ONKI SKOS Server for Publishing and Utilizing SKOS Vocabularies and Ontologies as Services

Jouni Tuominen, Matias Frosterus, Kim Viljanen, and Eero Hyvönen

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

Abstract. Vocabularies are the building blocks of the Semantic Web providing shared terminological resources for content indexing, information retrieval, data exchange, and content integration. Most semantic web applications in practical use are based on lightweight ontologies and, more recently, on the Simple Knowledge Organization System (SKOS) data model being standardized by W3C. Easy and cost-efficient publication, integration, and utilization methods of vocabulary services are therefore highly important for the proliferation of the Semantic Web. This paper presents the ONKI SKOS Server for these tasks. Using ONKI SKOS, a SKOS vocabulary or a lightweight ontology can be published on the web as ready-to-use services in a matter of minutes. The services include not only a browser for human usage, but also Web Service and AJAX interfaces for concept finding, selecting and transporting resources from the ONKI SKOS Server to connected systems. Code generation services for AJAX and Web Service APIs are provided automatically, too. ONKI SKOS services are also used for semantic query expansion in information retrieval tasks. The idea of publishing ontologies as services is analogous to Google Maps. In our case, however, vocabulary services are provided and mashed-up in applications. ONKI SKOS was published in the beginning of 2008 and is to our knowledge the first generic SKOS server of its kind. The system has been used to publish and utilize some 60 vocabularies and ontologies in the National Finnish Ontology Service ONKI www.yso.fi.

1 Introduction

Thesauri and other controlled vocabularies are used primarily for improving information retrieval [1]. This is accomplished by using concepts or terms of a thesaurus in content indexing, content searching or in both of them, thus simplifying the matching of query terms and the indexed resources (e.g. documents) compared to using natural language. For users, such as content indexers and searchers, to be able to use thesauri, publishing and finding methods for them as well as methods for integrating them with applications [2] are needed [3]. Thesauri are of great benefit for the Semantic Web[1], enabling semantically dis-

[1] http://www.w3.org/2001/sw/

ambiguated data exchange and integration of data from different sources, though not to the same extent as ontologies [4] where the semantics of concepts are defined in more refined and "machine understandable" ways.

Publishing and utilizing thesauri is a laborious task because the representation formats of thesauri and features they provide differ from each other. When utilizing thesauri one has to know how to locate a given thesaurus, and also be familiar with the software the thesaurus is published with. A thesaurus can even be published as a plain text file or even worse, as a paper document, with no proper support for utilizing it. In such a case the users have to implement applications for processing the thesaurus in order to exploit it. Therefore, standard ways for expressing and publishing thesauri would greatly facilitate the publishing and utilizing processes of thesauri.

The Simple Knowledge Organization System (SKOS)[2] [5] being developed within W3C is a data model and a syntax for expressing concept schemes such as thesauri. SKOS provides a standard way for creating vocabularies and migrating existing vocabularies to the Semantic Web. SKOS solves the problem of diverse, non-interoperable thesaurus representation formats by offering a standard convention for presentation. Existing thesauri can be transformed into SKOS format via conversion processes. When a thesaurus is expressed as a SKOS vocabulary, it can be published as a RDF file on the web, allowing the vocabulary users to fetch the files and process them in a uniform way. However, this does not solve the problem of users having to implement their own applications for processing vocabularies.

This paper argues that there are many common, sharable tasks in such vocabulary-aware applications related to e.g. term/concept finding, browsing, selecting, fetching, and query expansion. Lots of work and costs can be saved by implementing such functionalities in standard ways [6] and by providing them for production use as ready-to-use services without having to reimplement the functionalities separately for each local application case. In this way, the use patterns of utilizing vocabularies in the user interface can be harmonized, which makes systems easier to learn and use.

Ontology servers [7, 8] have been proposed for publishing ontologies and vocabularies on the Semantic Web. Such servers are used for managing ontologies and offering users access to them. For accessing SKOS vocabularies, there is the Web Service based SKOS API[3] developed in the SWAD-Europe project, and the terminology service by Tudhope et al. [9]. There is also ongoing research by the Networked Knowledge Organization Systems/Services (NKOS) community[4] that focuses on enabling knowledge organization systems as networked interactive information services to support the description and retrieval of diverse information resources through the Internet. However, general tools for providing out-of-the-box support for utilizing SKOS vocabularies in, e.g., content indexing,

[2] http://www.w3.org/TR/skos-reference/

[3] http://www.w3.org/2001/sw/Europe/reports/thes/skosapi.html

[4] http://nkos.slis.kent.edu/

without needing to implement application specific user interfaces for end users do not exist.

This paper fills this gap by presenting the ONKI SKOS Server for publishing and utilizing thesauri and lightweight ontologies. In the following, ways of presenting thesauri on the Semantic Web are first overviewed. After this, the approach and implementation of the ONKI SKOS Server is presented followed by descriptions of use cases of its services. In conclusion, contributions of the work are summarized and directions for further research outlined.

2 Presenting Thesauri on the Semantic Web

W3C's SKOS data model provides a vocabulary for expressing the basic structure and contents of concept schemes, such as thesauri, classification schemes and taxonomies. The concept schemes are expressed as RDF graphs by using classes and properties defined in the SKOS specification, thus making thesauri compatible with the Semantic Web. SKOS is capable of representing vocabularies which loosely conform to the influential ISO 2788 thesaurus standard [6].

Although semantically richer RDFS/OWL ontologies enable more extensive ways to perform logical inferencing than SKOS vocabularies, in several cases thesauri represented with SKOS are sufficient. In our opinion, the first and the most obvious benefit of using Semantic Web ontologies/vocabularies in content indexing is their ability to disambiguate concept references in a universal way. This is achieved by using persistent URIs as the identification mechanism and is a tremendous advantage when compared with the traditional idea of controlled vocabularies where plain concept labels are used as identifiers. In such a case, identification problems can be encountered e.g. with polysemous and homonymous labels, when dealing with multi-lingual resources, and with resources whose labels may change as time goes by. For example, human names are transliterated in many ways in different languages, may change due to marriage, a person may have nick names, and so on. In such cases the labels of concepts are not a permanent identification method, and the references to the concepts may become ambiguous or invalid.

URIs provide not only an identification mechanism, but also means for accessing the concept definitions and thesauri, when using the http URI Scheme[5]. Such URIs can act as URLs, and with a proper server configuration provide the users additional information about the referred concept[6]. In addition to these general RDF characteristics, SKOS provides a way for expressing relations between concepts suitable for the needs of thesauri, thus providing conceptual context for concepts.

In short, using a common representation model such as SKOS for thesauri greatly reduces the cost of (a) sharing thesauri, (b) using different thesauri in conjunction within one application, and (c) development of standard software to process them [6].

[5] http://www.iana.org/assignments/uri-schemes.html

[6] http://www.w3.org/TR/swbp-vocab-pub/

3 Utilizing Thesauri with ONKI SKOS Services

ONKI SKOS is an ontology server implementation for publishing and utilizing thesauri and lightweight ontologies. It conforms to the general ONKI vision and API, and is thus usable via ONKI ontology services as easily integrable user interface components and APIs [2].

The Semantic Web applications typically use ontologies which are either straightforward conversions of well-established thesauri, application-specific vocabularies or semantically richer ontologies, that can be presented and accessed in similar ways to thesauri [3, 10]. Since SKOS defines a suitable model for expressing thesauri, it was chosen as the primary data model supported by the ONKI SKOS Server.

ONKI SKOS can be used to browse, search and visualize any vocabulary conforming to the SKOS specification, and also RDFS/OWL ontologies with additional configuration. ONKI SKOS does simple reasoning, e.g., transitive closure over class and part-of hierarchies. The implementation has been piloted using various thesauri and ontologies from diverse domains. Piloted ontologies include ontologies for cultural heritage (Art & Architecture Thesaurus AAT[7], 28,000 concepts; Iconclass[8], 27,000 concepts), health ontologies (Medical Subject Headings MeSH[9], 24,000 concepts), business ontologies (United Nations Standard Products and Services Code UNSPSC[10], 21,000 concepts), geographical instance ontologies (The Getty Thesaurus of Geographic Names TGN[11], 143,000 concepts), upper ontologies (General Finnish Upper Ontology YSO [3], 21,000 concepts), governmental ontologies (Finnish Governmental Thesaurus VNAS, 6,300 concepts) and natural science taxonomies (Ontology of Birds of the World AVIO, 11,000 concepts; Ontology of Mammals of the World MAMO, 5,000 concepts). However, not all of these ontologies are freely accessible due to licensing issues.

When utilizing thesauri represented as SKOS vocabularies and published on the ONKI SKOS server, several benefits are gained. Firstly, SKOS provides a universal way of expressing thesauri. Thus processing different thesauri can be done in the same way, eliminating the use of thesaurus specific processing rules in applications or separate converters between various formats. Secondly, ONKI SKOS provides access to all published thesauri in the same way, so content indexers and end-users do not have to use thesaurus specific implementations of thesaurus browsers and other tools developed by different parties, which is the predominant way. Also, one of the goals of the ONKI ontology services is that all the essential ontologies/thesauri can be found at the same location, thus eliminating the need to search for other thesaurus sources.

The typical way to use thesaurus specific publishing systems in content indexing and searching is either by using their browser user interface for finding

[7] http://www.getty.edu/research/conducting_research/vocabularies/aat/

[8] http://www.iconclass.nl/

[9] http://www.nlm.nih.gov/mesh/meshhome.html

[10] http://www.unspsc.org/

[11] http://www.getty.edu/research/conducting_research/vocabularies/tgn/

desired concepts and then copying and pasting the concept label to the used indexing system[12], or by using APIs for accessing and querying the thesaurus [9]. Both methods have some drawbacks. The first method introduces rather uncomfortable task of constant switching between two applications and the clumsy copy-paste procedure. The second method leaves the implementation job of the user interface entirely to the parties utilizing the thesaurus.

While ONKI SKOS supports both the aforementioned thesauri utilizing methods, in addition, as part of the ONKI ontology services, it provides a lightweight web widget for integrating general thesauri accessing functionalities into web based applications on the user interface level. The ONKI Selector widget depicted in Figure 1 can be used to search and browse thesauri, fetch URI references and labels of desired concepts and storing them in a concept collector. Similar ideas have been proposed by Hildebrand et al. [11] for providing search widget for general RDF repositories, and by Vizine-Goetz et al. [12] for providing widget for accessing thesauri through the side bar of the Internet Explorer web browser.

When the desired concepts have been selected with the ONKI Selector they can be stored into, e.g., the database of the application by using an HTML form. Either the URIs or the labels of the concepts can be transferred into the application providing support for the Semantic Web and legacy applications. For browsing the context of the concepts in thesauri, the ONKI SKOS Browser can be opened by pressing a button. Desired concepts can be fetched from the browser to the application by pressing the “Fetch Concept” button. Thus, there is no need for copy-paste procedures or user interface implementation projects.

For information retrieval use cases, ONKI SKOS provides support for finding suitable query terms and expanding them by using ontological inference, e.g., based on the concept subsumption, partonymy or associative relations between concepts. In legacy systems, which do not support URIs, the labels of the concepts can be used as query terms. In this context, ONKI SKOS supports multilingual search. If a thesaurus contains, e.g., English and Finnish labels for the concepts, the relevant query concepts can be searched in English, while in the actual information retrieval task the used query terms are in Finnish. This is useful when the end-user does not understand the language of the terms used in indexing the content of the underlying system.

The ONKI SKOS Browser (see Figure 2) is the graphical user interface of ONKI SKOS. It consists of three main components: 1) *semantic autocompletion concept search*, 2) *concept hierarchy* and 3) *concept properties*. When typing text to the search field, a query is performed to match the concepts’ labels. The result list shows the matching concepts, which can be selected for further examination.

When a concept is selected, its concept hierarchy is visualized as a tree structure. The ONKI SKOS Browser supports multi-inheritance of the concepts (i.e. a concept can have multiple parents). Whenever a multi-inheritance structure is met, a new branch is formed to the tree. This leads to cloning of nodes, i.e.

[12] This is the way the Finnish General Thesaurus YSA has been used previously via the VESA Web Thesaurus Service, `http://vesa.lib.helsinki.fi/`.

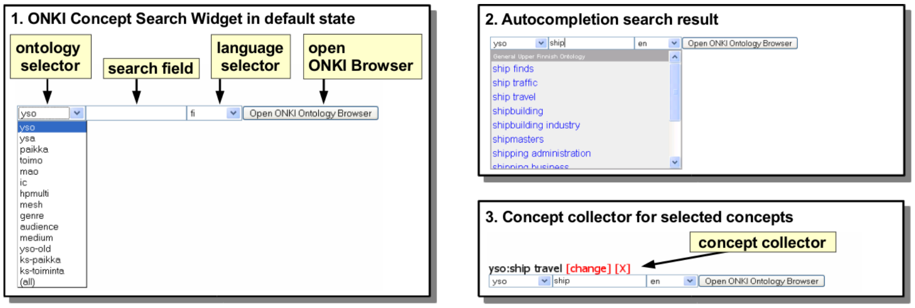


Fig. 1. ONKI Selector for concept finding.

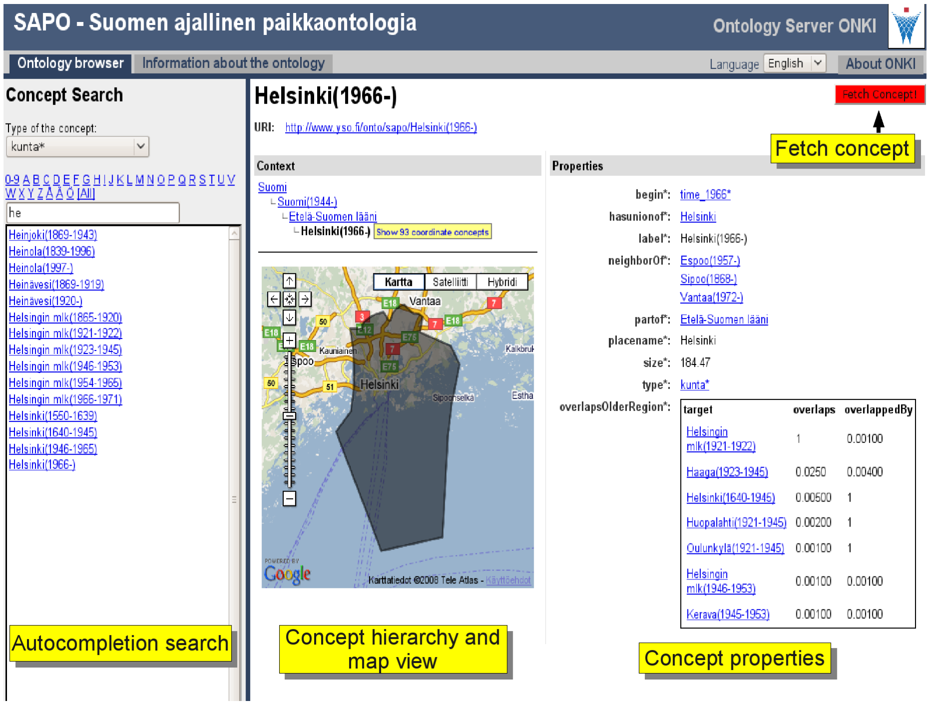
a concept can appear multiple times in the hierarchy tree. Next to the concept hierarchy tree, the properties of the selected concept are shown in the user interface. Mappings and other relations between concepts act as links allowing user to browse thesauri. In context of geographical ontologies, the place instances can be visualized on a Google Maps[13] view, as depicted in the Figure 2, if the instances contain coordinate information.

The National Finnish Ontology Service ONKI[14] provides documentation and tools for software architects developing ontology-based applications and integrating ONKI services into them [13]. With ONKI Selector Builder a developer can set the desired configuration properties for the ONKI Selector and generate the needed JavaScript code to integrate the ONKI Selector into an application. Also, a hands-on demonstration page for integrating the ONKI Selector is available. The page contains a HTML form, into which the user can type HTML/JavaScript code and see how the resulting page will look like in the end-user's web browser.

The Web Service and AJAX interfaces of the ONKI SKOS server can be used for querying for concepts by label matching, performing query expansion, getting label for a given URI and for querying for supported languages and type URIs of a thesaurus. For testing Web Service methods, a demonstration page is provided at the ONKI service. One can try out the methods of the ONKI API and examine the resulting SOAP request and response messages.

ONKI SKOS is implemented as a Java Servlet using the Jena Semantic Web Framework[15], the Direct Web Remoting library[16] and the Lucene[17] text search engine.

[13] http://maps.google.com/
[14] http://www.yso.fi/
[15] http://jena.sourceforge.net/
[16] http://directwebremoting.org/
[17] http://lucene.apache.org/java



Concept Search

Type of the concept: kunta*

0-9 A B C D E F G H I J K L M N O P Q R S T U V W X Y Z Å Ä Ö [All]

he

- Heinjoki(1869-1943)
- Heinola(1839-1996)
- Heinola(1997-)
- Heinävesi(1869-1919)
- Heinävesi(1920-)
- Helsingin mlk(1865-1920)
- Helsingin mlk(1921-1922)
- Helsingin mlk(1923-1945)
- Helsingin mlk(1946-1953)
- Helsingin mlk(1954-1965)
- Helsingin mlk(1966-1971)
- Helsinki(1550-1639)
- Helsinki(1640-1945)
- Helsinki(1946-1965)
- Helsinki(1966-)

Helsinki(1966-)

URI: http://www.yso.fi/onto/sapo/Helsinki(1966-)

Context

Suomi
└Suomi(1944-)
 └Etelä-Suomen lääni
  └Helsinki(1966-) Show 93 coordinate concepts

Properties

begin*:	time_1966*
hasunionof*:	Helsinki
label*:	Helsinki(1966-)

neighborOf*:	Espoo(1957-)
	Sipoo(1868-)
	Vantaa(1972-)

partof*:	Etelä-Suomen lääni
placename*:	Helsinki
size*:	184.47
type*:	kunta*
overlapsOlderRegion*:	

target	overlaps	overlappedBy
Helsingin mlk(1921-1922)	1	0.00100
Haaga(1923-1945)	0.0250	0.00400
Helsinki(1640-1945)	0.00500	1
Huopalahti(1921-1945)	0.00200	1
Oulunkylä(1921-1945)	0.00100	1
Helsingin mlk(1946-1953)	0.00100	0.00100
Kerava(1945-1953)	0.00100	0.00100

Fig. 2. The ONKI SKOS Browser.

4 Configuring ONKI SKOS with SKOS Structures

ONKI SKOS supports straightforward loading of SKOS vocabularies with minimal configuration needs. For using other data models than SKOS, various configuration properties are specified to enable ONKI SKOS to process the thesauri/ontologies as desired. The configurable properties include the properties used in hierarchy generation, the properties used to label the concepts, the concept to be shown in the default view and the default concept type used in restricting the concept search.

When the ONKI SKOS Browser is accessed with no URL parameters, information related to the concept configured to be shown as default is shown. Usually this resource is the root resource of the vocabulary, if the vocabulary forms a full-blown tree hierarchy with one single root. In SKOS concept schemes the root resource is the resource representing the concept scheme itself, i.e. the instance of *skos:ConceptScheme*.

The concept hierarchy of a concept is generated by traversing the configured properties. In SKOS these properties are *skos:narrower* and *skos:broader* and they are used to express the hierarchical relations between concepts. Hierarchical relations between the root resource representing the concept scheme and the top concepts of the concept scheme are defined with the property *skos:hasTopConcept*.

Labels of concepts are needed in visualizing search results, concept hierarchies, and related concepts in the concept property view. In SKOS the labels are expressed with the property *skos:prefLabel*. The label is of the same language as the currently selected user interface language, if such a label exists. Otherwise any label is used.

The semantic autocompletion search of ONKI SKOS works by searching for concepts whose labels match the search string. To support this, the labels of the concepts are indexed. The indexed properties can be configured. In SKOS these properties are *skos:prefLabel*, *skos:altLabel* and *skos:hiddenLabel*. When the user searches, e.g., with the search term "cat", all concepts which have one of the aforementioned properties with values starting with the string "cat" are shown in the search results. The autocompletion search also supports wildcards, so a search with a string "*cat" returns the concepts which have the string "cat" as any part of their label.

If a SKOS vocabulary contains concepts that are instances of subclasses of skos:Concept, the search can be limited to certain types of concepts only. To accomplish this, the types of the concepts (which are expressed with the property *rdf:type*) are indexed. It is also possible to limit the search to a certain subtree of the concept hierarchy by restricting the search to subconcepts of a specific concept. To support this, also the parents of concepts are indexed.

Many thesauri include structures for representing categories of concepts. To support category-based concept search in the ONKI SKOS Browser, another search field is provided. When a category is selected from the category search view, the concept search is restricted to the concepts belonging to the selected category. SKOS includes a concept collection structure, *skos:Collection*, which can be used for expressing such categories. However, *skos:Collection* is often used for slightly different purposes, namely for node labels[18]. Thus, instances of *skos:Collection* are not used for category-based concept search by default.

5 Converting Thesauri to SKOS—Case YSA

Publishing a thesaurus in the ONKI SKOS server is straightforward. To load a SKOS vocabulary into the server, only the location path of the RDF file of the vocabulary needs to be configured manually. After rebooting the ONKI SKOS, the RDF file is loaded, indexed and made accessible for users. ONKI SKOS provides the developers of thesauri a simple way to publish their thesauri.

There exists quite amount of well-established keyword lists, thesauri and other non-RDF controlled vocabularies which have been used in traditional approaches in harmonizing content indexing. In order to reuse the effort already invested developing these resources by publishing these vocabularies in ONKI

[18] A construct for displaying grouping concepts in systematic displays of thesauri. They are not actual concepts, and thus they should not be used for indexing. An example node label is "milk by source animal".

SKOS server, conversion processes need to be developed. This idea has also been suggested by van Assem et al. [6] and Summers et al. [14]. We have implemented transformation scripts for, e.g., MARCXML format[19], XML dumps from SQL databases and proprietary XML schemas.

An example of the SKOS transformation and publishing process is the case of YSA, the Finnish General Thesaurus[20]. YSA is developed by the National Library of Finland and is the most widely used controlled vocabulary in Finland. YSA is published in the VESA Web Thesaurus Service, where it receives over 12 million user hits per year. YSA is simultaneously published in the ONKI SKOS, currently in a pilot phase. During this year, the operation of the VESA service will be ended, and the ONKI SKOS becomes the official publishing channel of YSA.

To begin the SKOS transformation process, YSA is exported into MARCXML format from the thesaurus management system of the National Library of Finland. The resulting constantly up-to-date file is published at the web server of the National Library of Finland, from where it is fetched via OAI-PMH protocol[21] to our server. This process is automated and the new version of the XML file is fetched daily. Instead of fetching the full file every day, the incremental update feature of the OAI-PMH could be used, but we have not tested it. After fetching a new version of the file, the transformation process depicted in Figure 3 is started by loading the MARCXML file (ysa.xml). The Java-based converter first creates the necessary structure and namespaces for the SKOS model utilizing Jena Semantic Web Framework. Next, the relations in YSA are translated into their respective SKOS counterparts, which is depicted in Figure 4.

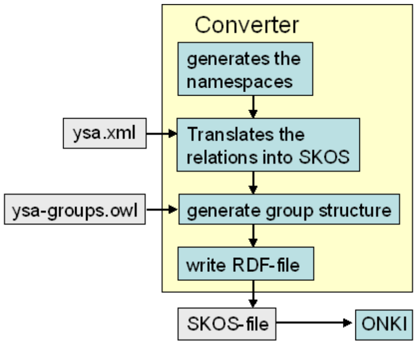
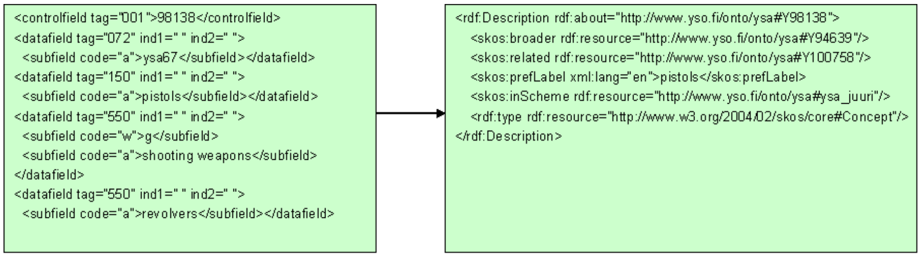


Fig. 3. The SKOS transformation process of YSA.

[19] http://www.loc.gov/standards/marcxml/

[20] http://www.nationallibrary.fi/libraries/thesauri/ysa.html

[21] http://www.openarchives.org/OAI/openarchivesprotocol.html



```
<controlfield tag="001">98138</controlfield>
<datafield tag="072" ind1=" " ind2=" ">
  <subfield code="a">ysa67</subfield></datafield>
<datafield tag="150" ind1=" " ind2=" ">
  <subfield code="a">pistols</subfield></datafield>
<datafield tag="550" ind1=" " ind2=" ">
  <subfield code="w">g</subfield>
  <subfield code="a">shooting weapons</subfield>
</datafield>
<datafield tag="550" ind1=" " ind2=" ">
  <subfield code="a">revolvers</subfield></datafield>
```

```
<rdf:Description rdf:about="http://www.yso.fi/onto/ysa#Y98138">
    <skos:broader rdf:resource="http://www.yso.fi/onto/ysa#Y94639"/>
    <skos:related rdf:resource="http://www.yso.fi/onto/ysa#Y100758"/>
    <skos:prefLabel xml:lang="en">pistols</skos:prefLabel>
    <skos:inScheme rdf:resource="http://www.yso.fi/onto/ysa#ysa_juuri"/>
    <rdf:type rdf:resource="http://www.w3.org/2004/02/skos/core#Concept"/>
</rdf:Description>
```

Fig. 4. An example of the SKOS transformation of YSA.

A URI for the new concept entry is created through the unique ID in the source file. The preferred and alternative labels are converted straightforwardly from one syntax to another. Similarly the type and scheme definitions are added to the SKOS model. Since the relations in the MARCXML refer not to the identifiers but rather to the labels, the source file is searched for an entry that has the given label and then its ID is recorded for the SKOS relation.

MARCXML record for the concept "pistols" can be seen in the left part of Figure 4. The ID of the concept is denoted by the *controlfield* with *tag* attribute's value "001" ("98138"). This ID value becomes the local name of the URI of the corresponding SKOS concept. The category in which the concept belongs to is depicted with the *datafield* with *tag* "072" ("ysa67"). The preferred label is the value provided in the *datafield* with *tag* value "150" ("pistols"). Hierarchical and associative relations are marked with *datafield* with *tag* value "550", while the *subfield*'s attribute *code* specifies which one of the relations is in question.

Once the SKOS transformation is ready, the converter fetches the labels for the concept categories from a separate file (ysa-groups.owl) - these labels are not included in the MARCXML file. Finally, a RDF file is written and imported into ONKI SKOS.

6 Use Cases of ONKI SKOS

ONKI SKOS has been tested in several content indexing and information retrieval pilot applications.

6.1 Content Indexing

The browser-based annotation editor SAHA[22] [15] has been used for creating semantic content for semantic portals. The two major applications demonstrating Semantic Web technologies are HealthFinland[23] [16] in the health promotion

[22] http://demo.seco.tkk.fi/saha/?lang=en

[23] http://demo.seco.tkk.fi/tervesuomi/

context, and CultureSampo[24] [17] in the cultural heritage context. Both systems use the SAHA editor with the services provided by ONKI SKOS and the National Finnish Ontology Service ONKI. ONKI Selector has been integrated into SAHA for finding suitable concepts in content indexing.

The web laboratory Owela[25] of the VTT Technical Research Centre of Finland has implemented a service for collecting and sharing text and image clips from the Web[26]. In the service one can organize the clips into folders and tag them with different categories. The ONKI Selector is used for tagging the clips keywords from vocabularies published in the ONKI service.

6.2 Information Retrieval

Kantapuu.fi[27] is a web user interface for browsing and searching for collections of Finnish museums of forestry. The collection items are annotated with Finnish terms from General Finnish Thesaurus YSA, Thesaurus for Museum Domain MASA and Agriforest Thesaurus. Kantapuu.fi search page is a web form into which query strings are typed as free text. The query strings can be placed into specific fields, e.g. "keywords", "place of use" or "time of use". We have created a demonstration page containing a Kantapuu.fi's search form with integrated ONKI Selectors which can be used for selecting query terms to be used in the Kantapuu.fi search[28]. The ONKI Selector is used for finding terms from vocabularies of the ONKI service. The used vocabularies are the same as those used in the annotation process of the items, or actually their ontologized versions. To find suitable query terms user can utilize the autocompletion search functionality or the ONKI Browser. Thus, the user does not need to be familiar with the vocabularies used in the annotations of the items, as in the case of free text search. The ONKI Selector performs query expansion based on the selected query terms. So, for example a query term "animals" would return items annotated with term "cats". When the desired query terms are selected, the actual search to the Kantapuu.fi system can be executed. The users of the pilot system have given positive feedback especially on the multilingual search possibility.

The ONKI Selector Widget has also been integrated into the Viikki Science Library[29] reference database system eViikki[30]. eViikki is a search interface for the library's collections, which consist of scientific literature on agriforestry. The literature is annotated with terms from General Finnish Thesaurus YSA and Agriforest vocabulary. The ONKI Selector is used for populating the "keywords" field of the search form of eViikki. The fetched concept labels are used in the information retrieval task. Query expansion is not performed currently.

[24] `http://www.kulttuurisampo.fi/`

[25] `http://owela.vtt.fi/`

[26] `http://owela.vtt.fi/tilkut`

[27] `http://www.kantapuu.fi/`

[28] `http://www.yso.fi/lusto`

[29] `http://www.tiedekirjasto.helsinki.fi/english/`

[30] `http://www-db.helsinki.fi/eviikki/eviikkihaku.html`

7 Discussion

The main contribution of this paper is to show, both in principle and as a deployed implemented service in use, how thesauri can be published and utilized easily on the Semantic Web, emphasizing the benefits of the use of W3C's SKOS data model as a uniform vocabulary representation framework. The ONKI SKOS server is presented as a proof of concept for a cost-efficient thesauri utilization method. By using ONKI SKOS, general thesauri accessing functionalities can be easily integrated into applications without the need for users to reimplement their own user interfaces for this.

The utilization of the SKOS structures in an ontology server was described in context of the ONKI SKOS server. The case of the Finnish General Thesaurus acts as one example on how an existing, widely used thesaurus can be converted into the SKOS format and be published on the ONKI SKOS server. At the moment some 60 international and national vocabularies, taxonomies, and lightweight ontologies have been published online using ONKI SKOS and the number is increasing. Finally, in order to demonstrate end-user benefits of ONKI SKOS services, we provided some real world use cases of ONKI SKOS in content indexing and information retrieval.

Future work for ONKI SKOS includes creating a more extensive Web Service interface for supporting, e.g., querying for properties of a given concept and for discovering concepts which are related to a given concept. The starting point for this API will be the SKOS API. Also, version control techniques and other support for vocabulary management (e.g. protocol for communicating vocabulary changes to users) would be needed to support the development phase of vocabularies.

Acknowledgements

We thank Ville Komulainen for his work on the original ONKI server. This work is a part of the National Semantic Web Ontology project in Finland[31] (FinnONTO) and its follow-up project Semantic Web 2.0[32] (FinnONTO 2.0, 2008-2010), funded mainly by the National Technology and Innovation Agency (Tekes) and a consortium of 38 private, public and non-governmental organizations.

References

1. Aitchison, J., Gilchrist, A., Bawden, D.: Thesaurus Construction and Use: A Practical Manual. 4th edn. Europa Publications (2000)
2. Viljanen, K., Tuominen, J., Hyvönen, E.: Publishing and using ontologies as mash-up services. In: Proceedings of the 4th Workshop on Scripting for the Semantic Web (SFSW 2008), 5th European Semantic Web Conference 2008 (ESWC 2008), Tenerife, Spain (June 1-5 2008)

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

[32] http://www.seco.tkk.fi/projects/sw20/

3. Hyvönen, E., Viljanen, K., Tuominen, J., Seppälä, K.: Building a national semantic web ontology and ontology service infrastructure—the FinnONTO approach. In: Proceedings of the 5th European Semantic Web Conference (ESWC 2008). (June 1-5 2008)
4. Staab, S., (eds.), R.S.: Handbook on ontologies. Springer-Verlag (2004)
5. Miles, A., Matthews, B., Wilson, M., Brickley, D.: SKOS Core: Simple knowledge organisation for the web. In: Proceedings of the International Conference on Dublin Core and Metadata Applications (DC 2005), Madrid, Spain (September 12-15 2005)
6. van Assem, M., Malaisé, V., Miles, A., Schreiber, G.: A method to convert thesauri to SKOS. In: Proceedings of the third European Semantic Web Conference (ESWC 2006), Budva, Montenegro (June 11-14 2006) 95–109
7. Ding, Y., Fensel, D.: Ontology library systems: The key to successful ontology reuse. In: Proceedings of SWWS'01, The first Semantic Web Working Symposium, Stanford University, USA. (August 2001) 93–112
8. Ahmad, M.N., Colomb, R.M.: Managing ontologies: a comparative study of ontology servers. In: Proceedings of the eighteenth Conference on Australasian Database (ADC 2007), Ballarat, Victoria, Australia, Australian Computer Society, Inc. (January 30 – February 2 2007) 13–22
9. Tudhope, D., Binding, C.: Towards terminology service: experiences with a pilot web service thesaurus browser. In: Proceedings of the International Conference on Dublin Core and Metadata Applications (DC 2005), Madrid, Spain (September 12-15 2005) 269–273
10. van Assem, M., Menken, M.R., Schreiber, G., Wielemaker, J., Wielinga, B.: A method for converting thesauri to RDF/OWL. In: Proceedings of the Third International Semantic Web Conference (ISWC 2004), Hiroshima, Japan (November 7-11 2004) 17–31
11. 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, Centrum voor Wiskunde en Informatica (CWI), Amsterdam (2007)
12. Vizine-Goetz, D., Childress, E., Houghton, A.: Web services for genre vocabularies. In: Proceedings of the International Conference on Dublin Core and Metadata Applications (DC 2005), Madrid, Spain (September 12-15 2005)
13. Viljanen, K., Tuominen, J., Hyvönen, E.: Ontology libraries for production use: The Finnish ontology library service ONKI. In: Proceedings of the 6th European Semantic Web Conference (ESWC 2009). (May 31 - June 4 2009)
14. Summers, E., Isaac, A., Redding, C., Krec, D.: LCSH, SKOS and Linked Data. In: Proceedings of the International Conference on Dublin Core and Metadata Applications (DC 2008), Berlin, Germany (September 22-26 2008)
15. 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)
16. Hyvönen, E., Viljanen, K., Suominen, O.: HealthFinland—Finnish health information on the semantic web. In: Proceedings of the 6th International Semantic Web Conference (ISWC 2007), Busan, Korea, Springer-Verlag (Nov 2007)
17. 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 Robering, K., ed.: Information Technology for the Virtual Museum. LIT Verlag, Berlin. (Nov 2007)