Building a National Semantic Web Ontology and Ontology Service Infrastructure —The FinnONTO Approach

Eero Hyvönen, Kim Viljanen, Jouni Tuominen, and Katri Seppälä

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

Abstract. This paper presents the vision and results of creating a national level cross-domain ontology and ontology service infrastructure in Finland. The novelty of the infrastructure is based on two ideas. First, a system of open source core ontologies is being developed by transforming thesauri into mutually aligned lightweight ontologies, including a large top ontology that is extended by various domain specific ontologies. Second, the ONKI Ontology Server framework for publishing ontologies as ready to use services has been designed and implemented. ONKI provides legacy and other applications with ready to use functionalities for using ontologies on the HTML level by Ajax and semantic widgets. The idea is to use ONKI for creating mash-up applications in a way analogous to using Google or Yahoo Maps, but in our case external applications are mashed-up with ontology support.

1 A National Ontology Infrastructure

The ambitious goal of the National Semantic Web Ontology project (FinnONTO 2003–2007)[1] [1] is to develop a semantic web infrastructure on a national level in Finland. The consortium behind the initiative—37 companies and public organizations—represents a wide spectrum of functions of the society, including libraries, health organizations, cultural institutions, government, media, and education. The project has produced a variety of scientific results, specifications, services, demonstrations, and applications:

1. *Metadata standards.* Nationally adapted standards for representing metadata in various application fields have been created, e.g. JHS 158[2] and [2].
2. *Core ontologies.* Several core ontologies[3] have been developed in order to initiate ontology development processes in Finland.

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

[2] http://www.jhs-suositukset.fi/suomi/jhs158, Public Recommendation for Geographic Metadata (in Finnish), Ministry of Internal Affairs.

[3] http://www.seco.tkk.fi/ontologies/

3. *Public ontology services.* An ontology library and web service framework ONKI[4] is being developed [3] to enable ontology usage in ontology development, content indexing, and information retrieval through public web and mash-up services.
4. *Tools for metadata creation.* A bottleneck limiting proliferation of the semantic web is production of metadata. For this purpose, semiautomatic content annotation tools have being developed [4, 5].
5. *Tools for semantic portal building.* Tools for semantic search and recommendation based on amalgamating the multi-facet search paradigm with semantic web technologies have been developed [6–8].
6. *Pilot applications.* The framework is being evaluated by implementing a number of practical semantic portal applications in the domains of eCulture (MuseumFinland [9] and CultureSampo [10, 11]), eHealth (HealthFinland [12, 13], eGovernment [14], eLearning [15], and eCommerce [16].

The vision of FinnONTO from the outset has been the idea of developing a national "semantic content infrastructure" for the semantic web. The core of the infrastructure consists of 1) an ontology library system of mutually aligned central ontologies and 2) the ONKI ontology server framework [17, 3] for cost-effective exploitation ontologies in legacy and other applications. The FinnONTO approach is based on international domain independent recommendations and best practices of W3C, such as RDF, SKOS, OWL etc., but builds, from the domain content perspective, upon existing national thesauri and vocabularies encompassing different application domains. It is argued that such a *cross-domain ontology system* is needed nationally on the web in same way as roads are needed for traffic and transportation, power plants and electrical networks are needed for energy supply, or GSM standards and networks are needed for mobile phones and wireless communication.

This paper presents the FinnONTO approach to developing a national cross-domain ontology infrastructure for the semantic web. In the following, the idea a creating a system of mutually aligned ontologies from thesauri is first elaborated. After this, the ONKI Ontology Service system is presented for application usage. The ontology framework is already operational on the web and is being used in creating the FinnONTO demonstration applications.

2 Creating a System of Mutually Aligned Ontologies

The traditional approach for harmonizing content indexing is to use keyword terms taken from shared vocabularies or thesauri [18, 19]. In Finland, for example, lots of thesauri conforming to the ISO [2788] thesaurus standard [19] and its Finnish version SFS [5471] are in use, such as YSA[5], MASA [20], or Agriforest[6].

[4] http://www.seco.tkk.fi/services/onki/

[5] http://vesa.lib.helsinki.fi/

[6] http://www-db.helsinki.fi/triphome/agri/agrisanasto/Welcomeng.html

In order to reuse the effort already invested in developing thesauri, it was therefore decided to develop a method of minimal changes for transforming thesauri into ontologies. Thesauri-based class ontologies form the core of FinnONTO ontologies, but there are also other important ontologies being developed in the projects, especially the geographic ontology SUO [21] that consists currently of some 800,000 places in Finland (and over 4 million places abroad), the historical place ontology SAPO [22], and the actor ontology TOIMO based on some 120,000 actors and organizations of the Universal List of Artist Names vocabulary (ULAN)[7] with some Finnish extensions [23], and a Finnish version of the iconographic vocabulary ICONCLASS[8].

2.1 A Method for Transforming Thesauri into Ontologies

The idea of transforming thesauri into ontologies was also suggested in [24] and by the SKOS initiative[9]. However, in contrast to these initiatives, we stress that although a syntactic transformation into SKOS can be useful, it is not enough from a semantic viewpoint. The fundamental problem with a traditional thesaurus [18, 19], is that its semantic relations have been constructed mainly to help the indexer in finding indexing terms, and understanding the relations needs implicit human knowledge. Unless the meaning of the semantic relations of a thesaurus is made more explicit and accurate for the computer to interpret, the SKOS version is equally confusing to the computer as the original thesaurus, even if semantic web standards are used for representing it.

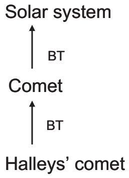

Fig. 1. An example of broader term hierarchy in the YSA thesaurus.

For example, consider the example of Figure 1 taken from the YSA thesaurus, where BT indicates the "broader term" relation used in thesauri. We can easily understand its meaning but the machine is confused: Is Halley's Comet an individual or a class of them, such as Comet? Can there be many Halley's comets or only one? Is a comet a kind of solar system or a part of a solar system? Is it a part as a concept or are all individual comets a part of some solar system? Do comets have the properties of solar systems, e.g. own planets, based on the BT

[7] http://www.getty.edu/research/conducting_research/vocabularies/ulan/
[8] http://www.iconclass.nl/
[9] http://www.w3.org/2004/02/skos/

relation. Using the BT relations for term expansion, a search for "solar systems" would retrieve comets although comets are not solar systems.

The idea of using ontologies is to define the meaning of indexing terms and concepts explicitly and accurately enough for the machine to use. This is essential in many application areas, such as semantic search, information retrieval, semantic linking of contents, automatic indexing, and in making contents semantically interoperable. With even a little extra work, e.g. by just systematically organizing concepts along subclass hierarchies and partonymies, substantial benefits can be obtained.

Therefore, our method for thesaurus-to-ontology transformation is not a syntactic one, but is done by refining and enriching the semantic structures of a thesaurus. As a general methodological guideline, criteria from DOLCE [25] were employed. However, the methodology is applied in a lightweight fashion in order to keep the job manageable.

The method used is based on making the following semantic refinements and extensions in the thesaurus structure:

1. *Missing links in the subclass-of hierarchy.* The BT relations do not (usually) structure the terms into a full-blown hierarchy but into a forest of separate smaller subhierarchies. In the case of YSA, for example, there were thousands of terms without any broader term. Many interesting relations between terms are missing in thesauri, especially concerning general terminology, where BT relations are not commonly specified in practice. As in DOLCE, the concepts on the top level were divided into three upper classes: 1) Abstract, Endurant, and Perdurant. A central structuring principle in constructing the hierarchies was to avoid multiple inheritance across major upper ontology categories.
2. *Ambiguity of the BT relation.* The semantics of the BT relation is ambiguous: it may mean either subclass-of relation, part-of relation (of different kinds, cf. [26]), or instance-of relation. This severely hinders the usage of the structure for reasoning [27]. For example, the BT relation cannot be used for property inheritance because this requires that the machine knows that BT means subclass-of and not e.g. part-of relation. In our method, existing BT relations were transformed into subclass-of and part-of relations, instance-of relations are not used. The difficult question of deciding what terms should be instances and what terms are classes is avoided by transforming even instance like terms, such as "Halleys' comet", into classes. The decision of whether to use classes or instances in annotations[10] is not made in the ontology but is left to the application developers for flexibility.
3. *Non-transitivity of the BT relation.* The transitivity of the BT relation chains is not guaranteed from the instance-class-relation point of view, when transforming BT relations into subclass-of relations. If x is an instance of a class A whose broader term is B, then it is not necessarily the case that x is an instance of B, although this a basic assumption in RDFS and OWL semantics

[10] Cf. http://www.w3.org/TR/swbp-classes-as-values/, W3C Working Group Note 5 April 2005

[28] if BT is regarded as the subclass-of relation. For example, assume that x is a "make-up mirror", whose broader term is "mirror", and that its broader term is "furniture". When searching with the concept "furniture" one would expect that instances of furniture are retrieved, but in this case the result would include x and other make-up mirrors, if transitivity is assumed. This means e.g. that term expansion in querying cannot be used effectively based on the BT relation in a thesaurus, but can be used in our corresponding lightweight ontology.

4. *Ambiguity of concept meanings.* Lots of terms in our thesauri are ambiguous and cannot be related properly with each other in the hierarchy using the subclass-of relation. For example, in YSA there is the indexing term "child". This term has several meanings such as "a certain period of human life" or "a family relation". For example, George W. Bush is not a child anymore in terms of age but is still a child of his mother, Barbara Bush. The computer cannot understand this and is confused, unless the meanings of "child" are separated and represented as different concepts (with different URIs) in different parts of the ontology.

In FinnONTO, the central ontology developed is the General Finnish Ontology YSO[11] [29]. YSO is based on the general Finnish keyword thesaurus YSA[12] that contains some 23,000 terms divided into 61 domain groups, such as Physics, History, etc. YSA is maintained by the National Library of Finland. Since YSA is widely used in Finnish organizations, YSO is an important step in solving semantic interoperability problems in Finland. The ontology is trilingual. Swedish translations of the YSO ontology labels were acquired from the Allärs thesaurus[13], and a translation of the terms into English was produced by the City Library of Helsinki. This makes it possible in the future to align YSO with international English ontologies of the semantic web.

In ontologizing the YSA thesaurus lots of terms turned out to be ambiguous, i.e., they could not be placed in one place in the hierarchy. In such cases the term had to be split into several concepts in YSO. However, a lesson learned in our work was that the general ambiguous concept encompassing several meanings, say "child", can be useful for indexing purposes and should be available in YSO. For example, assume a painting depicting children playing in a park with their mothers watching. When selecting keywords (concepts) describing the subject, it would be tedious to the indexer to consider all the meaning variants of "childness" in YSO, while the single ambiguous indexing term "child" of YSA would encompass them properly in this case. We therefore included some useful ambiguous concepts, such as "child", in YSO as special *aggregate indexing concepts.* They lay outside of the subclass-hierarchies but can be defined, e.g., in terms of them by using Boolean class expressions as in OWL[14].

[11] http://www.yso.fi/onto/yso/
[12] http://vesa.lib.helsinki.fi/
[13] http://vesa.lib.helsinki.fi/allars/index.html
[14] http://www.w3.org/TR/owl-ref/

A principle used in transforming YSA was that each YSA term should have a counterpart in YSO. This makes it possible to use YSO for more accurate reasoning about content annotated using YSA. Since the original term meanings in YSA change when the term is connected into an ontology, the original YSA terms had to be preserved in the YSO ontology as they are. YSO therefore consists of the following major parts: 1) a meaning preserving SKOS version of the original YSA, 2) an ontology of concepts corresponding to YSA terms, and 3) a mapping between the two structures.

The mapping makes it possible to explicitly tell the relation between YSO concepts and YSA terms. In our mapping schema, the relation between two concepts A and B is defined in terms of extensional overlap that can be expressed as two numerical values in the range [0,1]: 1) how much A overlaps B proportionally and 2) how much B overlaps A. This model is an adaptation of [22] where geographical overlap in area is considered. For example, if A is a subclass of B, then B overlaps A in meaning by 1, and A overlaps B by some value in the range [0,1]; equality means two overlap values 1, and partial overlaps can be expressed by selecting other values. In the first version of YSO equality of YSA and YSO concepts is used by default.

2.2 Aligning Ontologies

Thesauri are widely used for harmonizing content indexing. Different domain fields have thesauri of their own. The thesauri are typically developed by domain specific expert groups without much systematic collaboration with other fields. When using such thesauri in cross-domain environments, such as the web, semantic problems arise, e.g., due to ambiguity of literal word expressions. For example, in the finance domain the term "bank" has an obvious meaning as an institution, but when considering the nature or musical instrument domains, there are other meanings. In semantic web ontologies, the ambiguity problem is solved by dealing with unambiguous resources identified by URIs instead of literal words. However, support is needed for sharing the URIs across domains and users. If one needs to define the notion of "river bank", one should not mix this concept with "money bank". On the other hand, if one is defining the notion of "blood bank", it is possible to use the more general notion of "bank" and modify it, thus sharing this common notion with other kind of banks considered in other ontologies.

In focused domains and applications it may be possible to agree upon common ontological concepts, but in a larger cross-domain setting, this usually becomes more difficult. Different domains and applications may need different ontological representations even for the same real world objects, and different parties tend to have different philosophical opinions and needs on how to model the world. As a result, there is the danger that the global semantic web will not emerge but there will rather arise a set of isolated, mutually incompatible semantic web islands.

There are various complementary approaches for making semantic web ontologies interoperable. First, ontology mapping and alignment [30] can be used for mapping concepts with each other. Second, ontologies can share and be based on common foundational logical principles, like in DOLCE. This easily leads to complicated logical systems that may not scale up either epistemically or computationally to real word situations and practical usage. Third, horizontal top ontologies, such as the SUMO[15] can be created for aligning the concepts between vertical domain ontologies. Fourth, ontology engineering tools for creating ontologies in the first place as interoperable as possible can be created.

We adopted the idea that a shared top ontology is useful for enhancing semantic interoperability between various domain ontologies. In Finland the YSA thesaurus is widely used for content indexing in libraries, museums, and archives of various kinds both public and private. YSA can therefore be considered as a kind of semantic terminological "glue" between many other Finnish thesauri. Once the structure of the top ontology is defined, the same choices of hierarchical structures can be reused in many cases in the vertical ontologies that typically share lots of concepts with the top ontology. For example, when we created the cultural ontology MAO, based on the Finnish Cultural thesaurus MASA [20], about 2,000 out of MASA's 6,000 terms turned out to have a corresponding term in YSA. We now work e.g. on the Agriforest thesaurus[16], and on some other thesauri, where thousands of terms originate from YSA.

A simple method and a tool was created by which a Protégé[17] project is created containing an ontology O and YSO in their own namespaces. The classes of O are mapped with YSO classes having similar labels using equivalence, and for each class in O and YSO, subclasses from both ontologies are attached. Using this project, a human editor then checks the subclass-of chains along both ontologies by hand, and edits the structures. In contrast to the PROMPT Suite [31], our tool simply matches similar labels in subclass-of-hierarchies, and the focus is not to merge ontologies or create mappings, but rather to form an initial version of a double ontology project for maintaining an ontology O in accordance with another one (YSO).

Several ontologies have now been aligned with YSO pairwise. According to our experience, a lot of work is saved by reusing the work done with YSO in this way, and the resulting ontologies are aligned with each other at the same time with reasonable extra effort. Our goal is a system of mutually interlinked and aligned ontologies, as illustrated in Figure 2. In our vision, vertical domain ontologies add semantic depth to the top ontology. Interoperability is obtained by aligning the ontologies with each other.

3 Public Ontology Services

The Semantic Web is based on using shared ontologies for enabling semantically disambiguated data exchange between distributed systems on the web. This requires, from the ontology publisher's viewpoint, tools for publishing ontologies

[15] http://suo.ieee.org/

[16] http://www-db.helsinki.fi/eviikki/Welcome_eng.html

[17] http://protege.stanford.edu/

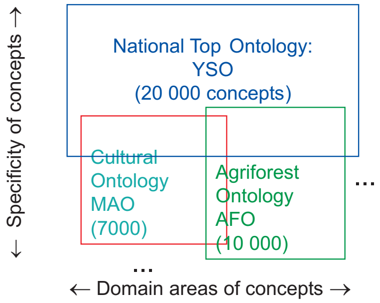

Fig. 2. YSO and related vertical ontologies intersect each other and share structures.

on the web to ensure the availability and acceptance of the ontologies. From the ontology user's viewpoint, support services are needed for utilizing ontologies easily and cost-effectively in the users' own systems that are typically legacy systems without ontology support. ONKI addresses both problems at the same time. It provides the publisher with a SKOS compatible server and an ontology browser whose functionalities, i.e., concept finding, semantic disambiguation, and concept fetching can be used in external applications as centralized ontology services [3]. A major contribution of ONKI is to provide these services as ready-to-use functionalities for creating "mash-up" applications very cost-efficiently. Two prototypes of the system—ONKI-SKOS [3] for all kinds of ontologies and ONKI-Geo [32] for geographical ontologies with a map mash-up interface—are operational on the web and are currently being used in several pilot applications.

3.1 ONKI Functionalities

One of the main lessons learned in our work on creating semantic portals [9, 14, 15, 12] is that metadata in data sources, such as museum databases, are often syntactically heterogeneous and contain typos, are semantically ambiguous, and are based on different vocabularies [33]. This results in lots of tedious syntactic correction, semantic disambiguation, and ontology mapping work when making the contents semantically interoperable, and when publishing them on the Semantic Web. A natural solution to this problem would be to enhance legacy cataloging and content management systems (CMS) with ontological annotation functions so that the quality of the original data could be improved and errors fixed in the content creation phase. However, implementing such ontological functions in existing legacy systems may require lots of work and thus be expensive, which creates a severe practical hindrance for the proliferation of the Semantic Web.

This relates to the more general challenge of the Semantic Web today: ontologies are typically published as files without much support for using them. The user is typically expected to open the files using some ontology editor, such as Protégé, have a closer look of the ontology, and then figure out whether it is of use to her. Once a suitable ontology is found, lots of programming effort is usually needed to utilize the ontology because it is likely to be used in a specific application and software environment. Even if the same ontological resources and structures could be shared by different content providers for interoperability, like in [9], it is not usually possible to share the *functionalities* of using the ontologies across applications. Instead, each application tends to re-implement generic functions for utilizing ontologies, such as semantic autocompletion and disambiguation [34], browsing and finding concepts, populating ontologies with new instances, etc. It is like re-creating map services from the scratch in different geographical web applications, rather than using available services such as Google Maps[18], Yahoo Maps[19], or Microsoft Live Search Maps[20].

The main functionalities of the ONKI service are 1) a *web widget for concept searching and fetching*, which is described in this section, and 2) *domain-specific ONKI Browsers*. The ONKI Browsers are user interfaces for searching and browsing ontologies, and they can be used independently or accessed via the web widget.

The general idea of the proposed mash-up approach is to provide the developer with a widget that can utilize ONKI services with minimal changes in the legacy system. In the case of an HTML-based legacy system, just a few lines of JavaScript code need to be added on the HTML page. In the case of other user interface technologies, the Web Service[21] interface can be used. The widget solves the problem of getting the right URIs into the application or a database; the actual usage of the acquired semantically correct data is the responsibility of the target application. Such a simple way for getting the URIs is crucial e.g. in various content creation systems for the semantic web, such as [33, 12].

The ONKI web widget on an HTML form is illustrated in Figure 3. The widget enables the user, e.g. a content annotator, to find correct ontological concepts to describe the content to be annotated. The user is provided with searching and browsing capabilities to aid this task. When the correct concept is found, its URI and label can be fetched to the target application. In addition to annotating content, the web widget can be used for supporting other tasks where ontological concepts need to be searched and fetched, such as content searching.

Part 1 of Figure 3 shows the default components of the widget. The ontology selector can be used to change the ontology used in search. At the moment, there are 14 different vocabularies and ontologies to choose from, including MeSH, Iconclass[22], the General Finnish Upper Ontology YSO[23] and HPMULTI[24]. The search field is used for inputting the query string. The language of concept labels used in matching the query string can be chosen by using the language

[18] http://maps.google.com/

[19] http://maps.yahoo.com/

[20] http://maps.live.com

[21] http://www.w3.org/TR/ws-arch/

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

[23] http://www.seco.tkk.fi/ontologies/yso/

[24] The European multilingual thesaurus on health promotion in 12 languages http://www.hpmulti.net/

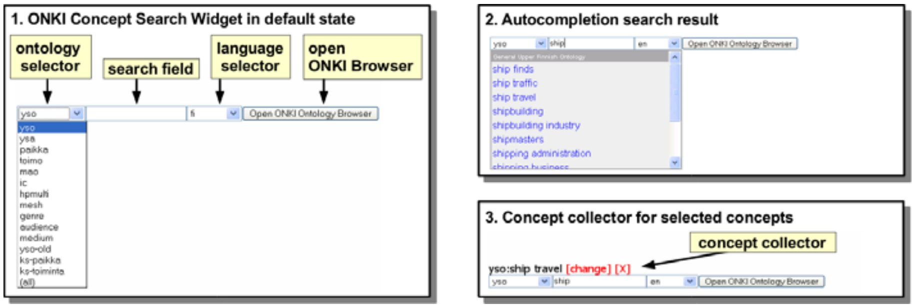

Fig. 3. ONKI Concept Search Widget.

selector. The choice of languages depends on the ontology selected. For example, for YSO, English and Swedish are supported in addition to Finnish, and the Finnish Geographic Ontology[25] can be used in Finnish, Swedish, and in three dialects of Sami languages spoken in Lapland. It is possible to use all languages simultaneously by selecting the option "all". The "Open ONKI Browser" button is used for opening the ONKI browser in a separate window.

The widget supports concept fetching to the target application. This can be done either by semantic autocompletion of the input field, or by pushing the "Open ONKI Browser" button for opening the ONKI Browser:

- *Using semantic autocompletion.* In part 2 of Figure 3, the user is typing a search string to the search field of the mash-up component. The system then dynamically performs a query after each input character (here "s-h-i-p-...") to the ONKI service, which returns the concepts whose labels match the string, given the language selection. The results of the query are shown in the web widget's result list below the input field. The desired concepts can be selected from the results. When selected, the concept's URI and label are fetched into the target application. In part 3 of Figure 3, the user has selected "ship travel" from the English version of the YSO ontology, and the concept's URI and label are stored onto the HTML page with the label shown, together with links `change` and `[X]`. In the case of a legacy application, which is not capable of handling URIs, only the labels of concepts can be fetched. By clicking the `change` link, it is possible to change the selected concept by using the ONKI browser (whose usage will be illustrated below). The annotation can be removed by clicking the link `[X]`.
- *Using ONKI Browser.* The alternative for using the autocompletion search, is to use the "Open ONKI Browser" button to search and browse concepts in a new ONKI Browser window. When the desired concept has been found by searching or browsing the ontology, the concept's URI and label are fetched into the application by pressing the "Fetch concept" button on the ONKI Browser page corresponding to the concept.

[25] http://www.seco.tkk.fi/ontologies/suo/

When the URI of a desired annotation concept is fetched, it is stored in a concept collector. The widget provides a default concept collector, but the concept collector can also be implemented in the target application. The default concept collector shows the fetched concepts in the widget's user interface, and also stores them in hidden input fields. When the form is submitted, the hidden input fields can be processed by the target application. This way the URIs of the annotation concepts can be transferred, e.g., into the database of the application. The body of a HTTP POST request message used to submit the form's hidden fields to the target application server looks like the following example below where two URIs are selected as values of the "dc:subject" field.

```
dc:subject=http://www.yso.fi/onto/yso/p14629&
dc:subject=http://www.yso.fi/onto/yso/p8038
```

3.2 Implementation

The web widget is implemented as an easily integrable Ajax component. The widget uses HTML and JavaScript for the user interface components, and the Direct Web Remoting (DWR)[26] library for the asynchronous cross-domain Ajax communication between the web browser and the ontology server. The DWR library enables the straightforward use of the ontology server's Java methods in the JavaScript code in the web browser.

An input field in e.g. a cataloging system of a museum could be defined like this:

```
<input id="dc:subject"/>
```

The web widget can be integrated into this example system by adding the following lines of HTML/JavaScript code into the HTML page.

```
1) <script language="javascript" type="text/javascript"
 src="http://www.yso.fi/onki.js"></script>

2) <input id="dc:subject"
 onkeyup="onki['yso'].search()"/>
```

The code line 1) is used to load the needed ONKI library files and is typically added to the HEAD section of the HTML page. The code line 2) is added to the BODY section of the HTML page in the locations where the ONKI widget component is desired. The string "yso" in the code line 2) refers to the ontology server instance used in the search.

When a page containing the integration code is accessed, the ONKI JavaScript library files are loaded into the web browser. When loaded, the JavaScript library generates the user interface components of the web widget into the desired

[26] http://getahead.org/dwr

locations of the page. In this way, plain input text fields are transformed into ONKI web widgets.

The web widget can be customized e.g. by hiding the ontology or the language selection menus, the search field, or the "Open ONKI Browser" button. The desired customizations can be defined in the integration code[27]. The appearance of the web widget can be modified by CSS rules, e.g. by using the class attributes of the HTML elements of the widget, overriding the default ones.

We have defined an ONKI Java API that has to be implemented by the domain-specific ONKI servers for them to be used with the web widget. The API includes the following methods:

- *search(query, lang, maxHits, type, parent)* - for searching for ontological concepts.
- *getLabel(URI, lang)* - for fetching a label for a given URI in a given language.
- *getAvailableLanguages()* - for querying for supported languages of an ontology.

By implementing the shared API, different domain-specific ONKI servers can be used by one single web widget. The widget can even be used to query multiple ontology servers within the same query. In addition for the use of the web widget, the ONKI API has been published as a Web Service conforming to the SOAP[28] standard.

The ontologies in an ONKI service are also published as RDF files to support e.g. semantic web applications that perform complex ontological inferencing. Such applications need access to complete ontology files to be able to process the ontologies in their entirety. When a new version of an ontology is to be published in an ONKI service, the ontology file is tagged with a current date. This date forms part of the URI used for identifying the corresponding version of the ontology file, and this URI can be used as an URL for locating and downloading the RDF file. For every published ontology there is also a static URI which identifies the latest version of the ontology's source file.

4 Discussion and Related Work

FinnONTO aims to build an open source ontology infrastructure that consists of a set of mutually aligned ontologies from different application domains. The core of the system is a horizontal top ontology and related vertical domain ontologies extend its class hierarchy in different directions and application domains. This approach contributes to earlier work on creating ontology infrastructures that has focused either on developing domain independent standards on the international level (e.g. W3C activities), on developing single global ontologies, such as the Open Directory Project[29] or OpenCyc[30], or on developing ontologies within

[27] http://www.yso.fi/onki/yso/app/annotation/integration-howto_en.html
[28] http://www.w3.org/TR/soap12-part1/
[29] http://dmoz.org/
[30] http://www.opencyc.org/

a particular application domain (e.g., SNOMED CT[31] in medicine and Getty Vocabularies[32] in cultural heritage). In contrast, FinnONTO tries to establish a comprehensive cross-domain ontology infrastructure, based on independent but mutually aligned ontologies. The experiment is conducted on a national level with a serious effort of actually starting to use ontologies in a larger scale.

The resulting ontology library is maintained as a centralized ONKI service[33] providing support for collaborative ontology publishing and ontology usage. A related approach is [35] presenting a distributed model for searching, reusing, and evolving ontologies. A novelty of the ONKI ontology services is that they can be used in external legacy and other applications as ready-to-use functionalities: we developed the idea of mash-up usage of ontologies in a way similar to Google Maps and other similar services. Another approach providing an integrable autocompletion widget for external systems is developed in [36].

We hope that the results of FinnONTO will encourage organizations to start developing machine (and human) "understandable" ontologies instead of thesauri targeted for human indexers, and that content providers will start using ontologies via ontology services such as ONKI. The FinnONTO framework is a practical approach to solving one of the most severe hindrances for the success of the Semantic Web: how to create good quality ontological metadata cost-efficiently. To test this, FinnONTO ontologies and ONKI ontology services will be maintained on a regular basis in a living lab environment for organizations to use in a national follow-up project of FinnONTO: Semantic Web 2.0 (FinnONTO 2.0, 2008–2010).

Our work is funded by the National Funding Agency for Technology and Innovation (Tekes) and a consortium of 38 companies and public organizations.

References

1. 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 (invited paper). In: Proceedings of the First International Semantic Computing Conference (IEEE ICSC 2007), Irvine, California, IEEE Press (2007) 216–223
2. Suominen, O., Viljanen, K., Hyvönen, E., Holi, M., Lindgren, P.: TerveSuomi.fi:n metatietomäärittely (Metadata schema for TerveSuomi.fi), Ver. 1.0 (26.1.2007). (2007) http://www.seco.tkk.fi/publications/.
3. Viljanen, K., Tuominen, J., Känsälä, T., Hyvönen, E.: Distributed semantic content creation and publication for cultural heritage legacy systems. In: Proceedings of the 2008 IEEE International Conference on Distibuted Human-Machine Systems, Athens, Greece, IEEE Press (2008)
4. 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 ESWC 2007, Innsbruck, Austria, Springer–Verlag (2007)

[31] http://www.ihtsdo.org/

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

[33] http://www.yso.fi/

5. Valkeapää, O., Hyvönen, E., Alm, O.: A framework for ontology-based adaptable content creation on the semantic web. J. of Universal Computer Science (2007)
6. Mäkelä, E., Hyvönen, E., Saarela, S., Viljanen, K.: OntoViews—a tool for creating semantic web portals. In: Proceedings of 3rd International Semantic Web Conference (ISWC 2004), Hiroshima, Japan, Springer–Verlag (2004)
7. 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 (2006)
8. 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), Springer–Verlag (2006)
9. Hyvönen, E., Mäkela, 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) 224–241
10. Hyvönen, E., Ruotsalo, T., Häggström, T., Salminen, M., Junnila, M., Virkkilä, M., Haaramo, M., Kauppinen, T., Mäkelä, E., Viljanen, K.: CultureSampo—Finnish culture on the semantic web. The vision and first results. In: Semantic Web at Work—Proceedings of STeP 2006, Espoo, Finland (2006) To appear in: Klaus Robering (Ed.), Information Technology for the Virtual Museum. LIT Verlag, 2008.
11. Ruotsalo, T., Hyvönen, E.: An event-based approach for semantic metadata interoperability. In: Proceedings of the ISWC 2007 + ASWC 2007, Busan, Korea, Springer–Verlag (2007)
12. Hyvönen, E., Viljanen, K., Suominen, O.: HealthFinland—Finnish health information on the semantic web. In: Proceedings of ISWC 2007 + ASWC 2007, Busan, Korea, Springer–Verlag (2007)
13. Suominen, O., Viljanen, K., Hyvönen, E.: User-centric faceted search for semantic portals. In: Proceedings of the ESWC 2007, Innsbruck, Austria, Springer–Verlag (2007)
14. 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). (2005)
15. 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.
16. Laukkanen, M., Viljanen, K., Apiola, M., Lindgren, P., Mäkelä, E., Saarela, S., Hyvönen, E.: Towards semantic web-based yellow page directory services. In: Proceeedings of the Third International Semantic Web Conference (ISWC2004), poster papers. (Nov 2004)
17. Komulainen, V., Valo, A., Hyvönen, E.: A tool for collaborative ontology development for the semantic web. In: Proceedings of International Conference on Dublin Core and Metadata Applications (DC 2005). (Nov 2005)
18. Foskett, D.J.: Thesaurus. In: Encyclopaedia of Library and Information Science, Volume 30. Marcel Dekker, New York (1980) 416–462
19. Aitchison, J., Gilchrist, A., Bawden, D.: Thesaurus construction and use: a practical manual. Europa Publications, London (2000)
20. Leskinen, R.L., ed.: Museoalan asiasanasto. Museovirasto, Helsinki, Finland (1997)
21. Hyvönen, E., Lindroos, R., Kauppinen, T., Henriksson, R.: An ontology service for geographical content. In: Poster Proc. of the ISWC + ASWC 2007, Busan, Korea. (2007)

22. Kauppinen, T., Hyvönen, E.: Modeling and reasoning about changes in ontology time series. In Kishore, R., Ramesh, R., Sharman, R., eds.: Ontologies in the Context of Information Systems, Springer–Verlag (2007)
23. Kurki, J., Hyvönen, E.: Relational semantic search: Searching social paths on the semantic web. In: Poster Proc. of the ISWC + ASWC 2007, Busan, Korea. (2007)
24. van Assem, M., Menken, M.R., Schreiber, G., Wielemaker, J., Wielinga, B.: A method for converting thesauri to RDF/OWL, Springer–Verlag (2004)
25. Gangemi, A., Guarino, N., Masolo, C., Oltramari, A., Schneider, L.: Sweetening ontologies with DOLCE. In: Proc. of the 13th International Conference on Knowledge Engineering and Knowledge Management. Ontologies and the Semantic Web, Springer–Verlag (2002)
26. Fellbaum, C., ed.: WordNet. An electronic lexical database. The MIT Press, Cambridge, Massachusetts (2001)
27. Guarino, N., Welty, C.: Evaluating ontological decisions with ONTOCLEAN. Comm. of the ACM **45**(2) (2001)
28. Antoniou, G., van Harmelen, F.: A semantic web primer. The MIT Press, Cambridge, Massachusetts (2004)
29. Hyvönen, E., Valo, A., Komulainen, V., Seppälä, K., Kauppinen, T., Ruotsalo, T., Salminen, M., Ylisalmi, A.: Finnish national ontologies for the semantic web—towards a content and service infrastructure. In: Proceedings of International Conference on Dublin Core and Metadata Applications (DC 2005). (2005)
30. Hameed, A., Preese, A., Sleeman, D.: Ontology reconciliation. In Staab, S., Studer, R., eds.: Handbook on ontologies. Springer–Verlag (2004) 231–250
31. Noy, N.F., Musen, M.A.: The PROMPT Suite: Interactive tools for ontology merging and mapping. International Journal on Digital Libraries **59**(6) (2003) 983–1024
32. Hyvönen, E., Lindroos, R., Kauppinen, T., Henriksson, R.: An ontology service for geographical content. In: Poster Proceedings of the 6th International Semantic Web Conference (ISWC + ASWC 2007), Busan, Korea, Springer–Verlag (Nov 2007)
33. Hyvönen, E., Salminen, M., Kettula, S., Junnila, M.: A content creation process for the Semantic Web (2004) Proceeding of OntoLex 2004: Ontologies and Lexical Resources in Distributed Environments, Lisbon, Portugal.
34. Hyvönen, E., Mäkelä, E.: Semantic autocompletion. In: Proceedings of the first Asia Semantic Web Conference (ASWC 2006), Beijing, Springer–Verlag (2006)
35. Maedche, A., Motik, B., Stojanomic, L., Studer, R., Volz, R.: An infrastructure for searching, reusing and evolving distributed ontologies. In: Proceedings of the 12th International Conference on World Wide Web (WWW 2003), Budapest, ACM (2003)
36. 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, Amsterdam (Nov 2007)