Automatic Exhibition Generation Based on Semantic Cultural Content

Eetu Mäkelä, Osma Suominen, and Eero Hyvönen

Semantic Computing Research Group (SeCo),
Helsinki University of Technology (TKK) and University of Helsinki
first.last@tkk.fi, http://www.seco.tkk.fi/

Abstract. In this paper, we argue for a need to shift focus in semantic search from the items themselves to using them as lenses to wider topics. A system for doing this in the cultural heritage domain is presented, duplicating on the web the way exhibitions in the real world are organized. An interface for specifying such exhibitions is presented, combining a general narrative pattern with semantic autocompletion and the novel concept of domain-centric view-based search. This also solves a number of problems view-based search has previously encountered in the cultural heritage domain. Presented also are multiple visualizations for the exhibition, supporting the user in making sense of the data and in doing exploratory search.

1 Introduction

Traditionally, Internet search has been about finding a document or documents that answer the question posed by the searcher. Semantic Web search systems have mostly also held this viewpoint [1], using properties and concepts in domain ontologies to locate search objects annotated with them. For semantically annotated content analogous to text documents, this works adequately, but for qualitatively different material, it creates problems. To understand why, one must take a step back to look at information needs.

The many classifications of information needs [2–7] all agree that there is a major partition between lookup queries like "For my meal, I need a *white wine* with a *spicy flavor*" and more general information needs such as "tell me all about *spicy white wines*". The former focuses on selecting, fact finding and question answering, while the latter deals with the more general objective of learning and investigation, containing in addition to searching also tasks such as comparison, interpretation, aggregation, analysis, synthesis and discovery [8]. Depending on domain, at least a significant part (22% [9]), or even the majority (70% [10], 67% [5]) of enquiries for information relate to these more general learning instead of spot queries.

Despite this, search research has only recently begun to move to this expanded domain, termed exploratory search [8]. We propose that a major reason for this is that as long as the information is encoded only inside documents,

learning and investigation searches are adequately catered for by the same functionality as fact finding, i.e. locating all matching documents and then perusing each for relevant data [6].

For semantically annotated content other than information documents, the situation is different. Often the useful information is not the object itself, but the relation between the object and the ontological resources associated with it. Now, for question answering such as what wine to have with a particular food, the answer is still a particular object with particular characteristics, and the old paradigm still works. For the more general type of queries, on the other hand, typical semantic web object databases fall short, as they contain no singular exposition about, e.g. "French spirits".

However, if looked at from another perspective, the data contains ample information to answer someone wanting to know about French liquors. It is merely encoded differently, distributed across the multiple object annotations and ontologies. To pull this information out, one must move the focus from individual items to the set of objects with particular properties as a whole, and even further. What one actually wants is to look at the combination of the domain concepts "French" and "spirits" through the lens of the items.

Actually, if an interface capable of such can be created, the pieced nature of the information becomes an advantage, as the pieces can be combined to shed light on a much wider variety of topics than anyone could write an explanatory article on. This capability is even further enhanced if the database contains material of multiple different kinds. For example in the cultural heritage domain, with suitable material, one could learn not only about 19th century Finnish crafts, 19th century Finnish paintings etc., but actually of the 19th century Finland as a whole.

Based on this analysis, we argue that to support exploratory search tasks, Semantic Web application designers need to shift focus from object location to the creation of structured, domain-centric presentations based on those items.

2 Looking at Culture Through Its Products

Luckily for interface designers in the cultural heritage domain, there is already a real world counterpart for this functionality to take inspiration from. What is wanted is very similar to how exhibitions in real-world museums function, presenting a particular temporally, spatially and functionally constrained aspect of culture through its objects and art. As such parallels are an excellent cue for understanding the structure of an information presentation, we decided to make as much use of this as possible when designing the interface for our CultureSampo[1] [11] cultural heritage portal.

Our idea is to let users create virtual exhibitions that mimic the way real museums are organized, containing themed exhibition rooms of items and displays that together, through the objects, tell the story of a particular subject.

[1] http://www.kulttuurisampo.fi/

Our implemented system combines an exhibition specification interface based on view-based query constraining with multiple visualizations grouping the items according to domain facets the user is interested in. In the following, both of these components will be discussed further in their own chapters.

2.1 Specifying the Desired Exhibition

The CultureSampo portal is aimed at the general public. Therefore, our exhibition generation interface had to be as easy to use and understand as possible, while still allowing for a wide variety of different presentations to be generated. To accomplish these goals, we first set the parameters for what kind of exhibition definitions had to be possible. Analyzing real exhibitions, we created a general but verbally understandable structural pattern for describing them, on which we could build our interface. The pattern, with each part optional, is:

> Tell me about *item type*
> related by *role* to *domain concept* [and ...]
> organized by *classification+role* [and *classification+role*]".

While constrained and procedurally structured, this pattern still allows for a wide range of exhibitions to be specified, from e.g. "Tell me about weapons" to "Tell me about everything related to 19th century Finland and agriculture, organized by item type and purpose of use" and "Tell me about toys manufactured in China, organized by time of manufacture and place of use". Figure 1 shows this in our actual interface. On the left are the exhibition specification controls, layed out to directly reflect our narrative structure.

Domain-Centric View-Based Constraining For the selector components used to fill this pattern, we looked to the recently popularized [12–16] paradigm of view-based search (also known as faceted browsing) combined with semantic autocompletion [17]. Of these, view based search is based on organizing the search data into multiple categorizing views and then picking categories as constraints from the views and has already shown good promise for fulfilling learning type search needs [18].

For our particular needs, the paradigm has a number of user benefits [19]. First, because the collection is visualized along different categorizations, the user is immediately familiarized with its contents and the way they are organized. Functionally, the user gets information on what the possible constraints are and how selecting them will affect the result set. Second, the multiple viewpoints allow the user to start constraining from the perspective most familiar to them. Finally, this visualization already intuitively shows the wider context in which the result set lays, thereby contributing to the users ability to answer questions of the result set as a whole, and not just of individual item.

In addition to interface benefits, the paradigm fits Semantic Web data well. The rich metadata in semantic databases is just the sort of multifaceted data whose exploration the paradigm supports. Also, because the metadata values are

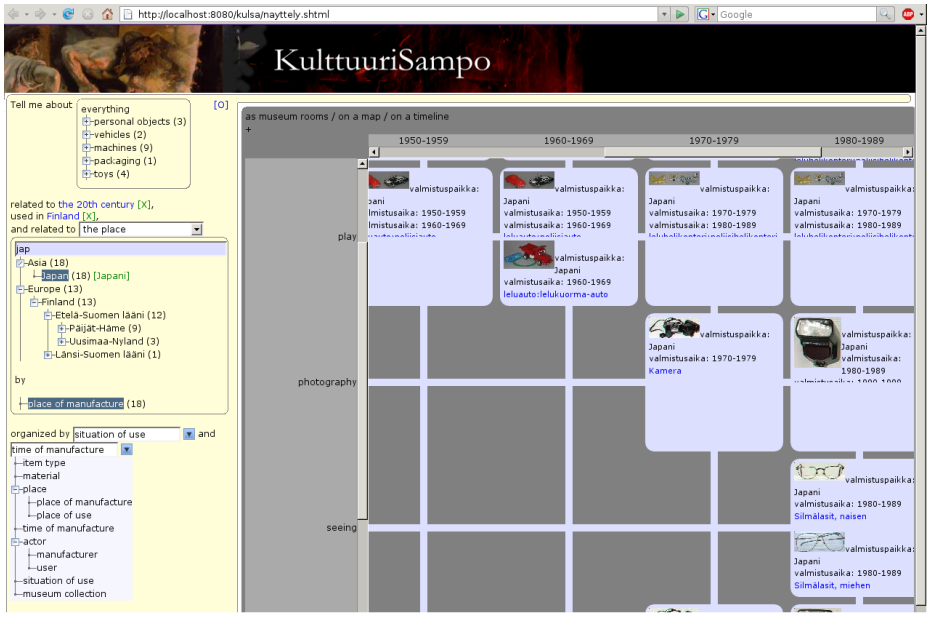


Fig. 1. The CultureSampo user interface, with important elements manually translated into English. The exhibition specification interface is located on the left, while the exhibition itself is visualized on the right. Showing is an exhibition on the types of items Japan exported to Finland in different parts of the 20th century.

resources organized in ontological hierarchies, they provide an excellent basis for creating usable, well-structured categorizing views.

Traditionally in Semantic Web view-based search systems views have been formed by selecting a property, such as "place of manufacture", and enumerating all the values of that property as selections. In the cultural domain, this has caused problems, as there are typically many content types with differing properties such as "mentioned place" (poem) and "depicted place" (painting, photograph) [14]. Fortunately, our move from the objects to the domain concepts presented us with a natural solution, the novel variation of domain-centric view-based search [20]. Here, the properties are relegated to a secondary role, and the views were built instead based on the ontological ranges of those properties, i.e. the set of topical domain ontologies. In CultureSampo, we ended up with nine views: object types, places, times, actors, events, styles, materials, techniques and museum collections, with attached properties such as place of manufacture, depicted place and place of birth.

In a prior version [11], we discarded the properties from the user interface completely, and our views selected all items related in any way to the domain concepts (e.g. show anything related in any way to Poland). However, without any reference to the properties, the users were lost as to what a selection did and why any particular item was included in the result set. In addition, the

expression power of the interface diminished, as one could no longer e.g. search for items made in Japan but used in Europe.

These problems were solved by two measures. First, in the presentation, for each item an explanation is included of the property-concept relationships that places that item in the result set. Second, the properties were brought back to the views, but in a different form, shown in the place facet of figure 1. Now, a view consists of two selectors: one for selecting the domain concept and another for limiting based on the role (property) that the concept has in relation to the search items. Here, the user is free to search both with and without specifying a role, actually increasing the expressivity of the view-based search paradigm.

In CultureSampo the views are not all constantly visible. This is because here they are used as selectors in the context of a larger pattern, which we wanted to emphasize. Showing many views at once by default would have cluttered the screen, reducing intuitive grasp of the interface. Instead, by default visible are two views, one static for constraining by item type, and another for constraining by a domain concept. The domain view visible in any given moment is selected from a dropdown menu (shown on the left in figure 4(a)). In addition, power-users can also bring up further concurrent views.

View-Independent Semantic Autocompletion The multiple views in the view-based search paradigm make it easy for users to browse their options. However, for users knowing precisely what they want, a shortcut and a single point of entry is desired. In our system, this is accomplished by a semantic autocompletion [17] component, shown in figure 2.

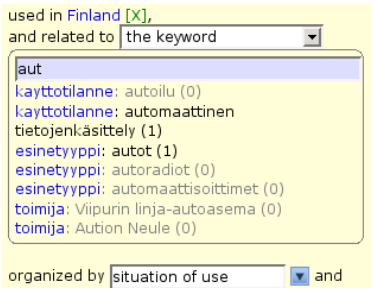


Fig. 2. The view-independent semantic autocompletion component of the CultureSampo exhibition specification interface.

Here, the user merely types in what they are looking for, and the system instantly responds with matching keywords to be used as possible constraints. These are both annotations directly related to the items, as well as matching selections in any of the facets. If the keyword typed gives sufficient specificity for the user, it isn't even necessary to make any further selections, as the query state is also instantly updated, using the union of the matches as a constraint. This

makes it possible for a user to interact with the system in a more experimenting way, typing in a keyword that pops into their mind and immediately seeing if the portal contains any related material, as well as what kind of exhibition it generates.

These keyword-search derived constraints can also be combined with those selected from domain views. For further supporting in-between user behaviours, all the domain views internally support a different form of semantic autocompletion, with the results shown directly in their hierarchical view context. This functionality is depicted in the place facet of figure 1.

2.2 Visualizing the Exhibition

As the user makes choices constraining the material, also the exhibition view is updated. Here, our primary association strived for is of a typical museum, with themed floors and rooms of exhibits, combined with custom presentations.

For the museum room visualization, the same categorizing view structures used for selection are utilized. The idea is simply to project the items in the result set onto a two dimensional matrix whose rows and columns are comprised of a flattened list of concepts in the two domain facets chosen for organization. This way, each cell in the matrix corresponds to room combining two themes, such as "18th century agriculture", followed in one dimension by "19th century agriculture", and "18th century hunting" on the other. This matrix is then visualized, either as is for a single-floor museum complex view depicted in figure 1 or row by row, for a more traditional floor and room museum plan, shown in figure 3. While the latter plan allows us to eliminate empty rooms on a floor by floor basis thereby optimizing display area, the single-floor view allows one to also see more large-scale structural changes. In figure 1, for example, one can see how in 1950-1970 most Japanese-made items that made their way into Finland were toys, but beginning in the 70's there is an increase in the import of high-tech products. Both visualizations are scrollable where they do not fit in the screen at once.

For particular domains, special presentations particularly suited to them are available [21]. In our current system, these are a timeline visualization for the time facet and a map visualization for the place facet. These are shown in figure 4. In both of these visualizations, the second dimension, if specified, is expressed by marker coloring.

3 System Architecture

A primary design requirement for the CultureSampo exhibition interface was to allow the user to explore the data in the portal in a highly interactive and experimenting fashion. To support this, the interface had to be very responsive, updating all views instantly to match user commands. In our implementation, this is achieved through a highly optimized view-based search engine combined

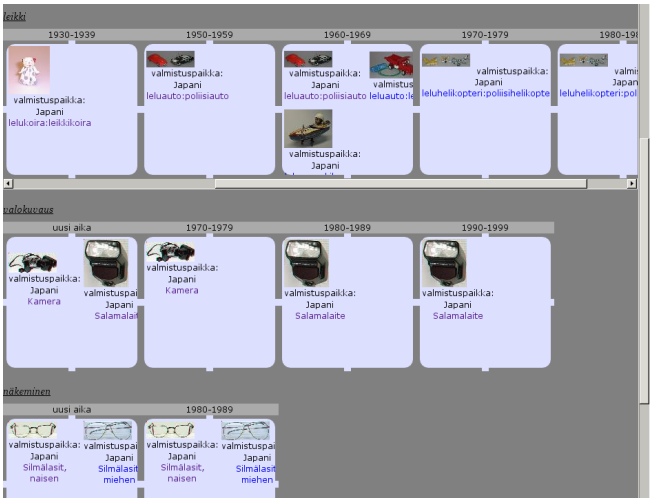
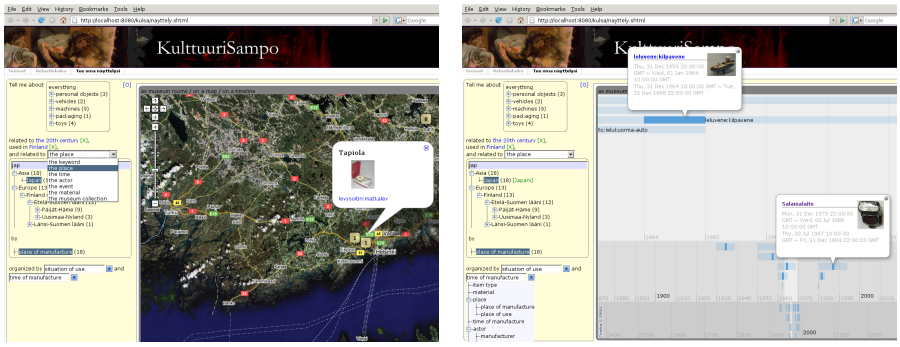


Fig. 3. The results organized according to museum floor and room in CultureSampo



(a) map (b) timeline

Fig. 4. Special presentation visualizations in CultureSampo

with logic that minimizes the amount of data needed to be sent at any time between the server and the client browser.

Our view-based search engine makes use of existing tools for implementing efficient indices. The hierarchical view trees are indexed using an interval indexing scheme [22] inside an SQL database. This allows transitive constraints to be processed using a single SQL index scan. For textual matching, either SQL or Apache Lucene[2] indices are used depending on specific needs. On top of these indices, there is a custom processing component that gathers all SQL and Lucene constraints respectively to single executable queries, as well as implements partial query caching for intersection queries. This last functionality is extremely

[2] http://lucene.apache.org/

important for throughput, as an overwhelming majority of queries in view-based search are those updating hit counts in the views, i.e. intersecting the current query with potential future constraints.

To further increase responsiveness, there is logic on both the server and in javascript on the client that minimizes the amount of information that needs to be sent over the network. For example, the children of a node in a view tree are only sent to the client on request. However, the server also keeps a list of nodes that have been sent. This allows their hit counts to be sent in a single batch without explicit individual querying. It also makes it possible for the server-side semantic autocompletion component to know if matched tree nodes are already known by the client or need to be sent along with the text-match response itself. The information itself is sent as javascript objects, automatically mapped from JavaBeans with Direct Web Remoting[3]. This additionally reduces the amount of information sent, as all layout of the results is done on the browser. For this side of the interface, the system makes use of the widget functionality provided by the Dojo javascript library[4]. This allows us to use general and reusable widgets [23] with attached HTML and CSS templates for the views and other interface components, instantiated on need as they are brought up.

4 Related Work

Many of the view-based search systems already mentioned [12–16] support exploratory search to some extent. Of these, one in particular needs to be mentioned. The Exhibit system [12] by the Simile project has clearly been designed with similar explorative search goals in mind and has an interface strikingly similar to our own, even down to providing map and timeline visualizations. Relating to our exhibition room visualization, they provide a grouping similar to our floors plus rooms plans, but not our two-dimensional matrix.

However, there is also a major difference between the two systems, indicative of the major importance our shift of focus has had. That is, Exhibit still follows the traditional viewpoint of concentrating only on the search objects themselves. Being based on traditional view-based search, the system only really supports a single item type at a time, being susceptible to the problem of view proliferation as the number of annotation schemas and consequently projected properties increases. Not utilizing domain-centric view-based search also means that its grouping and visualization capabilities cannot be used to shed light on domain concepts — a major idea of our system.

Architecturally, the two systems are quite different. Exhibit exists completely in javascript on the client side and supports only a few thousand items and simple flat classifications, being intented as a lightweight solution for casual users to publish their structured data on the web. Our system on the other hand is intented for publishing much larger collections and includes optimized interplay between the client and server to support this.

[3] http://getahead.org/dwr/
[4] http://dojotoolkit.org/

On the subject of multi-type view-based search in the cultural domain, the /facet -system [14] utilized in the MultimediaN project [24] also tackles this problem. Their choice is mostly to simply promote the "item type" facet as a first choice, as choosing from it drastically reduces the amount of property facets available. However, this only alleviates the problem, and diminishes the freedom of the user in starting from the facet most natural to them. Otherwise, the system also contains a functionality through which complex constraints spanning multiple types can be formed, allowing one to specify for example a pattern such as "Find all paintings by artists living in Paris". While this was deemed too complex for our needs, in other fields it might supplement our pattern-filling approach nicely.

5 Conclusions and Future Work

In this paper, we have argued the need for a shift of focus in semantic search from item location to presentation generation and support for exploratory search. In particular, we argue that often what is interesting in semantic databases are not the items themselves, but how they shed light on a theme described by a particular combination of domain concepts.

For the cultural heritage domain, museum exhibitions offer a suitable parallel to this idea. We have taken advantage of this in our CultureSampo portal, combining an intuitive, yet expressive exhibition generation interface with different kinds of exhibition visualizations. On the exhibition generation side, our major contribution is the narrative query pattern for forming exhibitions combined with the concept of domain-centric view-based search, which allow us to cater to both searching for items having particular properties, as well as pure domain exploration.

On the exhibition visualization side, we have created a simple, general-purpose visualization, as well as complemented it with special purpose visualizations. Even these simple visualizations already give significant support for a user wanting to make sense of the data. However, here there is still also much more that could be done. Our next user interface functionality will be to allow the user to select some rows or columns from the matrix for specific comparison and study. It may also be possible to aid such comparison work by automatically extracting from the data meaningful differences and similarities between neighboring exhibition rooms, such as "18th century agricultural items are more often made of wood than 19th century ones".

While our current user interface has been created exclusively for the CultureSampo portal, the actual architecture and functionality is very general, modular and configurable. The interface has also already been tested with alternative materials. Pursuant to this, we are studying ways to make the application configuration as effortless as possible, in order to provide the functionality we have over Exhibit also for generic Semantic Web content.

In addition to implementing these new functionalities, we also plan to conduct more thorough understandability and usability tests on this interface, as compared to competing choices.

Acknowledgements

This research is part of the National Finnish Ontology Project[5] (FinnONTO) 2003-2007, funded by the National Technology Agency (Tekes) and a consortium of 36 companies and public organizations.

References

1. Hildebrand, M., van Ossenbruggen, J., Hardman, L.: An analysis of search-based user interaction on the semantic web. Technical report, Centrum voor Wiskunde en Informatica (NL) (2007)
2. Wilson, T.D.: Information needs and uses: fifty years of progress. In Vickery, B., ed.: Fifty years of information progress: a Journal of Documentation review, London, Aslib (1994) 15–51 http://informationr.net/tdw/publ/papers/1994FiftyYears.html.
3. Belkin, N.J., Marchetti, P.G., Cool, C.: Braque: design of an interface to support user interaction in information retrieval. Inf. Process. Manage. **29**(3) (1993) 325–344
4. Cool, C., Belkin, N.J.: A classification of interactions with information. In Bruce, H., Ingwersen, P., Vakkari, P., eds.: Emerging frameworks and methods; Proceedings of the 4th international conference on conceptions of Library and Information Science (COLIS4), Greenwood Village, CO: Libraries Unlimited (2002) 1–15
5. Choo, C.W., Detlor, B., Turnbull, D.: Information seeking on the web: An integrated model of browsing and searching. First Monday **5**(2) (2000) http://firstmonday.org/issues/issue5_2/choo/index.html.
6. Jansen, B.J., Smith, B., Booth, D.: Learning as a paradigm for understanding exploratory search. In: Proceedings of the SIGCHI 2007 Exploratory Search and HCI workshop. (2007)
7. Anderson, L., Krathwohl, D.: A taxonomy for Learning, Teaching and Assessing: A Revision of Bloom's Taxonomy of Educational Objectives. Longman, New York (2000)
8. Marchionini, G.: Exploratory search: from finding to understanding. Commun. ACM **49**(4) (2006) 41–46
9. Cole, P.F.: The analysis of reference query records as a guide to the information requirements of scientists. Journal of Documentation (14) (1958) 197–207
10. Wilson, T.D.: Current awareness services and their value in local government. In: Proceedings of the 40th Confress of the Federation Internationale de Documentation. (1980)
11. Hyvönen, E., Ruotsalo, T., Häggström, T., Salminen, M., Junnila, M., Virkkilä, M., Haaramo, M., Kauppinen, T., Mäkelä, E., Viljanen, K.: CultureSampo—Finnish culture on the semantic web. The vision and first results. To appear in: K. Robering (Ed.), Information Technology for the Virtual Museum. LIT Verlag, 2007 (2007)

[5] http://www.seco.tkk.fi/projects/finnonto/

12. Huynh, D., Karger, D., Miller, R.: Exhibit: Lightweight structured data publishing. In: 16th International World Wide Web Conference, Banff, Alberta, Canada, ACM (2007)
13. Schraefel, M.C., Wilson, M., Russell, A., Smith, D.A.: mSpace: improving information access to multimedia domains with multimodal exploratory search. Commun. ACM **49**(4) (2006) 47–49
14. Hildebrand, M., van Ossenbruggen, J., Hardman, L.: /facet: A browser for heterogeneous semantic web repositories. In: The Semantic Web - Proceedings of the 5th International Semantic Web Conference2006. (2006) 272–285
15. Oren, E., Delbru, R., Decker, S.: Extending faceted navigation for rdf data. In: The Semantic Web - Proceedings of the 5th International Semantic Web Conference2006. (2006) 559–572
16. Mäkelä, E., Hyvönen, E., Sidoroff, T.: View-based user interfaces for information retrieval on the semantic web. In: Proceedings of the ISWC-2005 Workshop End User Semantic Web Interaction. (2005)
17. Hyvönen, E., Mäkelä, E.: Semantic autocompletion. In: Proceedings of the first Asia Semantic Web Conference (ASWC 2006), Beijing, Springer-Verlag, New York (2006)
18. White, R.W., Muresan, G., Marchionini, G.: Evaluating advanced search interfaces using established information-seeking models. Information Processing and Management (2007) to appear in special issue on Evaluating Exploratory Search Systems.
19. Mäkelä, E.: View-based search interfaces for the semantic web. Master's thesis, University of Helsinki (2006)
20. Mäkelä, E., Ruotsalo, T., Hyvönen, E.: Domain-centric view-based search. Submitted for review. (2007)
21. Alonso, O., Baeza-Yates, R., Gertz, M.: Exploratory search using timelines. In: Proceedings of the SIGCHI 2007 Exploratory Search and HCI workshop. (2007)
22. Christophides, V., Plexousakis, D., Scholl, M., Tourtounis, S.: On labeling schemes for the semantic web. In: WWW '03: Proceedings of the 12th international conference on World Wide Web, New York, NY, USA, ACM Press (2003) 544–555
23. Mäkelä, E., Viljanen, K., Alm, O., Tuominen, J., Valkeapää, O., Kauppinen, T., Kurki, J., Sinkkilä, R., Känsälä, T., Lindroos, R., Suominen, O., Ruotsalo, T., Hyvönen, E.: Enabling the semantic web with ready-to-use mash-up components (2007) Submitted for review.
24. Schreiber, G., Amin, A., van Assem, M., de Boer, V., Hardman, L., Hildebrand, M., Hollink, L., Huang, Z., van Kersen, J., de Niet, M., Omelayenko, B., van Ossenbruggen, J., Siebes, R., Taekema, J., Wielemaker, J., Wielinga, B.J.: Multimedian e-culture demonstrator. In: The Semantic Web - Proceedings of the 5th International Semantic Web Conference2006. (2006) 951–958