Combining Context Navigation with Semantic Autocompletion to Solve Problems in Concept Selection

Reetta Sinkkilä, Eetu Mäkelä, Tomi Kauppinen, 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. Many tasks on the semantic web require the user to choose concepts from a limited vocabulary e.g. for describing an indexed resource or for use in semantic search. Semantic autocompletion interfaces offer an efficient way for concept selection. However, these interfaces usually do not expose the semantic context of the matched concepts, thereby making it hard to know if a matched concept is the right one, as well as hiding possibly more appropriate choices. Ontology browsers, on the other hand, show context but do not allow quick discovery or embedding into other applications. To lessen these problems, we present an interface combining semantic autocompletion with in-place ontological context navigation. Because required context differs between ontologies, the implementation was designed to make it easy to add different contexts and visualizations. To test the applicability of our idea and implementation the system was tested on three ontologies with different requirements and structure.

1 Introduction

Selecting concepts from a limited vocabulary is a frequent task in semantic web applications. For example, in semantic indexing [1], concepts describing the item to be indexed must be located from domain ontologies. Many semantic search systems [2] also require the user to create search patterns by picking up concepts and relations to be matched against the instance database.

Traditionally, the interfaces for concept selection have fallen into two camps, popular at different times. Many early applications used class-tree -based selectors [3], but recently these seem to have been eclipsed by semantic autocompletion [4–6] interfaces. However, moving to semantic autocompletion some beneficial functionality from class-tree -based browsers has been lost. In this paper we analyze problems related to both types of concept selectors and introduce our own interface idea which combines the two paradigms to resolve these deficiencies.

** This research is part of the National Finnish Ontology Project (FinnONTO) 2003-2007, 2008-2010 funded by the National Technology Agency (Tekes) and a consortium of 38 companies and public organizations.

2 Problems with Current Concept Selectors

2.1 Autocompletion Interfaces

In autocompletion interfaces, the idea is that the user can expediently find concepts to be used in annotation or further semantic search by typing in parts of their labels, with the interface speedily returning matching concepts for selection as they type. Now, this basic approach works very well when the user knows the vocabulary, but has drawbacks when they do not. For example, in indexing, it is quite easy for the annotator to find suitable indexing concepts by trying different query strings, but it is hard to be sure that the concepts they have found are the best for that particular item.

For example, consider a scenario where a user is annotating with Iconclass [7] a picture of a serpent or a dragon swallowing its own tail. In reality, this is the serpent Ouroboros, a historical symbol of time and cyclicity also found directly in the Iconclass vocabulary. Yet, with an autocompletion search a user not recalling the name Ouroboros might enter query strings like “snake” or “monster” and be satisfied with these annotation terms.

An additional drawback with simple autocompletion occurs in situations where the semantic meaning of a concept cannot be ascertained based on its name or direct properties only. In Iconclass, for example, the meaning consists of the whole conceptual hierarchy above it. Returning to our example with the serpent Ouroboros, the broader concepts of “monster” in Iconclass include terms such as “human being”, “disabilities” and “deformations”. Thus, it is clear that “monster” is not a suitable term to describe the picture.

Contextual information may also be essential to distinguish ambiguous concept names. For example in the Finnish geo-ontology SUO [8] there are 491 places in Finland sharing the same name “Isosaari” (great island) that are instances of several geographical classes, such as Island, Forest, Peninsula, Inhabited area, etc. Referencing unambiguously to a particular ”Isosaari”, either when indexing content or during information retrieval, can be quite problematic and requires usage of context information and maps for semantic disambiguation.

Autocompletion interfaces also overlook the potential associated with related terms and other properties of a concept. To continue with the example scenario described before, the annotation process would be considerably facilitated if it could make use of the fact that in Iconclass the term “Serpent Ouroboros” is annotated as a related term to “snakes”.

Also, in Iconclass there exists a separate set of keywords which are *not* allowed indexing classes, but instead relate to and act as pointers to classes in different parts of the classification. Thus, the Iconclass keyword “human being” relates to twenty classes with aspects of “man in community”, “man and animals” and “religious viewpoints of man”. However, with a simple semantic autocompletion interface, “human being” could not be shown or made use of, as it is not in the range of selectable classes.

2.2 Ontology browsers

Another approach for a concept selection component is a class-tree browser, seen in various ontology editors [9–11] and also particularly in earlier search systems [3]. In contrast to an autocompletion selector, a tree-class browser overcomes the drawbacks related to familiarizing with the ontology. Also, the place of the viewed concepts in the concept hierarchy is revealed, which might help in gauging the actual meaning of a concept. Usually also other properties of the concept are shown, leading to the possibility of discovering more suitable concepts through browsing them.

A definite disadvantage of this approach is the slowness and inconvenience of use. A user may have to click through numerous branches of the class tree, particularly when unfamiliar with the vocabulary used. Additional confusion may occur in situations when the ontology in use contains heavy multiple inheritance, such as in the medical ontology MeSH[1] or when nearly synonymous concepts are located in vastly different parts of the ontology, such as in Iconclass.

Based on this analysis, it is clear that neither of the discussed concept selector types can in itself satisfy all the requirements of semantic concept selection. Instead, in order to keep the advantages of speed and low screen space usage of semantic autocompletion interfaces, but lessen the previously mentioned drawbacks, we have devised an interface combining an in-place ontological context navigation interface with semantic autocompletion.

3 The In-place Ontological Context Navigation Interface

Our implemented in-place context navigation interface[2] is depicted in figure 1. On the left is the autocompletion component, where the user has typed in “tork”, grasping for “torka”, the Swedish word for drought. Yet, of the ontologies in the system, only the general Finnish upper ontology YSO[3] has Swedish labels and thus, with a normal autocompletion interface for Iconclass, the user would be left stymied. But here, concepts matched in inapplicable domains merely act as pointers to the right direction. Mousing over any concept in the autocompletion results brings up the semantic context of that concept, by which the user can be assured of the meaning of that concept, as well as look for more appropriate choices. In addition, this functionality is repeated for each of these revealed contexts, thus enabling limited navigation of the ontology based on related concepts, a class tree, etc.

In this example, the user has first browsed through the YSO concept to the Iconclass keyword “kuivuus” (Finnish for drought), and finally through there to the Iconclass concepts permissible for annotation. And even here, the user has not selected the Iconclass concept for drought, but has elected “wadi”, or dry river bed, as more appropriate from the choices that the context visualization

[1] http://www.nlm.nih.gov/mesh/MBrowser.html
[2] Available at http://demo.seco.tkk.fi/irma/app/irma_iconclass.html
[3] http://www.yso.fi/onto/yso

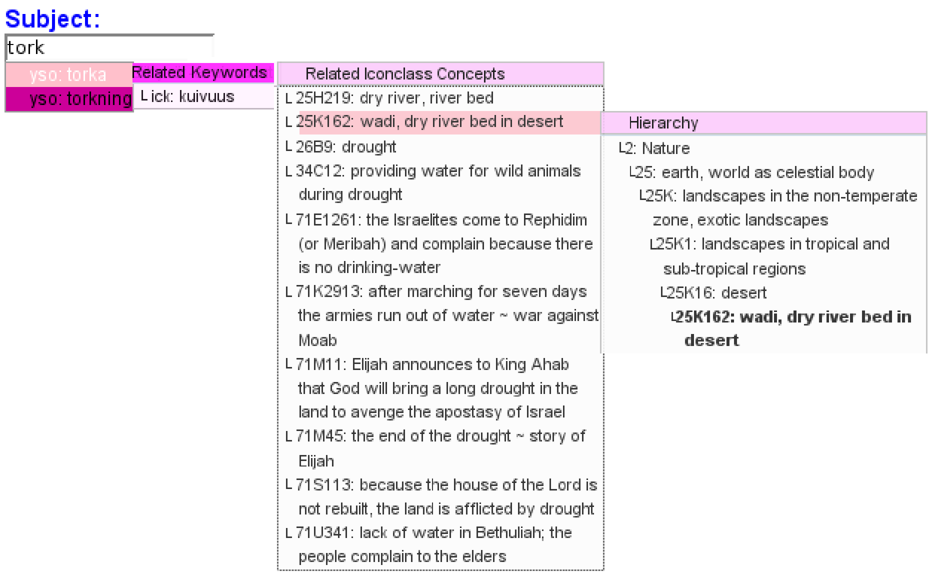


Fig. 1. Semantic autocompletion with in-place ontological context navigation

made available and evaluatable. Also note that the choice does not even contain the original query keyword in any language.

4 Design for modularity

So far the only contextual information we have focused on is the concept hierarchy and related concepts. However, what is useful context and how it should be visualized depends heavily on the particular ontology. For example, in a location ontology, relevant properties might include neighboring locations as well as information about consolidation of municipalities. Geographical objects would surely also benefit from being visualized on a map [12]. For an ontology of historic events [13], on the other hand, a timeline visualization would be relevant [14].

Due to this vast diversity of information structures and possible visualization techniques we found it crucial to develop a browser that would be easily adapted to satisfy different requirements. Thus, to be able to visualize random ontologies in a reasonable manner, the browser needs to be modular, easily extendable and configurable.

Our system is based on an extensible platform in which modular, configurable context providers are registered in a central Java service, each producing a bean object containing all necessary information for that context. These information objects are then pushed as objects through AJAX connections to a browser user interface, where JavaScript modules corresponding to the server versions lay out the information inside context boxes as desired. Due to efficiency concerns, our system uses precalculated indices for storing the information required for the context boxes, with reindexing done when the underlying RDF files are updated.

Thus, new contexts for new ontologies can be added with relatively little programming work, either only altering the JavaScript display code, or also creating a new index provider in the Java layer.

5 Testing the applied system

To test the applicability and mutability of our interface idea as well as our implementation, we selected three vastly different datasets. The first is a combination of Iconclass, Iconclass keywords and the Finnish Upper Ontology YSO. This dataset demonstrates how multi-ontological data can be utilized in our system, and what kinds of visualizations can be used in such an environment. The second environment deals with a medical ontology with lots of multiple inheritance, while the last test is on a spatio-temporal location ontology spanning both time and space.

5.1 Iconclass

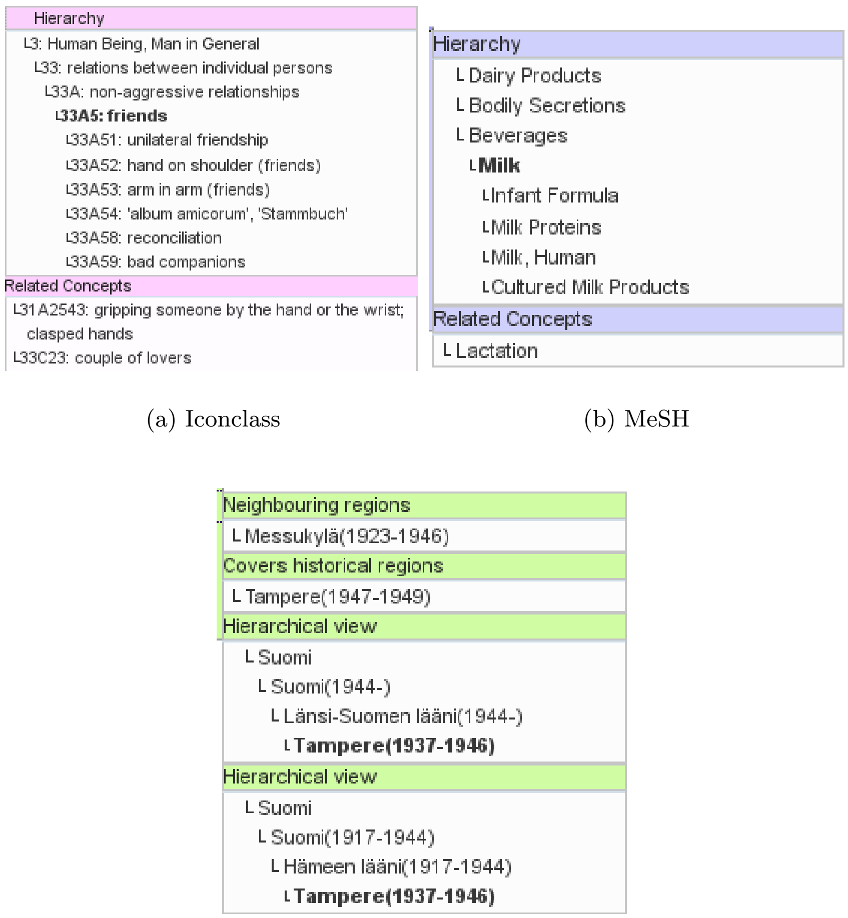
The first test of the system was with the Iconclass ontology along with its keywords, combined with the Finnish Upper Ontology YSO. Here, the application is intended as an annotation interface to circumstances where the users are not familiar with the annotation ontology Iconclass, but are acquainted with YSO. Here, the interface is eminently suitable because it provides a search interface for familiar YSO concepts, but offers linkage from them to the Iconclass ones. Also, considering the users' unfamiliarity with Iconclass, presenting the context hierarchy is advantageous for the process of familiarizing oneself with Iconclass especially because YSO and Iconclass are structurally very different.

Because the Iconclass version of our system contains three ontologies, we decided to color terms coming from different ontologies distinctly. From YSO concepts one may navigate to Iconclass keywords and from there onwards to Iconclass classes as illustrated in figure 1.

5.2 Medical Subject Headings MeSH

Another instance of our system was implemented with an ontological version of MeSH, The Medical Subject Headings[4], containing heavy multiple inheritance. Here, showing all the superclass trees as we had done with Iconclass (showed in figure 2(a)) was not feasible nor even desired, as the semantic meaning of the medical terms in MeSH does not consist of the whole conceptual hierarchy above. Rather, all interesting information are basically the classes directly below or above the selected concept, so we decided to show only those, as depicted in figure 2(b). Here the semantic context of an ontological concept "milk" shows

[4] http://www.nlm.nih.gov/mesh/MBrowser.html



(a) Iconclass (b) MeSH

(c) SAPO

Fig. 2. Context boxes for various ontologies

that "milk" is a "dairy product" a "bodily secretion" as well as a "beverage", and has the subclasses "milk proteins" and "infant formula".

Due to the modular structure of our implementation, the indexing was done with basically no changes to program structure. All that needed to be changed was the JavaScript user interface layer.

5.3 Time-Location Ontology Sapo

We tested the applied system also with the Finnish Time-Location Ontology SAPO [15, 12]. Currently SAPO consists of 667 different regions in time, that is, Finnish counties that have existed during a period from the beginning of the 20th century until today. In addition, the ontology contains information on neighboring regions and coverages (overlaps) between historical regions such as counties. A distinct feature of SAPO is that it has a hierarchy of relatively flattened form, yet there is much multiple inheritance.

Due to the character of time-location information we decided to depict the respective spatio-temporal containment hierarchies in separate context boxes (see picture 2(c)). Shown is the regional hierarchy of the Finnish city of Tampere, as it existed in 1937-1946. In the years 1917-1944 Tampere was part of the county of "Hämeen lääni", but in 1944 the county division changed, and Tampere was designated a part of "Länsi-Suomen lääni". The context visualization also shows the neighboring regions of Tampere at the designated time, showing the municipality of Messukylä, which is nowadays part of Tampere.

6 Discussion

The concept selection is an essential task in semantic web applications. At present concept selection is mainly conducted with an autocompletion interface or with a concept browser. Yet both these approaches have drawbacks, related to either speed, disambiguation of similar concepts, familiarizing the user with the vocabulary or generally in situations where the meaning of a concept cannot be ascertained based on its name or direct properties alone.

To solve these problems, an interface combining cross-ontology navigation with semantic autocompletion was designed and implemented, and tested in three diverse scenarios. Regardless of the widely differing requirements of these tasks, and because of the modularity of the implementation and the generalness of the paradigm, we found it effortless to modify the system for each environment.

As related work, showing context information in an ontological autocompletion interface has also been studied by Hildebrand et al [6]. Their approach includes some context navigation by grouping autocompleted results and suggesting hyponym terms. However, it is not configurable in the dimensions with regard to new contexts, and can not contain arbitrary amounts of context information.

Future work with our interface and the SAPO locational ontology includes integrating our component into the the CultureSampo culture heritage portal [16]. Here, the component is intended to be combined with a map visualization service, to enable a user to select spatiotemporal geographical instances and visualize them on a map, along with cultural objects somehow related to that place and time.

References

1. 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 (2007)

2. Mäkelä, E.: Survey of semantic search research. In: Proceedings of the Seminar on Knowledge Management on the Semantic Web, Department of Computer Science, University of Helsinki (2005)
3. Airio, E., Järvelin, K., Saatsi, P., Kekäläinen, J., Suomela, S.: CIRI - an ontology-based query interface for text retrieval. In Hyvönen, E., Kauppinen, T., Salminen, M., Viljanen, K., Ala-Siuru, P., eds.: Web Intelligence: Proceedings of the 11th Finnish Artificial Intelligence Conference. (2004)
4. Hyvönen, E., Mäkelä, E.: Semantic autocompletion. In: Proc. of the First Asia Semantic Web Conference (ASWC 2006), Beijing, Springer-Verlag (2006)
5. 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 Conference. (2006) 272–285
6. Hildebrand, M., van Ossenbruggen, J., Amin, A., Aroyo, L., Wielemaker, J., Hardman, L.: The design space of a configurable autocompletion component. Technical Report INS-E0708, CWI (2007)
7. Gordon, C.: An introduction to iconclass. In: Terminology for Museums. Proceedings of an International Conference. (1988) 233–244
8. Henriksson, R., Kauppinen, T., Hyvönen, E.: Core geographical concepts: Case finnish geo-ontology. In: First International Workshop on Location and the Web (LocWeb 2008), ACM Digital Library, 17th International World Wide Web Conference WWW 2008, Beijing, China. (2008)
9. Noy, N.F., Sintek, M., Decker, S., Crubezy, M., Fergerson, R.W., Musen, M.A.: Creating semantic web contents with protege-2000. IEEE Intelligent Systems **2**(16) (2001) 60–71
10. Sure, Y., Erdmann, M., Angele, J., Staab, S., Studer, R., Wenke, D.: Ontoedit: Collaborative ontology engineering for the semantic web. In Horrocks, I., Hendler, J., eds.: Proceedings of the First International Semantic Web Conference 2002 (ISWC 2002), June 9-12 2002, Sardinia, Italia. Volume 2342 of LNCS., Springer (2002) 221–235
11. Kalyanpur, A., Parsia, B., Sirin, E., Cuenca-Grau, B., Hendler, J.: Swoop: A web ontology editing browser. Journal of Web Semantics **4** (2006) 2
12. Kauppinen, T., Väätäinen, J., Hyvönen, E.: Creating and using geospatial ontology time series in a semantic cultural heritage portal. In: Proceedings of the 5th European Semantic Web Conference 2008, Tenerife, Spain. (2008) Accepted, forthcoming.
13. Hyvönen, E., Alm, O., Kuittinen, H.: Using an ontology of historical events in semantic portals for cultural heritage. In: Proceedings of the Cultural Heritage on the Semantic Web Workshop at the 6th International Semantic Web Conference (ISWC 2007). (2007)
14. Ringel, M., Cutrell, E., Dumais, S.T., Horvitz, E.: Milestones in time: The value of landmarks in retrieving information from personal stores. In Rauterberg, M., Menozzi, M., Wesson, J., eds.: INTERACT, IOS Press (2003)
15. Kauppinen, T., Hyvönen, E.: Modeling and reasoning about changes in ontology time series. In Kishore, R., Ramesh, R., Sharman, R., eds.: Ontologies: A Handbook of Principles, Concepts and Applications in Information Systems. Integrated Series in Information Systems, New York, NY, Springer-Verlag, New York (NY) (2007) 319–338
16. 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: In: K. Robering (ed.): Information Technology for the Virtual Museum. LIT Verlag, Berlin. (2007)