Enabling the Semantic Web with Ready-to-Use Web Widgets

Eetu Mäkelä, Kim Viljanen, Olli Alm, Jouni Tuominen, Onni Valkeapää, Tomi Kauppinen, Jussi Kurki, Reetta Sinkkilä, Teppo Känsälä, Robin Lindroos, Osma Suominen, Tuukka Ruotsalo, and Eero Hyvönen

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

Abstract. A lot of functionality is needed when an application, such as a museum cataloguing system, is extended with semantic capabilities, for example ontological indexing functionality or multi-facet search. To avoid duplicate work and to enable easy and cost-efficient integration of information systems with the Semantic Web, we propose a web widget approach. Here, data sources are combined with functionality into ready-to-use software components that allow adding semantic functionality to systems with just a few lines of code. As a proof of the concept, we present a collection of general semantic web widgets and case applications that use them, such as the ontology server ONKI, the annotation editor SAHA and the culture portal CultureSampo.

1 Introduction

To implement new semantic applications or to extend existing information systems (e.g. a museum cataloguing system) with semantic capabilities requires a lot of functionality dealing specifically with ontologies and metadata. Currently, needed functionalities are typically created for each application individually, requiring a lot of work, time and specific skills. Being able to lower these implementation costs would be hugely beneficial to the adoption of the Semantic Web [1] as a whole. On a general level, there are three tasks in any semantic information environment that need to be handled, either by humans or machines:

Semantic Content Consumption. Searching, browsing, visualizing and otherwise consuming semantic content. In this group belong for example library users in a semantic library environment and visitors of a semantic museum portal [2]. RSS aggregation services are a programmatic example.

Content Indexing. The production of semantic metadata by indexing and publishing content with references to shared vocabularies (e.g. museum curators indexing exhibits). Sometimes, the end-users themselves fill the role of content indexers, as in the social bookmarking site del.icio.us[1] and photo sharing site Flickr[2].

[1] http://del.icio.us/
[2] http://flickr.com/

Ontology Maintenance and Publishing. Creating and maintaining the ontologies used as references for both semantic indexing and retrieval. In organized fields, this is often done by dedicated information workers, but again in the case of Web 2.0 [3] sites, it may be the users themselves that develop their vocabulary in an ad-hoc manner alongside indexing.

We argue that in many cases, tasks in the contexts above contain many common general subtasks. Specifically, we have found at least the following common functionalities: 1) *concept and instance selection*, 2) *semantic linking*, 3) *concept and instance viewing*, and 4) *shared concept and instance storage and maintenance*. To avoid duplicate work and to enable more individuals and organizations to join the Semantic Web, we propose a *web widget*[3] approach, where these functionalities are created and published as ready-to-use software components which can easily and cost-effectively be added to applications.

A web widget is a reusable, compact software component that can be embedded into a web page or application to provide functionality. Most useful web widgets also combine on-line data resources with site data to create mash-ups, such as in usual use cases of the Google Maps and Google AdSense web widgets. Web widgets can also be combined together and published as new components, e.g. with the Yahoo! Pipes service[4]. What makes web widgets very interesting, is that they allow developers to easily add otherwise very complicated or costly features to virtually any application. For example, Google Maps provides a map and satellite image database of the planet Earth combined with search and browsing capabilities for all to use.

Semantic web widgets then, as we envision them, are software components that: 1) are hooked up with either Semantic Web data sources such as ontologies or instance databases, or the processed outputs of other components, 2) offer a single, compact functionality, yet do that as completely as possible (often including user interface elements), 3) are amenable to be combined with other components and data to solve complex problems, and 4) can be easily and cost-efficiently used for adding semantic functionalities to an application.

During our research, we found that while the semantic functionalities we want to implement themselves are general, the best implementation for them varies with the type of the underlying data they are to be hooked up with. For example, selection from a geographical location ontology naturally benefits from map-based user interfaces, actor ontologies need handling of pen names and transliterations [4], while concept ontologies such as the Suggested Upper Merged Ontology SUMO [5] and the health ontology SNOMED CT[5] require other methods. Therefore, we present several widget solutions to each of the tasks based on the different requirements of each content domain.

The context for this work is the goal of creating a national semantic web infrastructure in Finland [6], where critical Semantic Web resources, such as

[3] http://en.wikipedia.org/wiki/Web_widget
[4] http://pipes.yahoo.com/
[5] http://www.snomed.org/snomedct/

ontologies and the web widgets for using them, are published as centralized services.

In the following, as proof of the viability of the idea of semantic web widgets, we first present some of the components we have created for solving common subtasks, and then apply them to one combined mash-up service, the ONKI ontology server, and two end-user applications: the SAHA metadata editor for content creation and the CultureSampo cultural heritage portal. Finally, related work is presented, followed by discussion and suggestions for future work.

2 Common Subtasks in Semantic Applications and Widgets for Solving Them

2.1 Concept and Instance Selection

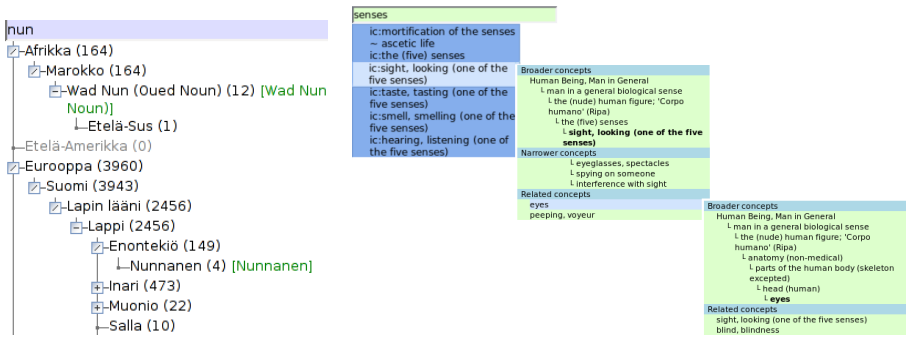
In ontological systems finding and selecting the right concepts and instances is a central task of its own in ontological user interfaces. For end-user applications, any search usually begins by first finding the right concepts with which to do the actual ontological querying [7]. For efficient semantic content indexing, accurate indexing entitities need to be found with as little effort as possible [8]. Also ontology developers need concept search when creating links between concepts, especially when developing distinct, yet heavily interlinked ontologies. In the following, web widgets to provide efficient concept selection in different situations are presented.

Semantic Autocompletion and Context Visualization When the user knows with relative certainty what they are looking for, and the labels of the entities do not overlap much, as in most non-instance vocabularies, a keyword search is a natural way of selecting concepts. For this, we have developed multiple text literal matching semantic autocompletion interfaces [9] that can be hooked into various semantic data sources to provide concept selection functionality. Particular among these are two that also expose the semantic contexts of the concepts matched.

There are two main reasons for desiring such functionality. First, this allows the user to get acquainted with the vocabularies and how they are organized. Second, particularly with large complex interlinked vocabularies, it is never guaranteed that the concept that first occurs to the user is the best one for their task. Showing the ontological context or otherwise derived concept recommendations is a powerful way of gently guiding the user and giving more options.

The first of the context exposing interfaces created, depicted in figure 1(a) shows the autocompletions directly inside a tree. This is applicable when the entities form a hierarchy and this is clearly the most important context for them. A particular case for this is in view-based search, where the view tree is usually already shown for visualization and selection purposes. The second is a navigation widget for exploring the contexts of matched entities, depicted in figure 1(b). Here the ontological context and otherwise derived concept recommendations

are shown to the user as a tree menu, with new levels of context opening when the user mouses over the concepts.



(a) inside a tree (b) with context navigation

Fig. 1. Semantic autocompletion

Map-based Search and Visualization URIs concerning geographic objects might carry coordinate information, e.g. in terms of WGS84 latitude and longitude. We have created a method, *n-point search* [10, 11], for selecting entities based on this kind of coordinate data. A search query in this method is done by pointing out n points on a map. The user clicks on the map and a search polygon is formed accordingly. If an area point of a certain place is found inside the user-given polygon, or a region-defining polygon is found to overlay the search polygon, the region instance is retrieved and added to the results.

We have also taken into account two special cases, namely, where $n = 1$ or $n = 2$. If $n = 1$ a circle is drawn around the point and the places that are inside the circle are retrieved. An alternative treatment would be to simply find the nearest places. If $n = 2$, we create a bounding box where the points are the opposite, e.g. South-West and North-East corners, of the search area. We have implemented the n-point search as a mash-up that itself uses Google Maps[6]. We used SVG [12] for drawing the polygon as a transparent layer on top of the map. In addition to selecting locations, the component is also able to visualize semantic content related to geographical locations on the map.

An example of using the component is depicted in figure 2. The n polygon corners are depicted as small crosses. The system has found from a historical place ontology [13] the municipality of Viipuri in annexed Karjala as it existed in 1906-1921, because its center points are situated within the user-specified search polygon. The municipality is visualized on the map as a red circle. The figure also depicts another useful feature of our component, in that it is able to overlay multiple maps of our own on top of the ones provided by Google [11]. Here, a

[6] http://www.google.com/maps

historical map overlay of the area shows how Viipuri looked at the time specified, which can be constrasted to how the place looks now.

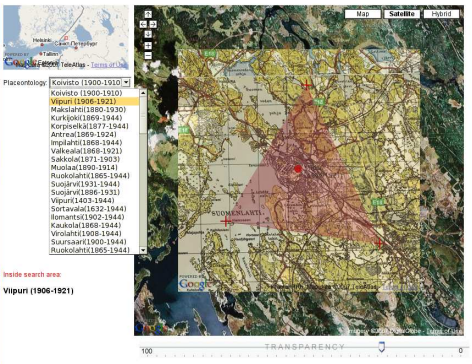


Fig. 2. Search and visualization using location data and overlaid maps.

Fig. 3. A floatlet (encircled) displays links to MuseumFinland.

Multi-Facet Search View-based, or multi-facet search is a paradigm that has recently become prominent as an easy to use interface for querying semantic content [14–16]. Here, the idea is to offer multiple views to different aspects of the content, both to visualize it and to select a subset from it by specifying constraints in the views [7].

We have implemented a general multi-facet search engine that plugged into an instance database uses the other created visualization and selection components as views. This also represents one of the more complex interaction patterns between components, as first the views provide constraint selection for the search engine, which then calculates a result set based on the instance database, and feeds this result set back to the views for visualization. As an example, the selection tree in figure 1(a) is actually part of such a configuration, with the numbers representing how many items annotated with that concept are in the current multi-facet search result set. An earlier version of the component [17] has already been used in the portals MuseumFinland[7] [2], Orava[8] [18], Veturi[9] [19] and SW-Suomi.fi[10] [20]

Concept Location, Disambiguation and Extraction from Text It is often useful to be able to locate concepts in textual resources. In annotating documents, suggestions for annotations can be found from the text [21, 22]. In web

[7] http://www.museosuomi.fi/
[8] http://demo.seco.tkk.fi/orava/
[9] http://demo.seco.tkk.fi/veturi/
[10] http://demo.seco.tkk.fi/suomifi/

browsing, on the other hand, semantic content can be linked to topics being discussed in the text [23].

We have implemented a component [8] that can locate concepts and instances in text documents based on the labels associated with them. The extraction process starts with document preprocessing, which in the case of HTML documents means extracting the textual content from the document. Then, the text is tokenized and lemmatized if needed. Next, the extraction component iterates over the tokenized document and finds strings corresponding to the concept labels. In cases where labels are ambiguous (e.g. "bank"), the component can make use of the ontological neighbourhoods of the concepts by counting nearby occurrences of neighbour concept labels of each candidate in order to guess the proper meaning. After finding the matches, the component then tags the occurences of the concepts and instances in the document and outputs this tagged copy.

2.2 Semantic Linking

Semantic metadata combined with logical rules makes it possible to automatically link related content together to support semantic browsing (semantic recommendations) [24, 2]. Automatic linking is an especially important feature in Semantic Web based systems where the content is typically aggregated from different sources. Here, manual linking is difficult because the content providers typically consider only the local view on their content excluding the global view on the aggregated content [25].

For automatic link generation, we have created functionalities based on three different techniques. First, used in the already mentioned portals MuseumFinland, Orava and SW-Suomi.fi is the rule-based linking server Ontodella [24], which is accessed by a simple HTTP request containing the URI of the current document. The system then responds, based on the item metadata and ontologies linked to it, with a set of related documents. Each recommendation also contains a human-readable explanation of the relation between the current and the recommended document. For example, when looking at a nautical flag in MuseumFinland, the object is linked to sailor's clothes because, based on the metadata and the ontologies, they are used in the same situation, seafaring.

Our second link generation component is based on calculating a similarity measure between items in the linked instance database using an event-based schema, which allows one to compare items annotated using dissimilar annotation schemas [26]. This is the component used in CultureSampo, the case application described later in this paper.

Finally, particularly for inter-portal semantic linking, we have exposed the multi-facet functionalities of our portals to mash-up use. For utilizing them, we propose the concept of *floatlets*, semantic linking widgets that can be easily plugged into any web page. Based on metadata and shared ontologies, the floatlet is able to make semantically relevant queries to the portals and show them in the context of the page. For example, figure 3 shows how the Finnish

Broadcasting Company's video archive[11] has been semantically linked based on metadata with relevant content in MuseumFinland. In the example, the current video is about the history of speed skating, which has also been described in its metadata. Based on this information, the floatlet is able to query for old skates from MuseumFinland. By clicking on the floatlet links, the skates can be examined in more detail in their original portal. From the semantic portal publisher's point of view, floatlets provide a new way for publishing and promoting their content on the web. By clicking on the floatlet links, new visitors move over to the floatlet's host portal.

The idea of floatlets is similar to Google AdSense[12] which is used for adding advertisements to web pages. However, in floatlets the returned links are based on explicit ontological annotations. This allows the web developer to specify in detail what information is to be linked, either manually or based on the metadata of the current web page.

2.3 Concept and Instance Views

To be able to comprehend the meanings of concepts and their relations, content visualization techniques are needed. The simplest way to approach this is to show the properties of a concept or an instance to the user, e.g. simply as a list of properties and their respective values. If a value is a resource, it can also act as a link to allow browsing the content.

Ontologies are typically organized into hierarchical structures based on subsumption, partition or other properties. This structural context of the concept gives important information about its meaning and relations to other concepts. These functions can be fulfilled by such context visualizations as already described with reference to semantic autocompletion and depicted in figure 1.

2.4 Shared Concept and Instance Storage and Maintenance

In many semantic applications, there is a need for storing ontological metadata, be they the annotations of an indexer or links forged between ontologies by an ontology maintainer. However, existing systems may lack the means by which to store arbitrary semantic constructs or even just URIs. For example, a museum indexing system may contain databases for museum items and actors, but provide only a text field for storing locations. On the Semantic Web however, locations need to be stored as instances with properties of their own.

To address this need, we have developed a browser-based metadata editor SAHA, which can be used either as is or as depicted in figure 4, as a web widget in existing indexing systems lacking semantic capabilities [27]. Connecting SAHA to indexing systems can be done simply by linking to SAHA with a GET parameter specifying the identifier (URI) of the document being currently edited. By clicking on the link, a SAHA window opens, containing indexing

[11] http://www.yle.fi/elavaarkisto/
[12] http://www.google.com/adsense/

fields relevant for the current document. Afterwards, the annotations located in the indexing system and SAHA can be combined by their common document identifiers.

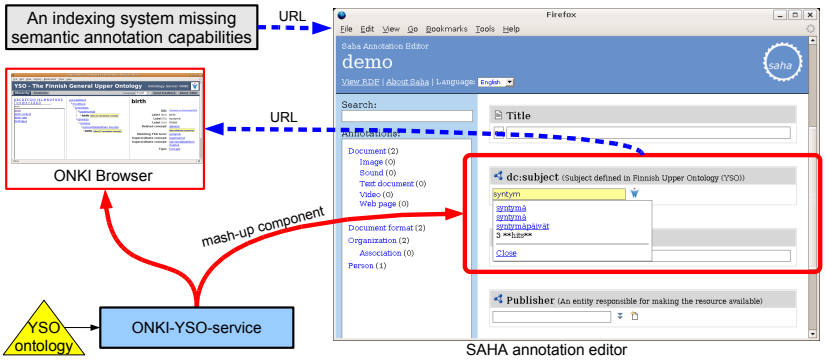


Fig. 4. The Finnish General Ontology connected to SAHA.

3 Case Applications

In the following, we present how we have combined and applied the previously described semantic web widgets in actual systems, both in content creation as well as end-user consumption.

3.1 The ONKI Server: Publishing Ontologies as Web Widgets

Currently, ontologies are typically shared by downloading them, and each application must separately implement the functionatily to support them. To avoid duplicated work and costs, and to ensure that the ontologies are always up-to-date, we argue that the ontologies should also be published as shared services. To demonstrate this aproach, we have implemented the ontology library service ONKI, which is used for maintaining, publishing and using ontologies [6].

As part of the ONKI concept, a user-interface web widget combining a semantic autocompletion search sub-component with concept fetching functionality has been implemented, which can be added to any application requiring access to a certain ontology. The widget looks like an ordinary text field, but when the user types in characters, matching concepts found from the ONKI server are listed. By selecting a concept from the result list, the concept's URI, label or other information is fetched to the client application for further processing and storage. In the context of a browser-based application, this fetching functionality has been implemented with JavaScript window referencing [28].

Another ONKI feature is the concept browser, which can be integrated to an application as an "ONKI button". When the button is pressed, a separate ONKI Browser window (see figure 5) opens, in which annotation concepts can be searched for and browsed, making use of semantic autocompletion, tree context visualization and concept property view components. For each concept entry, the browser shows a *Fetch concept* button which, when pressed, transfers the current concept information to the client application. For geographical data, a separate browser application, ONKI-Paikka[13] [29] has been created. This browser, shown in figure 6, has been implemented by combining ontological information with our geographical search and visualization widget.

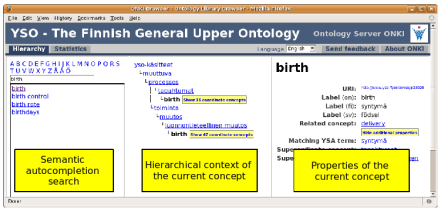


Fig. 5. The ONKI Ontology Browser's concept view.

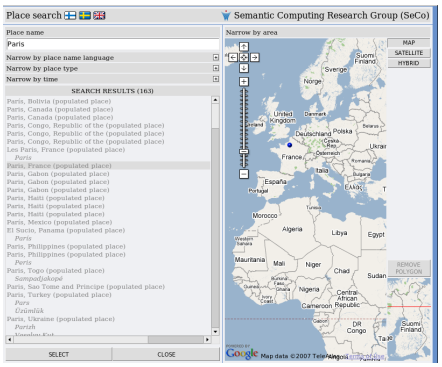


Fig. 6. The ONKI-Paikka browser.

Integrating these ONKI services to client applications only requires a minimal modification to the user interface implementation. For example, in the case of HTML pages and AJAX, only a short snippet of JavaScript code must be added to the web page for using the ONKI functionalities.

To test the ONKI solution, we have used the widgets in the browser-based annotation editor SAHA[14] [8, 27]. For example (figure 4), the Finnish General Ontology YSO [6] has been published as an ONKI service[15], and has been added as a web widget to SAHA for selecting annotation concepts. SAHA can also make use of our automatic text extraction component in extracting potential annotations from web resources.

In the case of the ONKI browsers, all concept and instance URIs are intended to be designed so that they function also as URLs. When the URI of a concept is accessed with a web browser, the relevant view is opened in the ONKI browser. This means that the URI itself acts as a functional link when added to a HTML

[13] http://www.seco.tkk.fi/services/onkipaikka/
[14] http://www.seco.tkk.fi/services/saha/
[15] http://www.yso.fi/onto/yso/

page. In accordance to W3C[16], if the URI is accessed with an RDF aware system, the machine readable RDF presentation of the content is returned instead of the ONKI browser's HTML presentation. This makes it easier to use ONKI services also in programmatic mash-up applications.

Compared to general ontology server interfaces such as the SKOS API[17], our approach is to publish highly specified functionalities as semantic web widgets that solve a specific user task, such as the need for concept search and fetch. In this, our approach complements the general APIs. The general APIs make it possible to create completely new functionality but require more programming work, while semantic web widgets make handling the most common tasks as cost-effective as possible.

3.2 CultureSampo: A Semantic Portal for Cultural Content

CultureSampo [30] is a semantic portal that gathers together a comprehensive collection of Finnish culture, including photographs, paintings, poems and biographies. Much of the functionality of the portal has been accomplished by combining the various components described above, as depicted in figure 7

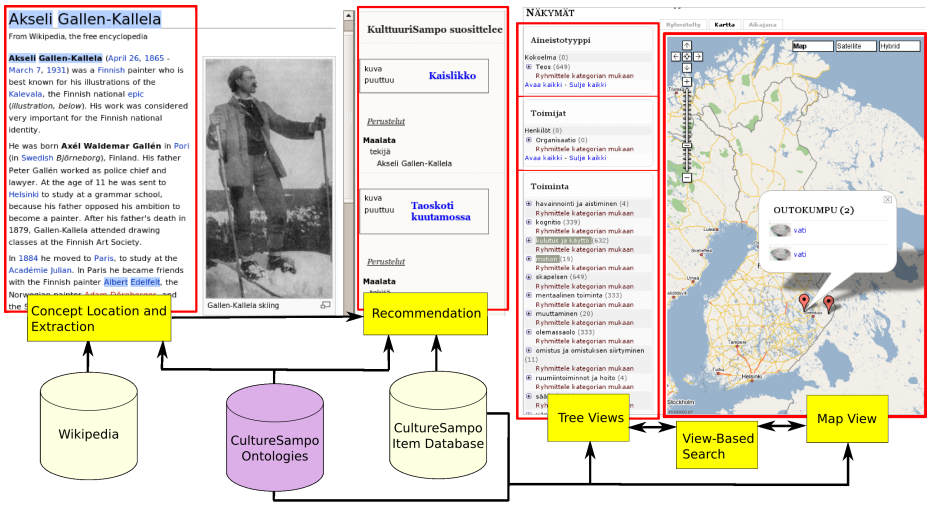


Fig. 7. The mash-up architecture of the CultureSampo portal

First, we have harnessed our automatic concept extraction component to enchance external web pages with CultureSampo content when viewed through the portal. For example, on the left in figure 7, a web page from Wikipedia[18] is

[16] http://www.w3.org/TR/swbp-vocab-pub/
[17] http://www.w3.org/2001/sw/Europe/reports/thes/skosapi.html
[18] http://www.wikipedia.org/

integrated into the portal. The person names highlighted in blue on the page are detected individuals, and their names are links to their biography in the portal. On the right of the page, other recommended items based on the content of the document are shown. These are calculated by feeding the concepts extracted from the page to our recommendation component. CultureSampo also provides multi-facet search functionality that utilizes our engine component combined with both tree and map-based search and visualization views. This functionality, along with a screenshot, is depicted on the right side in figure 7.

4 Discussion

4.1 Contributions

This paper presented the idea of using the mash-up approach for implementing semantic functionalities as web widgets which can easily be included in applications, such as adding concept search functionality to an indexing application.

A major benefit of the approach is that potentially highly complicated and expensive technical features and semantic data resources can be created once and published for others to use in a compact package, which can easily be integrated to an application. By making the adoption of semantic technologies as easy as possible, one may hope to further the adoption of the Semantic Web as a whole.

One of the benefits of publishing the widgets as centralized services is that updates in content and functionalities are instantenously available for the users. This is an especially important feature when the data evolves constantly, e.g. when user generated content is involved.

4.2 Related Work

Our own prior work on a semantic portal creation tool [31] already included a general semantic view-based search tool [17] and the semantic linking service Ontodella [24], as well as a framework for combining them into a complete portal. However, the user interface components were not yet modular, and neither were the search or recommendation functions used outside that environment.

On the other hand, many semantic web browsing and editing environments do provide general configurable visualization and selection widgets inside them, such as DBin [32], Piggy Bank [33], OntoWiki [34] and Haystack [35, 36]. These, however, are intended for use only inside the specific program environment, while our components are for general use.

Complementing our pursuits, there have recently been many announcements about mash-ups that make currently existing data available in RDF. DBpedia.org [37] has published Wikipedia material, the D2R server [38] has been used for publishing the DBLP article database[19], while the RDF Book Mashup[20] provides book information from Amazon and Google Base. From our viewpoint,

[19] http://www4.wiwiss.fu-berlin.de/dblp/
[20] http://sites.wiwiss.fu-berlin.de/suhl/bizer/bookmashup/

these mash-ups provide possible data sets to which our components could be hooked. They provide the data, we provide the functionality.

4.3 Future Work

Future directions for this research include looking for new general semantic web tasks that could be implemented using the web widget approach, especially in ontology development and maintenance. For example, supporting cross-link maintenance between ontologies, ontology change history maintenance and using ontology history knowledge in searches seem to be potential further research directions. The proposed semantic web widgets could also be developed further. For example, their functionalities could be provided via additional technologies alongside the current JavaScript and SOAP Web Service APIs.

Acknowledgements

This work is part of the National Semantic Web Ontology (FinnONTO) project[21], funded mainly by the National Funding Agency for Technology Innovation (Tekes). We wish to thank Ville Komulainen for his work on the ONKI Browser.

References

1. Berners-Lee, T., Hendler, J., Lassila, O.: The semantic web. Scientific American **284**(5) (May 2001) 34–43
2. Hyvönen, E., Mäkelä, E., Salminen, M., Valo, A., Viljanen, K., Saarela, S., Junnila, M., Kettula, S.: Museumfinland – finnish museums on the semantic web. Journal of Web Semantics **3**(2) (2005) 25
3. O'Reilly, T.: What is web 2.0 — design patterns and business models for the next generation of software. WWW article (September 30 2005) http://www.oreillynet.com/pub/a/oreilly/tim/news/2005/09/30/what-is-web-20.html.
4. Zhang, S.L.: Planning an authority control project at a medium-sized university library. College and Research Libraries **62**(5) (2001)
5. Niles, I., Pease, A.: Towards a standard upper ontology. In Welty, C., Smith, B., eds.: Proceedings of the 2nd International Conference on Formal Ontology in Information Systems (FOIS-2001). (October 17-19 2001)
6. Hyvönen, E., Viljanen, K., Mäkelä, E., Kauppinen, T., Ruotsalo, T., Valkeapää, O., Seppälä, K., Suominen, O., Alm, O., Lindroos, R., Känsälä, T., Henriksson, R., Frosterus, M., Tuominen, J., Sinkkilä, R., Kurki, J.: Elements of a national semantic web infrastructure - case study finland on the semantic web. (May 18 2007) Submitted for review.
7. Mäkelä, E.: View-based search interfaces for the semantic web. Master's thesis, University of Helsinki (June 2006)

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

8. Valkeapää, O., Alm, O., Hyvönen, E.: Efficient content creation on the semantic web using metadata schemas with domain ontology services (system description). In: Proceedings of the European Semantic Web Conference ESWC 2007, Innsbruck, Austria, Springer (June 4-5 2007)
9. Hyvönen, E., Mäkelä, E.: Semantic autocompletion. In: Proceedings of the first Asia Semantic Web Conference (ASWC 2006), Beijing, Springer-Verlag, New York (August 4-9 2006)
10. Kauppinen, T., Henriksson, R., Väätäinen, J., Deichstetter, C., Hyvönen, E.: Ontology-based modeling and visualization of cultural spatio-temporal knowledge. In: Developments in Artificial Intelligence and the Semantic Web - Proceedings of the 12th Finnish AI Conference STeP 2006. (October 26-27 2006)
11. Kauppinen, T., Deichstetter, C., Hyvönen, E.: Temp-o-map: Ontology-based search and visualization of spatio-temporal maps. Demo track at the European Semantic Web Conference ESWC 2007, Innsbruck, Austria (June 4-5 2007)
12. Ferraiolo, J., Jackson, D., Fujisawa, J.: Scalable Vector Graphics (SVG) 1.1 specification W3C recommendation. Technical report, World Wide Web Consortium W3C (January 14 2003)
13. Kauppinen, T., Hyvönen, E.: Modeling and reasoning about changes in ontology time series. In: Ontologies: A Handbook of Principles, Concepts and Applications in Information Systems (January 15 2007)
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. (November 5-9 2006) 272–285
15. Oren, E., Delbru, R., Decker, S.: Extending faceted navigation for rdf data. In: International Semantic Web Conference. (November 5-9 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. (Nov 2005)
17. Mäkelä, E., Hyvönen, E., Saarela, S.: Ontogator — a semantic view-based search engine service for web applications. In: Proceedings of the 5th International Semantic Web Conference (ISWC 2006). (Nov 2006)
18. Känsälä, T., Hyvönen, E.: A semantic view-based portal utilizing Learning Object Metadata (August 2006) 1st Asian Semantic Web Conference (ASWC2006), Semantic Web Applications and Tools Workshop.
19. Mäkelä, E., Viljanen, K., Lindgren, P., Laukkanen, M., Hyvönen, E.: Semantic yellow page service discovery: The veturi portal. In: Poster paper, 4th International Semantic Web Conference. (Nov 2005)
20. Sidoroff, T., Hyvönen, E.: Semantic e-goverment portals - a case study. In: Proceedings of the ISWC-2005 Workshop Semantic Web Case Studies and Best Practices for eBusiness SWCASE05. (Nov 2005)
21. Dill, S., Eiron, N., Gibson, D., Gruhl, D., Guha, R.: Semtag and seeker: Bootstrapping the semantic web via automated semantic annotation. In: In Proceedings of the 12th International World Wide Web Conference, ACM Press (2003) 178–186
22. Kiryakov, A., Popov, B., Terziev, I., Manov, D., Ognyanoff, D.: Semantic annotation, indexing, and retrieval. Journal of Web Semantics **2**(1) (2004) 49–79
23. Dzbor, M., Domingue, J., Motta, E.: Magpie: towards a Semantic Web browser. Proceedings of the 2nd International Semantic Web Conference (2003)
24. Viljanen, K., Känsälä, T., Hyvönen, E., Mäkelä, E.: Ontodella - a projection and linking service for semantic web applications. In: Proceedings of the 17th International Conference on Database and Expert Systems Applications (DEXA 2006), Krakow, Poland, IEEE (September 4-8 2006) 370–376

25. Calí, A., Giacomo, G.D., Lenzerini, M.: Models for information integration: Turning local-as-view into global-as-view. In: Proc. of Int. Workshop on Foundations of Models for Information Integration (10th Workshop in the series Foundations of Models and Languages for Data and Objects). (2001)
26. Ruotsalo, T., Hyvönen, E.: A method for determining ontology-based semantic relevance. In: Proceedings of the 18th International Conference on Database and Expert Systems Applications. (September 3-7 2007)
27. Valkeapää, O., Hyvönen, E.: A browser-based tool for collaborative distributed annotation for the semantic web. (September 26 2006) 5th International Semantic Web Conference, Semantic Authoring and Annotation Workshop, November, 2006.
28. Komulainen, V.: Public services for ontology library systems. Master's thesis, University of Helsinki, Department of Computer Science (January 2007)
29. Lindroos, R., Kauppinen, T., Henriksson, R., Hyvönen, E.: Onki-paikka: An ontology service for geographical data (2007) Submitted for review.
30. Hyvönen, E., Ruotsalo, T., Häggström, T., Salminen, M., Junnila, M., Virkkilä, M., Haaramo, M., Mäkelä, E., Kauppinen, T., , Viljanen, K.: Culturesampo–finnish culture on the semantic web: The vision and first results. In: Developments in Artificial Intelligence and the Semantic Web - Proceedings of the 12th Finnish AI Conference STeP 2006. (October 26-27 2006)
31. Mäkelä, E., Hyvönen, E., Saarela, S., Viljanen, K.: Ontoviews – a tool for creating semantic web portals. In: Proceedings of the 3rd International Semantic Web Conference (ISWC 2004). (May 2004)
32. Tummarello, G., Morbidoni, C., Nucci, M., Panzarino, O.: Brainlets: "instant" semantic web applications. In Bizer, C., Auer, S., Miller, L., eds.: Proc. of 2nd Workshop on Scripting for the Semantic Web at ESWC, Budva, Montenegro. Volume 183 of CEUR Workshop Proceedings ISSN 1613-0073. (June 2006)
33. Huynh, D., Mazzocchi, S., Karger, D.: Piggy bank: Experience the semantic web inside your web browser. J. Web Sem. **5**(1) (2007) 16–27
34. Auer, S., Dietzold, S., Riechert, T.: OntoWiki - a tool for social, semantic collaboration. In Cruz, I.F., Decker, S., Allemang, D., Preist, C., Schwabe, D., Mika, P., Uschold, M., Aroyo, L., eds.: International Semantic Web Conference. Volume 4273 of Lecture Notes in Computer Science., Springer (2006) 736–749
35. Karger, D.R., Bakshi, K., Huynh, D., Quan, D., Sinha, V.: Haystack: A general-purpose information management tool for end users based on semistructured data. In: Proceedings of the CIDR Conference. (2005) 13–26
36. Quan, D., Huynh, D., Karger, D.R.: Haystack: A platform for authoring end user semantic web applications. In: Proceedings of the Second International Semantic Web Conference. (2003) 738–753
37. Auer, S., Lehmann, J.: What have innsbruck and leipzig in common? extracting semantics from wiki content. In: Proceedings of the European Semantic Web Conference ESWC 2007, Innsbruck, Austria, Springer (June 4-5 2007)
38. Bizer, C., Cyganiak, R.: D2R server — publishing relational databases on the semantic web. Poster paper at the 5th International Semantic Web Conference (November 2006)