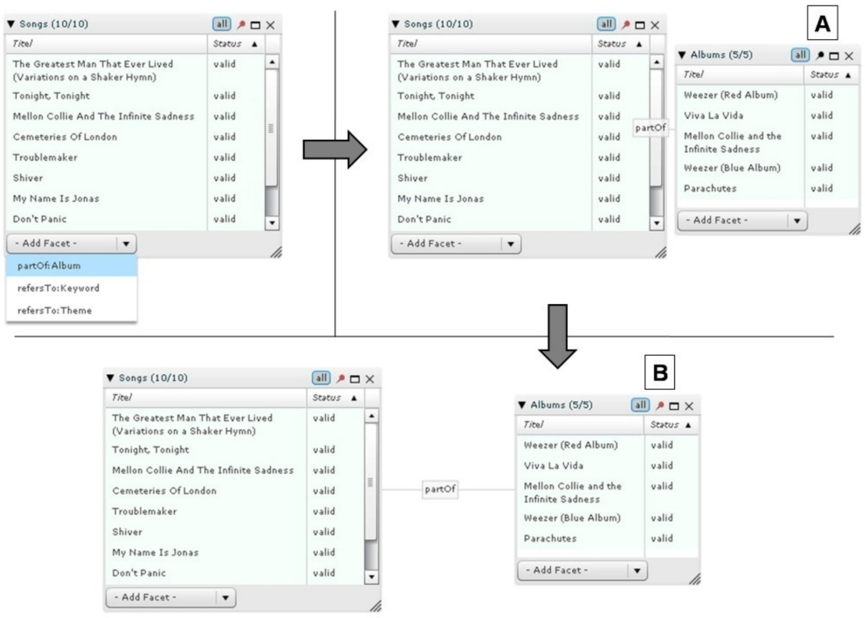
Initially, the graph consists of a single list-valued node that can gradually be expanded by other nodes that serve as facets for filtering the list in the original node. In the upper left corner of Figure 1, the initial node contains a list of songs. All directly related facets are shown in a pull down menu below the list. If the user selects one of these relations, a new node is added to the graph and gets connected to the original node by a labeled edge. The label represents the predicate that connects the different types of information in the RDF data, in this case "partOf".

A force-directed layout algorithm [17] is performed that assigns spring-like forces to position the nodes of the graph so that all the edges are of more or less equal length and there are as few crossing edges as possible. The forces are iteratively applied to the nodes, pulling and pushing them until a stable state is reached. This allows the user to follow how the graph evolves and to understand how a new node or relation affects the graph layout.

The permanent movement of all nodes, however, is also a disadvantage because already found information can move to other locations and hence must be retrieved again by the user. Therefore, we introduce a pinning mechanism that forces already shown information to hold its position. When a new node is added to the graph, the pinning of this node is executed after a short period of time. This delay allows the force-directed algorithm to position new nodes in an appropriate way and at the same time prevents already existing information to change its location.

[4] http://www.gFacet.org

[5] http://www.w3.org/TR/rdf-sparql-query/



Titel	Status ▲
The Greatest Man That Ever Lived (Variations on a Shaker Hymn)	valid
Tonight, Tonight	valid
Mellon Collie And The Infinite Sadness	valid
Cemeteries Of London	valid
Troublemaker	valid
Shiver	valid
My Name Is Jonas	valid
Don't Panic	valid

Titel	Status ▲
Weezer (Red Album)	valid
Viva La Vida	valid
Mellon Collie and the Infinite Sadness	valid
Weezer (Blue Album)	valid
Parachutes	valid

Fig. 1. The user adds new facets to the graph by selecting them from a menu below the list. Facets are automatically arranged by a force-directed layout.

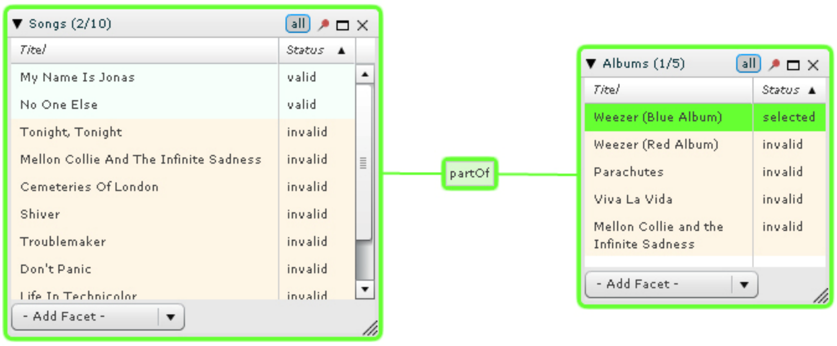
In Figure 1, the new node can move away from the existing node until both do not overlap anymore, because of the delay in pinning its position. After the delay, the pinning is automatically executed and the node is fixed. The color of the needle symbol at the upper right corner of each node indicates whether a node is pinned or not. If the symbol is grey (Figure 1, A), the node is movable, if it is colored (Figure 1, B), the node is pinned. This pinning can also be controlled by the user by clicking at the needle symbol. That way, the user can decide whether a node should stay on a fixed position or should be rearranged by the force-directed algorithm in order to improve the overall appearance of the graph.

Graph-based Facets

The new node can be used as a facet to filter the connected nodes. By selecting a certain row in the list of albums, the list of songs gets filtered to only songs that are part of this album (see Figure 2). In order to prevent hiding possibly valuable information, all other songs as well as all other albums are still visible but marked invalid. The information about the validity of each row is shown in the column “status” and is additionally represented in the background color (see Figure 2). Initially, the rows are arranged according to their status in the following order:

1. **Selected:** Rows that are selected by the user.
2. **Valid:** Rows that are exclusively related to valid or selected rows and are not selected itself.
3. **Invalid:** Rows that are related to at least one invalid row and are not selected itself.

The number of valid rows together with the number of all rows is shown within brackets behind the type of information that is contained in a node, for example “Songs (2/10)” in Figure 2. Deselecting the by default selected button “all” at the top of the node reduces the visible rows in the list to only valid ones.


Songs (2/10)

Titel	Status
My Name Is Jonas	valid
No One Else	valid
Tonight, Tonight	invalid
Mellon Collie And The Infinite Sadness	invalid
Cemeteries Of London	invalid
Shiver	invalid
Troublemaker	invalid
Don't Panic	invalid
Life In Technicolor	invalid

- Add Facet -

partOf

Albums (1/5)

Titel	Status
Weezer (Blue Album)	selected
Weezer (Red Album)	invalid
Parachutes	invalid
Viva La Vida	invalid
Mellon Collie and the Infinite Sadness	invalid

- Add Facet -

Fig. 2. Information can be filtered by selecting elements in a node.

In order to facilitate users’ understanding of how the elements in the nodes are filtered by certain selections, changes are highlighted by different colors (see Figure 2). Whenever a certain selection causes a decrease of valid rows in at least one node, this is reported by a specific color. The color depends on the node where the selection takes place and provides three different kinds of information:

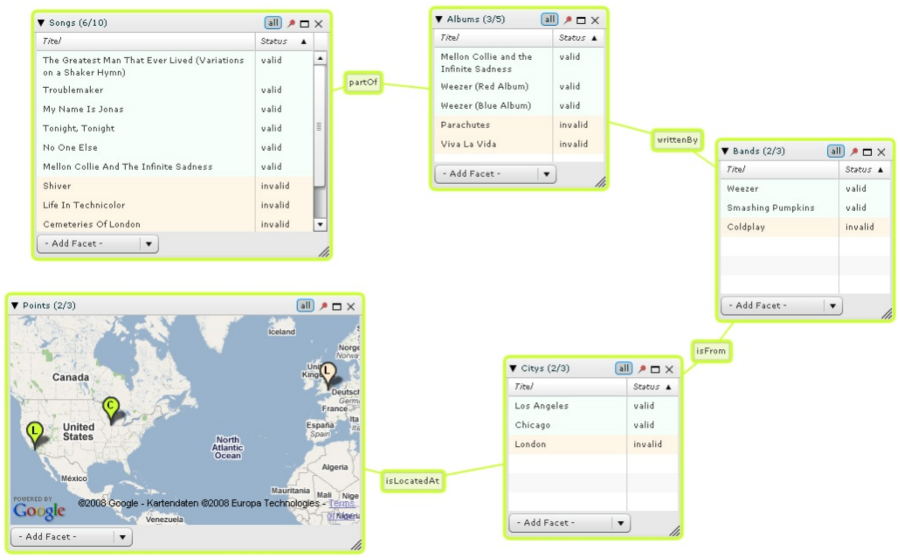
1. **Selections:** Selected rows are highlighted in the distinct color of their nodes (see “Weezer (Blue Album)” in Figure 2).
2. **Restrictions:** Nodes that are restricted by a selection get surrounded by a ring in the corresponding color. If they are restricted by several selections from different nodes they get surrounded by several rings in different colors.
3. **Distributions:** Labeled edges that connect restricted nodes with the source of their restriction are colored correspondingly (see “partOf” in Figure 2).

The user can click on an already selected row to deselect it and thereby remove the restrictions and the corresponding coloring that were caused by this selection.

Hierarchical Facets

Our graph-based approach is particularly suitable for the presentation of relationships. Even nested relationships can be easily represented by nodes and labeled edges. In order to access information from distant perspectives, the graph can be gradually expanded by adding new nodes along the available relations.

Such indirectly related nodes can be used for filtering in the same way as directly related nodes. If the user selects certain rows to filter directly connected nodes as described above, the filtering of these nodes automatically leads to an update of again all directly related nodes and so on. The changes are iteratively propagated through the graph structure until no further reduction is caused and the elements in the nodes reach a stable status. A detailed description of this propagation algorithm is beyond the scope of this paper.



Titel	Status ▲
The Greatest Man That Ever Lived (Variations on a Shaker Hymn)	valid
Troublemaker	valid
My Name Is Jonas	valid
Tonight, Tonight	valid
No One Else	valid
Mellon Collie And The Infinite Sadness	valid
Shiver	invalid
Life In Technicolor	invalid
Cemeteries Of London	invalid

Titel	Status ▲
Mellon Collie and the Infinite Sadness	valid
Weezer (Red Album)	valid
Weezer (Blue Album)	valid
Parachutes	invalid
Viva La Vida	invalid

Titel	Status ▲
Weezer	valid
Smashing Pumpkins	valid
Coldplay	invalid

Titel	Status ▲
Los Angeles	valid
Chicago	valid
London	invalid

Fig. 3. Hierarchical facets can be used to filter information from distant perspectives.

Figure 3 illustrates how indirect relationships are visualized in our graph-based interface and how they can be used for filtering. The selection of Los Angeles and Chicago in the map restricts the valid music bands to only those that are from these cities. This leads to a restriction of the albums and hence to a restriction of the songs to only those that are part of these albums. That way, geographical locations of cities can be used to filter music songs. Similarly, other domains can be included in the graph to access and filter information from distant perspectives.

Since geographic coordinates expressed as latitude-longitude are not suitable for human processing when presented in a list, we used Google Map's geocoding service[6] to provide a map visualization. On a map, the distribution or distance of cities can be immediately understood and the corresponding areas and countries they are located in can be easily recognized. Our general aim is to provide a data- and task-related visualization where the system initially decides on the presentation form and the user then refines or corrects the system choice if it does not meet his needs. However, the currently implemented forms of presentation are limited to only lists and maps and are automatically selected. We leave the development of a semi-automatic approach with a broader range of presentation forms to future work.

4 User Study

We conducted a small user study to get some first insights on how well users can find information contained in RDF data using gFacet. Our primary interest was in how well users could apply graph-based facets to filter information. To perform this evaluation,

[6] http://code.google.com/apis/maps/

we asked users to find answers to three different questions. To assess users' understanding of the visual interface, we used an eye tracking system[7] in combination with a questionnaire. The results from the eye tracking helped us to better understand the actions performed and the answers given by the users. Note that the aim of this study was to examine whether our approach of graph-based facets is understood and purposefully applied by the user to find information contained in RDF data. We leave a comparative study as future work. 10 participants with an average age of 31, ranging from 21 to 54 years, took part in the study. Three of the participants were female and seven were male. Four participants had already heard about faceted browsing, six had not.

Search Tasks. To determine how well the participants can find certain information, we defined three tasks. With every task, they had to understand an additional functionality of our tool in order to find the information and answer the question. Since prior knowledge about faceted browsing can be supposed to have an impact on the results, we divided the participants into two groups according to their previous knowledge about faceted browsing. In the following we call the group without previous knowledge "Group noPK" and the group with previous knowledge "Group PK".

Task 1. For the first task, the users had to understand how new facets are added to the initially visible list of songs and how these facets can be used to filter the songs to only those that refer to the theme "rock". Three of the four participants of Group PK were able to find these songs. By contrast, only one of the six participants of Group noPK found them. The rates of success for each of the three tasks can be seen in Figure 6.

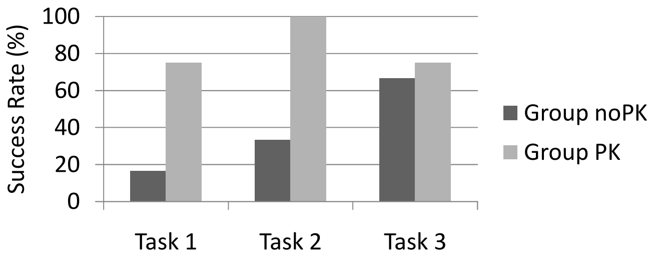


Fig. 6. Success rates by tasks.

Task 2. In order to solve the second task, the users had to understand the principle of hierarchical facets and how it can be used in our tool. To confirm their understanding, the participants were asked to find all songs that are part of albums written by the band "Smashing Pumpkins". All the participants of Group PK were able to use the hierarchical facet to find the right answer. The eye tracking results in Figure 7 suggest that the participants understood the way the connected nodes were filtered based on the selection they made in the hierarchical facet. The eye movement starts at the selected row and follows the highlighted restrictions through the graph structure. It seems as if the edges between the nodes together with the highlighting have guided the eyes to

[7] Tobii T60 Eye Tracker

look at the right spots. This can be also seen in the heatmap, where red areas indicate the highest number of eye fixations and green areas the fewest. However, only two participants of Group noPK found the right answer within the given time period.

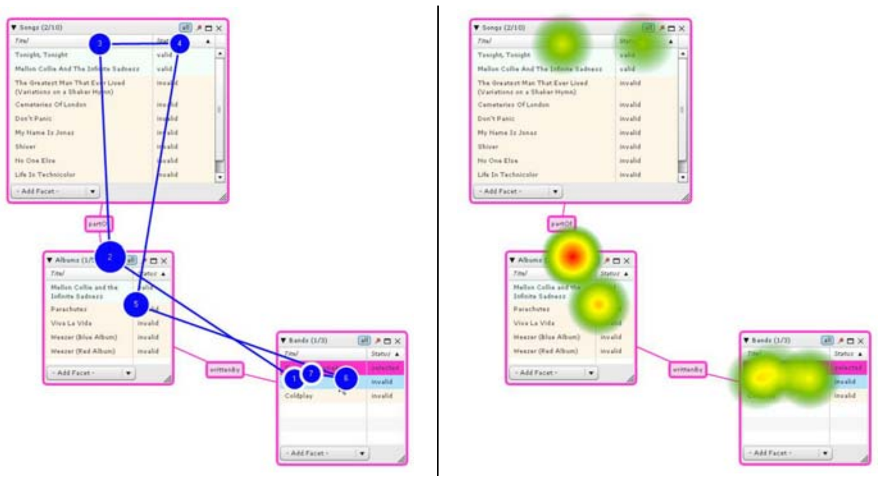


Fig. 7. Typical eye movements and areas of interest.

Task 3. In the third task we wanted to evaluate if the participants were able to use the songs to filter the list of albums. They had to select a certain song in order to get the album it belongs to. All in all, only three participants did not succeed this task, the other seven did.

5 Conclusion and Future Work

In this paper, we presented a new approach for browsing the Web of data that combines a graph-based visualization with faceted filtering techniques. The approach supports the dynamic and user-defined integration of distant domains in order to allow the user to access information from different perspectives. This is facilitated by the separation of content and presentation as well as the fragmentation of the information and its relations that comes along with the transformation of the Web of documents to a Web of data.

We use a graph-based visualization to facilitate an understandable integration of different domains and a facet-based filter mechanism to support an efficient exploration of highly structured and interrelated datasets. The combination of a graph and facets is of particular benefit when dealing with complex dependencies as for example within hierarchical facets.

As a general conclusion of our user study, it can be stated that the direct representation of the relationships between facets by labeled edges in a graph facilitates users' understanding of their dependencies. However, the study also showed that participants with previous knowledge about faceted browsing performed better than participants without.

In future work we will extend the visual appearance of the facets and include further presentations forms next to lists and maps. Furthermore, we plan to conduct a comparative study of our tool in order to draw more general conclusions.

References

1. Resource Description Framework (RDF). http://www.w3.org/RDF/.
2. Goyal, S., Wetenthaler, R.: RDF Gravity. Salzburg Research (2004). http://semweb.salzburgresearch.at/apps/rdf-gravity/.
3. SIMILE: Welkin (2004-2005). http://simile.mit.edu/welkin/.
4. IsaViz: A Visual Authoring Tool for RDF (2001-2006). http://www.w3.org/2001/11/IsaViz/.
5. Deligiannidis, L., Kochut, K., Sheth, A.: RDF data exploration and visualization. In: Proc. of the first workshop on CyberInfrastructure 2007, ACM Press, pp. 39-46. (2007)
6. Angles, R., Gutierrez, C.: Querying RDF data from a graph database perspective. In: Proc. ESWC 2005, pp. 346-360. (2005)
7. Frasincar, F., Telea, A., Houben, G.-J.: Adapting graph visualization techniques for the visualization of RDF data. In: Visualizing the Semantic Web 2006, pp. 154-171. (2006)
8. Schraefel, m. c., Karger, D.: The Pathetic Fallacy of RDF. In: International Workshop on the Semantic Web and User Interaction (SWUI) 2006. (2006)
9. Yee, K.-P., Swearingen, K., Li, K., Hearst, M.: Faceted metadata for image search and browsing. In: CHI 03, ACM Press, pp. 401-408. (2003)
10. A. Inc. itunes. http://www.apple.com/itunes/.
11. Schraefel, m.c., Smith, D., Owens, A., Russell, A., Harris, C., Wilson, M.: The evolving mSpace platform: leveraging the semantic web on the trail of the memex. In: Proc. of Hypertext 2005, ACM Press, pp. 174-183. (2005)
12. SIMILE: Longwell RDF Browser (2003-2005). http://simile.mit.edu/longwell/.
13. Hildebrand, M., Ossenbruggen, J., Hardman, L.: /facet: A browser for heterogeneous semantic web repositories. In: Proc. ISWC 2006. (2006)
14. Quan, D., Huynh, D., Karger, D.: Haystack: A Platform for Authoring End User Semantic Web Applications. In: Proc. of ISWC 2003, pp. 738-753. (2003)
15. Berners-Lee, T. Chen, Y., Chilton, L., Connolly, D., Dhanaraj, R., Hollenbach, J., Lerer, A., Sheets, D.: Tabulator: Exploring and Analyzing linked data on the Semantic Web. In: Proc. of the 3rd International Semantic Web User Interaction Workshop 2006. (2006)
16. Huynh D.: Nested faceted browsing. http://people.csail.mit.edu/dfhuynh/projects/nfb/.
17. Fruchterman, T., Reingold, E.: Graph drawing by force-directed placement. In: Softw. Pract. Exper. 1991, John Wiley & Sons, pp. 1129-1164. (1991)