Navigation Based on Adaptive Views

Karol Rástočný *
Institute of Informatics and Software Engineering
Faculty of Informatics and Information Technologies
Slovak University of Technology in Bratislava
Ilkovičova 3, 842 16 Bratislava, Slovakia
xrastocny@stuba.sk

Abstract

We propose novel approach for navigation in the Semantic Web, which visualizes its repositories in two-dimensional graph of resources and their properties. Our view-based navigation helps users more easily understand and learn browsed information space. This decreases users' effort to find relevant results in similar and related resources in fast-growing web repositories. Although graphs naturally visualize relations between resources, their size can make them unreadable and abstruse for conventional users. We avoid these problems by combination of layouting algorithm, supporting tools, next actions recommendation and hierarchy of adaptive views on results, where recommendation of next actions and adaptability to the user's interests are based on analysis of users' actions, their time and order.

Categories and Subject Descriptors

H.3 [**Information Storage and Retrieval**]: Information Search and Retrieval; H.5 [**Information interfaces and presentation**]: User Interfaces

Keywords

Semantic Web, adaptive views, hierarchical results clustering, graph visualization, navigation

1. Introduction

Increasing amount of information in web repositories affects the number of identified results for users' queries, which can decrease the relevance of these results. Users can increase the success rate of their query for conventional search engines (like Google or Bing), when they can exactly describe required result with keywords. This problem is quite successfully solved with a keyword-based query expansion [13] and with approaches based on exploratory search [5]. But it will often happen that a user has found required result already and he wants to find similar and/or related results, now. One type of this service is provided by Google (Google Similar), but results offered by this service are only displayed in list view without any information, why they are evaluated as similar (e.g. why the second result for a similar image to cotton bolls photo is a photo with a dog). In our approach we address this problem via view-based search within the Semantic Web using navigation in a two dimensional graph of objects from ontological repository and their properties.

Our graph navigation approach is based on bachelor work of Adrian Rakovský [7], where the concept of web browsing based on graph visualization was proposed. The main benefit of this approach is that users can see dependencies between the resource (original result) and new results visualized in graph. On the other hand, we identified quite large deficit in our work: Visualized graph can quickly grow to enormous size, so it becomes unclear and unusable for common users. To avoid this we extend this approach via result clustering, facet marking, adaptability to user's interests, next action recommendation and abstract zoom. The first three extensions aim to reduce the number of nodes displayed in the graph and the last two help users to orientate in the graph and understand the graph.

Next problem is that user search activities – lookup, learn and investigate [5], are not performed all at the same time, but users often perform them in succession during search sessions. Obviously, when a user begins a search session, he does not know information space, where he is searching very well. So he should be able to redefine his query, while he learns about identified results. When user has already found wanted result and learned basic information about it, he obviously wants to know more about related resources, so he need support for navigation in them. We address this problem via navigation based on a hierarchy of adaptive views integrated in a faceted browser that helps users search, navigate and explore related results. These views display information adaptively with respect to user interests and the type of provided content.

---

*Master degree study programme in field Software Engineering. Supervisor: Doctor Michal Tvarožek, Institute of Informatics and Software Engineering, Faculty of Informatics and Information Technologies, STU in Bratislava. Work described in this paper was presented at the 7th Student Research Conference in Informatics and Information Technologies IIT.SRC 2011.

Rástočný, K. Navigation Based on Adaptive Views. Information Sciences and Technologies Bulletin of the ACM Slovakia, Special Section on Student Research in Informatics and Information Technologies, Vol. 3, No. 2 (2011) 104-108

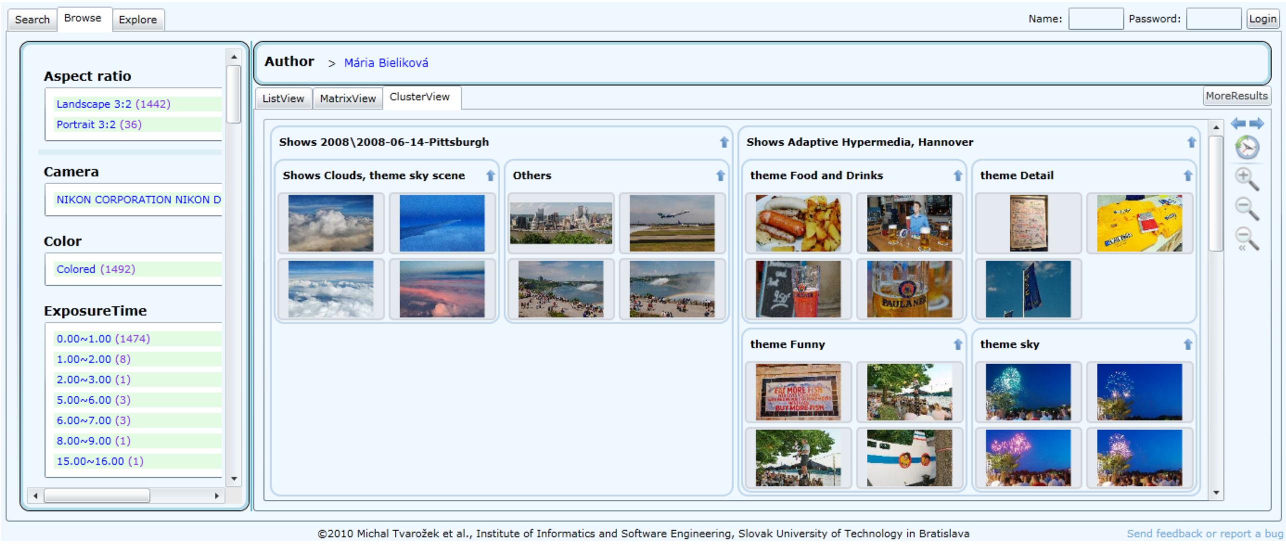

Figure 1: The hierarchical cluster view integrated in Factic. First-level clusters contain four most prominent second-level clusters, which display four representatives with the highest membership degree. Buttons on the right provide functionality of navigation in users actions history and zoom.

2. Related Work

The most of current tools, that enhance user navigation and browsing in the legacy Web by application of its metadata (where can be useful the Semantic Web), use for user navigation facets, table or graph view. Each of these concepts has specific advantages and disadvantages that can be utilized or suppressed in varying degrees by concrete implementation.

mSpace is classical representative of faceted browsers, which obtains facets for navigation from pre-calculated multidimensional model of multimedia ontology [8]. This style of browsing in information space is easily understandable and usable by users. Disadvantage is that users are confronted with all facets and identified results without any adaptation to users' interests and behaviors. This problem is being solved by Factic, which is concept of the Semantic Web faceted browser with adaptively generated user interface [11]. It also provides visualization of selected resource (result) in a graph, where users can see semantic connection to other elements from database (ontological repository). The graph visualizes RDF triples by three nodes (subject, predicate, object) connected with oriented edges [10], so users can see related resources and type of relation between them. This visualization can help users to navigate among similar and related resources, but this graph will become unclear at a short time. Authors of this concept chose uniform graph layout based on springs physics algorithm. This algorithm visualize changes in a graph and gives symmetric graph layout, which is very important graph property for users [12].

Visualization with a graph is used by RKBExplorer, too. The graph is used for visualization of relations among information in repositories of partners of project ReSIST [4]. The graph visualizes with notes only resources of same type. These nodes are connected with curved oriented edges that represent relations among them. Problem is that these edges do not inform user about type of relation, only that any relation exists. This problem is partly solved by GOLEM, which visualize hierarchical relations from gene ontology [9]. This graph layout is very clear and readable, but it is usable only for the domain specific visualization. The problem with displaying of the type of a relationship in graph is completely solved in IsaViz with predicates' labels over edges. This tool is primarily focused on browsing and editing of RDF models, so its user interface is mostly oriented to specialists in the sphere of ontologies not for conventional users. It displays complete graph at once, instead of gradual expanding of graph. This can make graph unreadable for large RDF models or models with height density of relations among resources.

3. Navigation Approach

We improve information discovery in the Semantic Web for end users via adaptive views organized into a hierarchy, which respects all phases of users' navigation on the Web – from support of desired result identification by hierarchical clusters view of faceted browser results, to browsing of similar and related resources (results) by graph views on resources and their relations.

3.1 Search Results Browsing

Users starts their navigation in information space with finding of desired results by creating of initial search query. After exploring of obtained results they obviously refine their initial queries to obtain more suitable results. Users performs query modifications several times, while they finally find desired result. To decrease their effort to result exploration, we do not presents results (in our case results, photos, of faceted browser Factic) only in simple list or matrix view, but we display results in hierarchical clusters (see Figure 1). This view displays the first-level results clusters and their the most significant sub-clusters with thumbnails of results. By double clicking on cluster, its sub-clusters with results from next level are shown.

3.2 Resource Exploration

Details view about the result is invoked by double-clicking on its thumbnail in clusters view. This view is automatically adapted to type of results, so users see details about the result in view to which they are habituated from

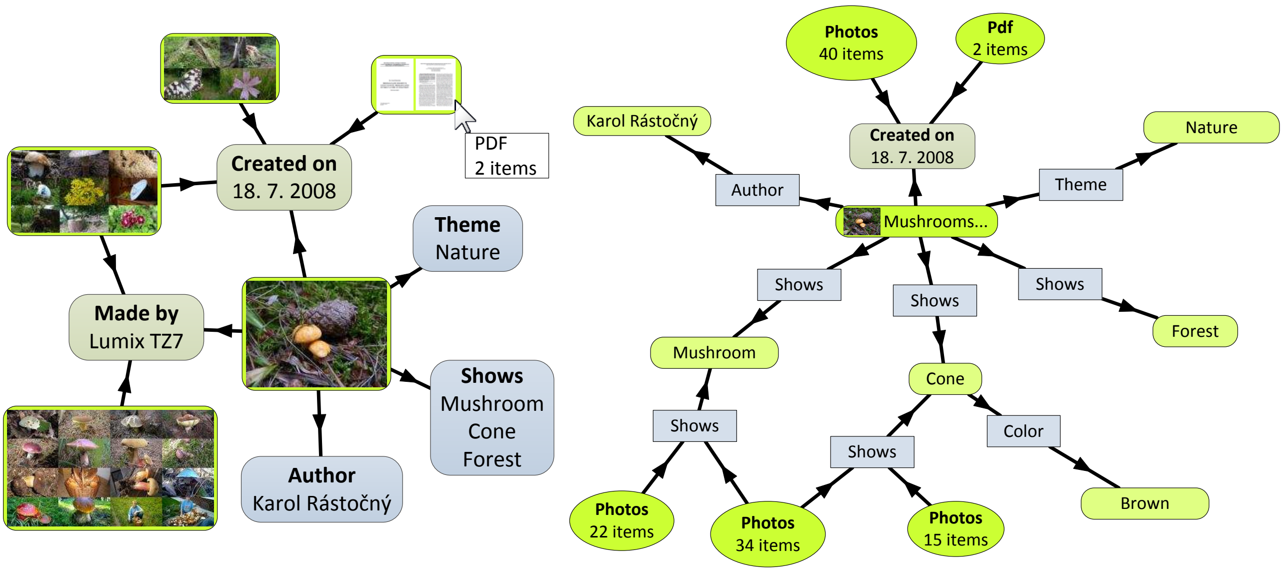

Figure 2: Visualization of the object attributes graph showing fewer nodes improving readability (left) and the RDF graph showing attributes of objects with the values of the original object's attributes, also allowing users to change the context (right).

other, frequently used applications. For example in case of photo, main part of view contains photo and the rest of view contains table of photo's properties and download links in available resolutions and file-types.

3.3 Navigation in Related and Similar Resources

After detail exploration of desired result, users obviously continue with navigation in related resources. In case of classical web pages, they provides this by clicking on links on the page. In case of the Semantic Web, where resources are not classical web pages, users are commonly navigated through links to resources that are properties of initial resource (found result). This type of navigation hides history information (from where and how users have come here) and requires from users to pass though details view about uninteresting resources (e.g. when users want to see photos from same author, they have to open details about author and then identify list of photos in the view).

We support navigation in related results by view-based navigation in graph visualization of related results and their properties based on the concept of web browsing based on graph visualization [7]. Navigation in related and similar resources based on graph visualization helps users to understand relations among resources. The second benefit is history persistence, when all visited nodes stay displayed in graph, so users can return directly to previously visited resources. But this concept of navigation has two main problems. The first problem is that graph becomes unclear after short usage. To avoid this problem we propose several extensions that decrease number of displayed nodes in the graph and help users to orientate in graph more easily based on *result clustering*, *facet marking*, *next action recommendation* and *adaptability to user's interests.*

The second problem is that graph structure defined in this concept is not well readable for conventional users that are not habitual to reading across three nodes (the first node contains information about subject, the second node informs about type of property and finally the third node contains value of the property) and they prefer straight readability (type of the property and its value should be in one place). We avoid this problem with semantic zoom with four levels: literal attributes graph view, object attributes graph view, restricted RDF graph view and full RDF graph view. These levels represent views on graph. Transitions between them are natural, easily understandable for users and give better preview about related resources to users.

3.3.1 Literal and Object Attributes Graph Views

Literal attributes graph view is the first view of navigation. This view is firstly displayed to users after items detail view, when users make decision that they want to start navigation from the selected result. This view contains only literal attributes of results. When users zoom in to the graph, the view is changed to object attributes graph. Unexpanded literal attributes are removed from view and nodes that represent object attributes of results are displayed during this change.

At the beginning of navigation, literal and object attributes graphs consist of the node with initial result (selected by the user) and several nodes around it, that represent its attributes. After expansion of attributes nodes, new results with same attributes are displayed. New results are automatically clustered by their type (e.g. photo, pdf). After that, users can take out one or more results from cluster or brows results in the cluster in hierarchical cluster view.

In these views are nodes with results represented by results' thumbnails. Nodes with attributes integrate types of properties and values of these properties. It means that in visualization of attribute node is displayed label of the property on the top, and the labels of property's values are displayed bellow it (see Figure 2). Nodes representing clusters are shown as logarithmically sized ovals depending on the number of results in the cluster. They contain information about type of clustered results, their count and optional thumbnail collection (if possible).

3.3.2 RDF Graph View

Object graph view has good readability for conventional users, but it does not allow users to change context of the navigation to objects that are values of properties, because they are joined with property in one node (e.g. change concept of navigation from photos to authors). This problem is solved via the next level with restricted RDF graph view. Transaction from object attributes graph view to restricted RDF graph view invokes dividing of attribute nodes to two nodes: the node with type of property and object node, which represents its value (see Figure 2).

With next application of zoom in, view is changed to full RDF graph. This view is obtained from restricted RDF graph view by disabling adaptation to users' interests and by enrichment with nodes hidden by adaptation. Graph displayed in this view can be large with a lot of nodes. Therefore it is intended for domain specialists, who want to achieve overview about data in the repository.

3.3.3 Users Interaction with Graph

We proposed users interaction with graph in such a way that users can perform all their actions right in the graph by a mouse, without necessary to use a keyboard or tool buttons in the tool-bar:

- *Node expansion and roll up* – nodes connected to expanded (rolled up) node are added (removed) to graph by double-clicking on the node or by zoom in (zoom out) through rotation of the mouse scroll wheel;
- *The result extraction from the cluster* – by results cluster node expansion, results in the cluster are displayed in hierarchical cluster view. By double-click on the result, this result is extracted from cluster;
- *Node hiding* – by clicking on the red cross in the right upper corner of the node, the node with its connections is hidden from view (Figure 3);
- *Lock node* – the node's position can be locked by clicking on the lock in the right lower corner of the node (Figure 3), that preserves its movement by layouting algorithm;
- *Facet marking* – by facet (the node's attribute) marking, users can specify if newly added results by node expansion have to or may not have marked properties. Facet marking is possible by three-state checkbox in the left lower corner of the node (Figure 3).

3.3.4 Next Action Recommendation

We propose next action recommendation in the graph navigation as supporting tool, which helps users to orientate in the graph. The recommendation is based on a comparison of user's activities in current session and activities of users in other already finished sessions. We utilize activity flow graphs for this comparison.

Activity flow graphs are built apart for each session, whereby we collect four base activities: node expansion, roll up, facet marking and node hiding. One node of this graph contains normalized action sequence, so that all action sequences that generate equivalent graph visualization have same representation. If a situation where new

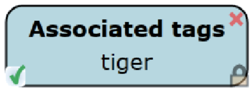

Figure 3: Control elements of the node: node hiding (right upper corner), node lock (right lower corner) and facet marking (left lower corner). They are shown on the mouse hover.

actions sequence has same representation as previous one occurs, previous one action sequence is set as active node (active action sequence).

Edges in activity flow graph represent transition between action sequences by application of one action. Edges' weights are set to 1 by default. When last action sequence in active node of activity flow graph has leaded to display details about result, all edges' weights in path to root of activity flow graph are changed to 2.

With substitution of action sequence with search query we obtain query flow graphs [1]. In recommendation from query flow graphs, their restriction to sub-graphs and establishing of their distances is turned out as advisable [2], [3]. We utilize this in our action flow graphs, where we compare graphs only by paths from active nodes to their roots. Based on this, we count weights of possible next actions and after that we select top most weighted actions.

4. Evaluation

We have performed usability evaluation of our prototype with ten volunteers with different computer skills. Each volunteer had received a brief instruction about the basic functionality of our prototype. After that, two specific navigation goals were given to them and we observed their actions and asked them to explain their actions. The first task was aimed to teach users how to work with the tool and contains few subtasks with the intention of navigate user to goal. The second task was more abstract and its goal was evaluate usability of our tool for real-world users.

After completion of these goals, they obtain short questionnaire with few questions aimed to usability, graphical user interface, provided functionality and quality of created clusters of results. Expected answers to these questions were values from 1 to 5, similarly to school classification.

After experiments we compare number of logged user actions and time they need to solve tasks with reference values (see Table 1) that was obtained by simulation of the most probable scenario of tasks' solutions. On an average, participants performed about 78% more actions than reference value during solving of the first task. This difference was caused by users' exploration of tool's features. We obtained better results during second task. To find solution of the second task participants needed on an average only about 22.5% actions more than reference value. Solving of both tasks took them one time more time than we measured out. We consider these results as really good, because we selected group of people that do not know our dataset and they had to explore and learn it during solving of tasks.

By answers and users behaviour analysis we determine, that users do not have any problems with interaction with

Table 1: Evaluation of measured values during experiment.

	1^{st} task		2^{nd} task	
	Actions	**Time**	**Actions**	**Time**
Minimal value	18	109 s	5	49 s
Maximal value	43	306 s	27	197 s
Average value	28.3	174 s	14.7	125 s
Standard deviation	7.9	64.04 s	6.4	59.41 s
Reference value	16	84 s	12	60 s
Difference	+76.88%	+107.14%	+22.5%	+108.33%

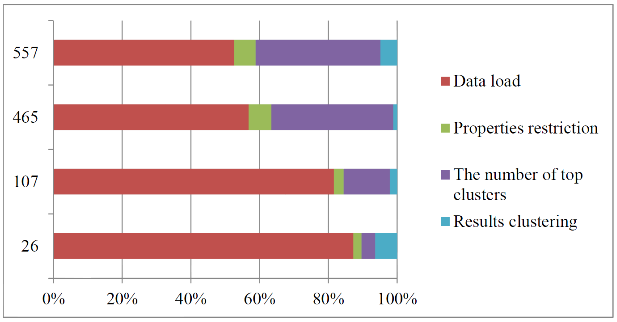

Figure 4: Analysis of processing time of clustering by dependence from the number of results.

implemented tool (classification 2.2 ± 0.63) and that navigation in graph visualization was natural for them (classification 1.6 ± 0.84). We notice only small issue that users naturally filtered out nodes with distance from their actual center of interest more than two edges. This observation we can utilize in graph visualization by automatic hiding of outlying nodes, what can increase efficiency of user navigation.

Participants responded, that results clusters was correct (classification 1.2 ± 0.42), but they noticed, that some results had wrong classification to clusters, but this was caused by errors in dataset and by unbalanced description of photos in dataset by their details.

We have noticed problem with response time of clustering. The response time was good for the number of results less than 200, but response time grew rapidly with number of results. We determine problem with response time of used database (we use Seasame) and determining of the number of top clusters by analysis of processing time of clustering (Figure 4). We can solve problem with the number of top clusters by replacing of currently used algorithm based on algorithm from dynamic SVD clustering [6] by straight calculation from the number of restricted properties.

5. Conclusions

We propose a novel approach to navigation in the Semantic Web based on adaptive views that helps users in all phases of navigation in the Semantic Web, from identifying of initial result to navigation in related and similar resources. Navigation starts with hierarchical cluster view, which helps users to identify wanted result in set of faceted browser's results. After selection of the result, adaptive view with detailed information about it is displayed. Users usually want to know more about related and similar resources to the result. We support this by navigation in graph visualization.

We evaluated our approach via several experiments with real-world users, during that we collected information about participants' interaction with our implemented prototype and information about their opinions and expectations on the provided tool. On the basis of the evaluation's results we are able to pronounce, that our proposed method is natural and simply understandable for users.

Acknowledgements. This work was partially supported by the Scientific Grant Agency of Slovak Republic, grant No. VG1/0675/11.

References

[1] P. Boldi, F. Bonchi, C. Castillo, D. Donato, A. Gionis, and S. Vigna. The query-flow graph: model and applications. In *Proc. of the 17th ACM Conf. on Inf. and Knowledge Managmt.*, CIKM '08, pages 609–618, New York, NY, USA, 2008. ACM.

[2] P. Boldi, F. Bonchi, C. Castillo, D. Donato, and S. Vigna. Query suggestions using query-flow graphs. In *Proc. of the 2009 Workshop on Web Search Click Data*, WSCD '09, pages 56–63, New York, NY, USA, 2009. ACM.

[3] I. Bordino, C. Castillo, D. Donato, and A. Gionis. Query similarity by projecting the query-flow graph. In *Proc. of the 33rd Int. ACM SIGIR Conf. on Research and Development in Information Retrieval*, SIGIR '10, pages 515–522, New York, NY, USA, 2010. ACM.

[4] H. Glaser, I. C. Millard, and A. Jaffri. Rkbexplorer.com: a knowledge driven infrastructure for linked data providers. In *Proc. of the 5th European Semantic Web Conf.*, ESWC'08, pages 797–801, Berlin, Heidelberg, 2008. Springer-Verlag.

[5] G. Marchionini. Exploratory search: from finding to understanding. *Commun. ACM*, 49:41–46, April 2006.

[6] G. Mecca, S. Raunich, and A. Pappalardo. A new algorithm for clustering search results. *Data Knowl. Eng.*, 62:504–522, 2007.

[7] A. Rakovský. Web browsing based on graph visualization. Bachelor Thesis, STU in Bratislava, 2009. (in Slovak).

[8] m. schraefel, M. Wilson, A. Russell, and D. A. Smith. mspace: improving information access to multimedia domains with multimodal exploratory search. *Commun. ACM*, 49:47–49, 2006.

[9] R. Sealfon, M. Hibbs, C. Huttenhower, C. Myers, and O. Troyanskaya. Golem: an interactive graph-based gene-ontology navigation and analysis tool. *BMC Bioinformatics*, 7(1):443, 2006.

[10] M. Tvarožek. Exploratory search in the adaptive social semantic web. *Information Sciences and Technologies Bulletin of the ACM Slovakia*, 3(1):42–51, March 2011.

[11] M. Tvarožek and M. Bieliková. Reinventing the web browser for the semantic web. In *Proc. of the 2009 IEEE/WIC/ACM Int. Joint Conf. on Web Intelligence and Intelligent Agent Technology - Volume 03*, WI-IAT '09, pages 113–116, Washington, DC, USA, 2009. IEEE CS.

[12] F. van Ham and B. Rogowitz. Perceptual organization in user-generated graph layouts. *IEEE Trans. Visualization and Comp. Graphics*, 14(6):1333 –1339, nov.-dec. 2008.

[13] R. W. White and G. Marchionini. Examining the effectiveness of real-time query expansion. *Inf. Process. Manage.*, 43:685–704, May 2007.