

Normalized Access to Ontology Repositories

Kim Viljanen, Jouni Tuominen, Eetu Mäkelä and Eero Hyvönen

Semantic Computing Research Group (SeCo)

Aalto University, School of Science, and University of Helsinki

<http://www.seco.tkk.fi/>, firstname.lastname@aalto.fi

Abstract—Ontology repositories, such as NCBO Bioportal, ONKI and Cupboard, help finding and using ontologies on the Semantic Web. However, currently each ontology repository constitutes a separate island with its own user interface, APIs, users, ontology languages and set of ontologies. Because there is not a universal way to access all ontology repositories, doing global search, browsing, and inference over all available ontology repositories turns out to be technically difficult and is generally not done. Ontologies are not reused as much as they could and hence the full potential of ontologies is not achieved. To address the problem, we propose the Normalized Ontology Repository (NOR) approach to make the ontology repositories universally accessible while maintaining their unique functionalities and strengths. The SKOS language is used as the lowest common denominator for presenting the ontologies. In addition, a simple API for searching and accessing the ontologies is defined. As a proof-of-concept evaluation, we present three case implementations to demonstrate the NOR approach: 1) the distributed architecture of the ONKI repository, 2) the metasearch for ONKI and NCBO Bioportal, and 3) publishing informal ontological concept collections as NOR end-points, demonstrated with the semantic portal CultureSampo and the metadata editor SAHA.

I. INTRODUCTION

Ontologies and ontology repositories have been considered to be a key resource for enabling the vision of the Semantic Web [1]–[3]. Ontology repositories are used for publishing, sharing and reusing ontologies and vocabularies for content indexing, information retrieval, content integration, and other purposes. Current implementations of ontology repositories include for example the NCBO BioPortal¹ [4], the Finnish Ontology Library Service ONKI² [5], Cupboard [6], the forthcoming Open Ontology Repository (OOR)³ [2], and there are many other systems, too [3].

An ontology is a shared specification of a conceptualization, defining concepts of a specific area of interest to allow sharing of knowledge [7]. Ontologies typically contain textual information about the concepts, relations between the concepts and, perhaps most importantly, define the unique identifiers (the URI in the context of the Semantic Web) of the concepts. With the help of the identifiers, the concepts can be referred to, for example, as values in metadata. Therefore, one typical use case for ontology repositories is to support the user in finding relevant ontological concepts from the underlying ontologies. With the help of concept search and browsing functionalities, the user can find the best matching concepts for her needs.

For example, if the user is creating metadata about an article about fishes, she could use an ontology repository containing an ontology about fishes to find out the correct URI for the concept “fish”, and then use this identifier as the value in her metadata.

In addition to formal ontologies, a vast amount of other kinds of concept collections of various degrees of formality exist that could be useful as identifiers for the Semantic Web. We call them informal ontologies. One example of such an informal ontology is Wikipedia, where the URL of each Wikipedia page correspond to a concept. For example the BBC is using the Wikipedia identifiers for interlinking content [8] which is based on RDF representation of the Wikipedia, the DBpedia [9]. Other examples of informal ontologies include registries maintained by libraries, such as books and people (e.g. ULAN⁴), and identifiers maintained by other organizations, such as locations (e.g. GeoNames⁵). In addition, many websites and their underlying content management systems use site specific categories and other types of concept collections that could be useful for others, too. For example, both ebay⁶ and Amazon⁷ have extensive product categorizations.

A problem of current ontology repositories is that each system constitutes a separate island with its own functionalities and its own set of ontologies [10]. Each system has its own user-interface, own API, and support different ontology languages. This limits the user from using efficiently different ontology repositories together because, for example, searching for a specific concept simultaneously from many repositories is not possible but requires visiting each repository separately. As a solution to the problem, we propose a universal access method to ontology repositories based on a normalized presentation of the ontology content and a shared API. We call this the Normalized Ontology Repository (NOR) approach. In addition to ontology repositories, we argue that it is relevant to consider non-ontological concept collections also as valuable sources for concept identifiers. Therefore, we suggest that the NOR approach could be used for publishing such non-ontological sources, too.

In the following, we first discuss why there is a need for a multitude of ontology repositories with different functionalities instead of just creating one application or web service to address all the needs. Then we present the NOR approach

¹<http://bioportal.bioontology.org/>

²<http://onki.fi>

³<http://ontolog.cim3.net/cgi-bin/wiki.pl?OpenOntologyRepository>

⁴<http://www.getty.edu/research/tools/vocabularies/ulan/>

⁵<http://www.geonames.org>

⁶<http://www.ebay.com>

⁷<http://www.amazon.com>

for creating universal access to different ontology repositories. After this, three implementations of the NOR approach are described. Finally, related work is presented, the results of this paper are summarized and discussed.

II. MOTIVATION

To motivate our work, we discuss the following questions: Is there a need for different ontology repositories? Would simultaneous access to ontology repositories be useful? Do existing Semantic Web technologies and practices address the needs of ontology repositories?

A. One Size Does Not Fit All

Instead of using a multitude of different implementations for ontology repositories, one could argue that a single implementation could address all the different needs of the users, and the needs created by the different types of ontologies and application domains. In addition, there could be a single global ontology repository that would contain all ontologies. While both scenarios could theoretically be possible, we argue that there are a multitude of challenges in such approach due to the following reasons:

Different ontologies and user needs require different functionalities. For example, the ONKI ontology repository supports different types of ontologies and implements different visualizations, such as a geographical map interface for geographical ontologies and a tree visualization for concept hierarchies, as depicted in Fig. 1. This is done to address the different needs of different ontologies such as general (abstract) ontologies versus geographical ontologies. For example, the BioPortal has been designed originally to address the needs of the biomedical domain. Also different ontology languages require different technical implementations to maximize the benefits of the given formalism.

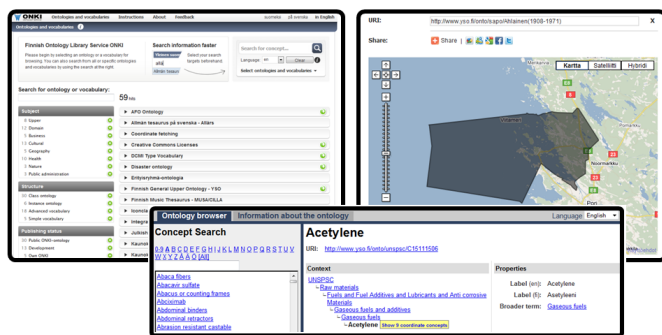


Fig. 1. Some of the user interfaces of the ONKI repository, including ontology listing, map visualization for geographical ontologies, and concept hierarchy visualization.

Some ontologies are not available as files but only as services. Typically, ontologies are published as files that can be uploaded to ontology repositories. For example, if the ontology changes constantly or if the size of the ontology is substantial, publishing the content as a file may not be

practical. In addition, due to business reasons, some ontologies are not published as files but only as a service. In these cases, the ontology is only available via the specific API or a user-interface, but can not be uploaded to a shared ontology repository.

Security or business reasons may require using internal ontology repositories. For example, security reasons of an organization may require that selected ontologies are available only for internal use or that the server logs of who checked what ontological concepts remain confidential. Such requirements can be addressed with private, internal ontology repositories that are fully controlled by the organization itself.

All concept collections are not ontology repositories. As we pointed out in the introduction, a vast amount of informal ontological concept collections exists that are useful as identifiers for the Semantic Web, such as the Wikipedia, the various registries maintained by libraries and categorizations located in various websites, such as ebay and Amazon. A single ontology repository system most probably will not replace all of the different systems that are used for maintaining such informal ontologies and vocabularies of various degrees of formality.

B. Simultaneous Access to Repositories

The typical way to use ontology repositories and ontologies in applications is that the application developers choose a specific ontology repository and specific ontologies in advance which are then used in the application. The end-user does not have to bother what ontology repository or ontology to choose when using the application. We argue that the practice of using only one repository leads up to the following problems.

The optimal concept or ontology might not be found. Because searching simultaneously ontology repositories is not possible, the end-user or the application developer might not notice that there exists a suitable ontology or ontological concept for one's needs. Due to this, the quality of ontology-enabled applications and metadata may decrease when the perfect concepts are not found and used. Also redundant new ontologies and concepts may be created if ontology developers do not find existing ontologies, which makes it more difficult to maintain the interoperability of data, because data originating from different sources is described using a different (redundant) ontology, which means that the ontologies must first be interlinked before they can be used for interlinking the underlying data.

High quality ontologies might be underused. The more a specific ontology is used, the more established it is to act as the de facto standard for representing concepts and metadata of its specific domain. This also increases the ontology developers and publishers benefits for creating and maintaining the ontology. If a specific ontology repository is not found by the potential users, then also the high quality ontologies contained in the repository will be underused, and the benefits from creating the ontology decreases. In addition, as pointed out before, informal ontological concept collections might not be considered as valuable sources for concept identifiers such as the Wikipedia or subject heading thesauri maintained

by libraries because they are not published in an existing ontology repository or made available using standard Semantic Web formats. Publishing them as ontology repositories would increase the benefits of existing work.

Ontologies are not interlinked between repositories. The ontology repositories are currently not acting as model citizens of the Semantic Web, since their ontological content is not interlinked between repositories. That is, the ontologies and the ontology repositories do not implement and follow the Linked Data [11] practices at the moment. For example, (automatic) linking to relevant concepts in other ontologies could help the users to find the best ontologies and concepts for each need.

Publishing same ontologies in many repositories creates challenges. Some ontologies may be published in many Ontology Repositories because they have been considered to be useful by the repository publisher, or because they have been uploaded by the ontology developer to as many repositories as possible to reach as many users as possible. Maintaining the same ontology in many repositories may lead to redundant work because new versions of the ontology has to be updated in all repositories. Also, some repositories may contain older versions of the ontology which creates compatibility problems when using metadata based on the ontologies.

Internal ontology repositories require maintenance. Internal, potentially confidential, ontology repositories may contain public ontologies. Maintaining the public ontologies to the latest versions require additional work. To avoid this, simultaneously access to both internal and public ontology repositories would be beneficial.

C. Shortcomings of Existing Technologies

General Semantic Web search engines, such as Swoogle⁸ [12] and Sindice⁹ [13] are not focused on ontologies but provide general search of all kind of RDF data. Ontology search engines, such as Falcons¹⁰ [14] and Ontosearch2 [15] address ontology specific needs, but do not address the problem of accessing informal ontology repositories. In addition, all of the previously mentioned search engines are based on crawling the ontology sources, which means, that they are may not always be up-to-date. In addition, ontologies that are only available as services, via an API or user-interface, may not be indexed.

Ontologies are represented using various languages, such as the Semantic Web languages RDFS, OWL and SKOS, Common Logic, Excel, HTML, database tables, and application specific languages. A shared practice is missing on how to publish ontologies on the Semantic Web [2].

SPARQL¹¹ is the standard way to provide an application interface to Semantic Web databases, and it can be used also to access ontology repositories. However, implementing a SPARQL end-point can be difficult if the underlying system

is not based on Semantic Web technologies. Making SPARQL queries require also advance knowledge on what ontology language has been used to be able to make a matching query and to interpret the result.

To conclude our analysis, in the foreseeable future there will be many different ontology repositories. Accessing simultaneously these repositories would be useful and is not solved in an optimal way with current technologies.

III. THE NORMALIZED ONTOLOGY REPOSITORIES

As a solution to the problems presented above, we propose the Normalized Ontology Repository (NOR) approach. NOR consist of 1) a normalized presentation for ontology concepts, making thus the different ontology language schemas interoperable, and 2) a simple API for accessing the ontology repository.

A. Normalized Representation of Ontological Concepts

Ontologies are presented using different ontology languages, such as OWL, RDFS and SKOS, and there exists many informal ontologies, too. From the interoperability point of view, this creates a problem, because each ontology language must handled as a separate case. In the worst case, an application developer have to handle ontologies presented in many different languages to build an application that utilizes ontologies. Due to this, for example, the ONKI repository has a rule-based configuration language to adjust the system to support ontologies represented in various kinds of RDF based languages.

To avoid complicated mappings and inference of hierarchical and other relations, we propose that each ontology repository should provide a normalized, dumbed down presentation of the ontology concepts in addition to the native format of the ontology. As the normalization language we suggest using the RDF based Simple Knowledge Organization System (SKOS)¹², which is a RDF based language for presenting thesauri, classification schemes, subject heading systems and taxonomies within the framework of the Semantic Web. SKOS is by design intended to serve as a common denominator between different modeling approaches and therefore we decided to use it compared to other alternatives, such as OWL or RDFS.

Hiding ontological details makes it easier for the applications using the NOR compatible ontology repositories. After finding an interesting concept, the user can be directed to the specific Ontology Repository with its full functionality for using the specific ontology. Our intention is to make it easier to access the basic information of ontological concepts in an unified way, not to restrict the user from using the original, full-blown ontology languages and functionalities of the underlying ontology repositories for specific needs.

In practice, a NOR compatible ontology repository must provide a concept lookup method:

- *concept?uri=[concept identifier]*

⁸<http://swoogle.umbc.edu>

⁹<http://sindice.com>

¹⁰<http://ws.nju.edu.cn/falcons>

¹¹<http://www.w3.org/TR/rdf-sparql-query/>

¹²<http://www.w3.org/2004/02/skos/>

which returns the normalized SKOS version of the given concept, identified by the concept URI. For example, to get the normalized concept representation of *yso:p907* from the ONKI ontology repository, the lookup request URL is:

```
http://onki.fi/nor/concept?uri=http%3A%2F%2Fwww.yso.fi%2Fonto%2Fyso%2Fp907
```

which returns the following SKOS representation¹³ of the given concept followed by the (optional) native representation:

```
# Namespace declarations omitted
# Normalized SKOS representation begins
<http://onki.fi/nor/concept?uri=http%3A%2F%2Fwww.yso.fi%2Fonto%2Fyso%2Fp907>
  a skos:Concept;
  skos:prefLabel "fish"@en, "kala"@fi;
  skos:broader
    <http://onki.fi/nor/concept?uri=http%3A%2F%2Fwww.yso.fi%2Fonto%2Fyso%2Fp6580>;
#additional properties omitted
#link to the native concept format
  nor:describes yso:p907
.
# Native representation begins (optional)
yso:p907
  a yso:Concept;
  yso:prefLabel "fish"@en, "kala"@fi;
  #...additional properties omitted
.
```

The SKOS presentation above describes key information about the given concept (*yso:p907*) such as the labels (in English “fish”, in Finnish “kala”), and the URL to the normalized broader concept *yso:p6580* (foods). In addition, the native representation *yso:p907* is also presented as part of the normalized concept lookup response.

To avoid cluttering the native presentation by adding additional RDF triplets to it, the native and normalized formats are kept apart from each other with the following RDF property¹⁴:

- *nor:describes*

The property is used for referring to the native concept presentation from the normalized SKOS representation. To avoid making unintended conclusions, we did not use, for example, the *owl:sameAs* property which would have meant that the normalized and the native presentations would refer to the same thing, which may not be true.

Finally, in some cases the ontology repository publisher may have decided to use SKOS as the native representation for the concepts. If so, the *nor:describes* relation and the native representation can be omitted.

B. Concept Search

To make searches to a NOR compatible ontology repository we define the following method:

- *search?q=[query]&l=[language]*

The search method is used for finding concepts matching the given query string and language. Currently, the query string can only contain a keyword, but in future the query language may be extended. The method returns a list of matching

concepts presented using a JSON based response format. Other result languages and formats may be considered in the future, but we deemed this representation to be simpler than, for example, representing the same information as an ordered list in RDF.

For example, a search for “fish” to the ONKI ontology repository is done with the following URL:

```
http://onki.fi/nor/search?q=fish&l=en
```

The system responds with the following result:

```
{ "results" : [
  { "concept-label" : "fish",
    "concept-label-language" : "en",
    "concept" :
      "http://www.yso.fi/onto/yso/p907",
    "normalized-concept" :
      "http://onki.fi/nor/concept?uri=http%3A%2F%2Fwww.yso.fi%2Fonto%2Fyso%2Fp907",
    "native-concept" :
      "http://www.yso.fi/onto/yso/p907",
    "ontology-abbreviation" : "yso",
    "ontology-label" : "Finnish General Upper
      Ontology",
    "ontology-label-language" : "en",
    "ontology-uri" :
      "http://www.yso.fi/onto/yso"
  }
  ...
],
  "metadata" : { "containingHitsAmount" : 50,
                 "moreHitsAmount" : 1467 }
}
```

In the result, *concept* is the URI of the concept, *normalized-concept* is the URL of the normalized representation of the given concept, and *native-concept* is the URL to the native representation of the concept.

C. Ontology Repository Metadata

To find NOR compatible ontology repositories, a list of repositories that conform to the NOR principles would be helpful. However, to avoid the problems of centralized systems, we do not require ontology repositories to publish information about themselves to any specific registries.

To help finding suitable repositories and ontologies for one’s need, we suggest that the NOR ontology repositories publish metadata about the available ontologies using the following method:

- *ontologies*

which returns metadata about the ontologies in the repository and the NOR end-point URL of each ontology. The metadata of the ontologies can be represented using, for example, the Vocabulary of Interlinked Datasets (void)¹⁵ metadata language. Additional information about the ontology, such as the title and description, may be expressed using e.g. the Dublin Core metadata schema, the Ontology Metadata Vocabulary (OMV)¹⁶, and the upcoming Catalogue Vocabulary (dcat)¹⁷.

¹³presented using the RDF Turtle syntax

¹⁴nor namespace: <http://purl.org/finnonto/schema/nor>

¹⁵<http://rdfs.org/ns/void#>

¹⁶<http://omv2.sourceforge.net>

¹⁷http://www.w3.org/egov/wiki/Data_Catalog_Vocabulary

To express the URL of the NOR end-point for a given ontology, we define the following RDF property:

- *nor:endpoint*

For example, in the case of the ONKI ontology repository, the ontology metadata is available at:

`http://onki.fi/nor/ontologies`

which returns following metadata (excerpt):

```
# Namespace declarations omitted
<http://www.yso.fi/onto/yso>
  a void:Dataset ;
  dc:title "Finnish General
  Upper Ontology"@en ;
  dc:creator
    <http://www.yso.fi/onki-ns/onki/Finnonto> ;
  dc:license <http://creativecommons.org/
  licenses/by/3.0/> ;
  foaf:homepage
    <http://www.seco.tkk.fi/ontologies/yso>,
  nor:endpoint <http://onki.fi/nor/> .
...
```

The metadata can be used for creating for example a catalogue of NOR compatible ontology repositories and concept collections. In addition, this could be used for implementing a metasearch service to search simultaneously multiple underlying NOR compatible ontology repositories.

D. General Remarks on the API

We argue, that a simple HTTP API is easy to implement both for the ontology repository developers and for the developers that want to access the NOR compatible ontology repositories. In addition, a simple API is easy to implement even if the underlying ontology repository is not based on RDF but is, for example, a relational database of people, which could be highly relevant to publish as a NOR endpoint. Thus, compared to e.g. using the RDF query language SPARQL, the simple API approach makes it easier for both publishers and users to benefit of the NOR network.

This does not limit, however, the underlying ontology repositories from implementing in addition, for example, a SPARQL end-point. A key idea behind NOR is that the native functionalities of the underlying ontology repositories are available for users that need more functionalities than what the simple NOR API and normalized presentation can provide.

The proposed API is summarized in the Table I.

TABLE I
THE NORMALIZED ONTOLOGY REPOSITORY API.

Method name	Parameters	Return value
concept	concept identifier	normalized concept representation
search	query string, language	matching concepts
ontologies	-	ontology metadata

IV. CASE STUDIES AND EVALUATION

To analyze the idea of NOR, we have implemented three proof-of-concept prototypes which will be presented and discussed in the following.

The NOR approach generalizes and unifies experiences gained from our work on the ONKI repository and the ONKI API. Therefore, a majority of the following case studies are based on ONKI, viewed from the NOR perspective. The functionalities of the NOR API is a subset of the ONKI API's functionalities, however, a key difference is that the ONKI API represents the SKOS description of a concept in an ONKI specific JSON format to avoid the overhead of parsing RDF in the ONKI frontend, but in the NOR API we propose using RDF to represent this information.

A. NOR as an Internal Architecture

The ONKI SKOS ontology server [16] has been used for publishing over 70 ontologies in the Finnish Ontology Library Service ONKI [5], which has been running as a pilot service from September 2008. The system is in living lab use with ca. 10 000 unique human visitors monthly¹⁸, and there are over 300 registered users of the APIs and widgets. Even though ONKI SKOS supports especially vocabularies presented in SKOS, the server can be used for publishing ontologies presented in RDF(S) and restricted OWL. To access the multitude of ONKI SKOS servers, the ONKI system implements a front-end service for making metasearches to the ONKI SKOS and other ONKI back-ends using a shared HTTP API (see Fig. 2). The back-ends and their respective ontologies are described with metadata to enable the front-end to locate the available ontologies and to display information about the ontologies to the users.

Searching for concepts using an ONKI backend server is done with its HTTP API method *search*, which returns concepts matching to the query string in a specified language. The *getFullPresentation* method returns all information about a given concept, such as the preferred and alternative labels, the transitive parent concept tree, and the related concepts. Independently of the language each ontology is presented in, each concept is always returned in a uniform SKOS inspired JSON format which describe the normalized basic information of the given concept.

Building on this underlying distributed architecture, three clients have been designed and implemented. The ONKI3 Browser¹⁹ is a metasearch and browsing user interface for accessing the ONKI SKOS and other back-end servers. For example, making a global query to all ontology servers can be done. Also, a directory listing of the ontologies in the ONKI Ontology Repository is provided based on the metadata about the published ontologies. The ONKI3 user interface was mostly implemented using PHP²⁰.

Another client is the JavaScript-based ONKI Selector widget [17] for adding ontological concept search to HTML forms. The third client is a simple URI resolver for dereferencing the end-user's ontology concept URI requests to a suitable representation provided via the ontology repository network, such as HTML or RDF.

¹⁸Measured with Google Analytics.

¹⁹<http://onki.fi/en/browser>

²⁰<http://www.php.net/>

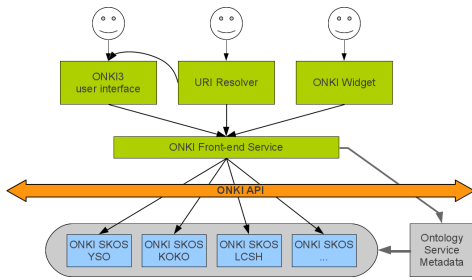


Fig. 2. The ONKI architecture is based on a distributed metasearch approach.

The loosely coupled ONKI architecture has turned out to be a flexible and modularized approach for implementing an ontology repository consisting of multiple back-end ontology servers. The normalized representation of the underlying ontology repositories have made it easy to implement a user-interface for accessing all underlying repositories. Making multiple HTTP requests to back-end servers may be slow in the worst case, but in our test implementation this lag has not been a problem.

B. Searching Simultaneously BioPortal and ONKI

To test the NOR approach in a distributed setting of multiple independent ontology repositories, we implemented a proof-of-concept metasearch prototype to search simultaneously the ONKI SKOS [16] servers described above and the NCBO BioPortal [4]. The NCBO BioPortal is an open repository of biomedical ontologies and it has been used for publishing over 200 ontologies [4]. BioPortal provides functionalities, such as concept and ontology search and browsing, peer reviewing of the ontologies, and support for creating and viewing mappings between ontologies.

The ONKI-BioPortal metasearch prototype allows the user to find the relevant concepts from the participating ontology repositories, without having to know in advance which repository to make the search to.

Since the ONKI front-end [5] was already designed using a metasearch approach, the ONKI-BioPortal prototype was implemented by creating a wrapper for BioPortal which implements the ONKI API's search and concept lookup (*getFullPresentation*) methods. When calling the wrapper, it makes requests to BioPortal, parses BioPortal's XML messages, and transforms them to the ONKI JSON format. Since the BioPortal API does not contain a concept lookup method that would return all information about the specific concept with a single request, multiple HTTP REST requests have to be made to get all the needed information about a concept.

Fig. 3 presents the ONKI-BioPortal search prototype user-interface displaying the result for a metasearch query for "fish product". The result consists of 22 hits which are found from the BioPortal and the ONKI SKOS back-ends. The hits originating from BioPortal are labeled as "BioPortal" for demonstrating purposes, but in actual use, instead of "BioPortal" the name of the ontology should be displayed.

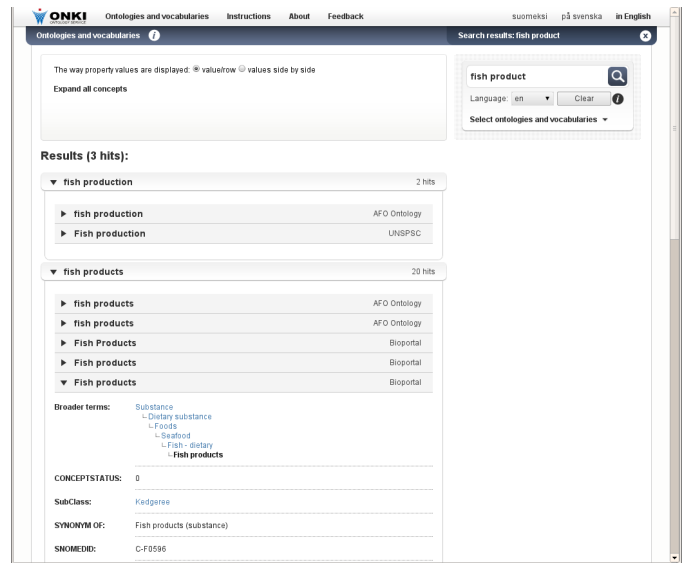


Fig. 3. The user-interface of the ONKI-BioPortal metasearch prototype.

C. NOR for Informal Ontological Concept Collections

Besides Ontology Repositories, applications often need to refer also to informal ontological concept collections, such as authority or place databases. However, the functionalities required for such data sources are usually very similar to those required for ontology repositories. For example, in an editor environment, similar semantic autocompletion search functionalities are used for both ontological and non-ontological concept collections, along with the same functionality for describing and visualizing the possible choices returned from such a search. Informal ontological concept collections also often change more rapidly than their ontological counterparts, so it makes even more sense to access the original system through programmatic APIs than exporting and publishing the data in a ontology repository. In order to test how the NOR approach fared in the context of such informal ontological concept collections, the ONKI API was implemented in two applications: the semantic portal CultureSampo [18] and the SAHA metadata editor [19]. Both are Semantic Web applications, but their focus is not on ontologies but to display and edit all kinds of semantic data.

For CultureSampo, the ONKI API was actually implemented to benefit those using SAHA to edit data. This was because the CultureSampo database contains, for example, a large number of places, people and organizations that are useful to people indexing new content. For added freedom, the CultureSampo ONKI API was parametrized, so that the types of objects that search operations return can be specified dynamically. This way, one can say for example that they want an autocompletion facility of all the organizations, all the places, or all the historical events in CultureSampo.

While SAHA was already a client to the ONKI API of the ONKI Ontology Repository and CultureSampo, the API was

implemented also into SAHA itself. This was done to make possible the creation of a network of dynamically updated, collaboratively curated concept collections. The multiple projects using the SAHA editor to index content often need to add new places, organizations or people to their list of reference values. However, until now, these have all resided in the private data spaces of the different projects using SAHA. Now, the intention is to move these created concepts into SAHA projects of their own, so that one SAHA project will hold collaboratively curated place database, while another contains a database of organizations and people. These can then be linked through the ONKI APIs to each other, as well as to the primary indexing projects. In this way, the various projects can start to directly benefit each other.

V. RELATED WORK

The work is partially based on our previous work on the national ontology library ONKI [5], [20] and is related to the open ontology repository (OOR)²¹ initiative which aims at developing an interoperability infrastructure for ontologies [2].

Compared to more general methods of accessing RDF data, such as SPARQL²² and Linked Data [11], the NOR approach focuses on ontologies. For example, when searching for concepts with the NOR API, one does not need to know what RDF properties are used in the data to express the labels. In addition, the ontology repositories can be optimized to respond quickly to specific API queries. A normalized presentation of ontological concepts (SKOS) could, however, also be beneficial for querying the data via SPARQL, and browsing the ontology repositories as linked data. For example, one does not have to know which specific hierarchical relation (e.g. *rdfs:subClassOf* or *skos:broader*) has been used, because the normalized hierarchical relation is constant.

APIs for accessing ontologies and vocabularies published by other authors previously include the SKOS API²³ and the OWL API²⁴. Compared to them, the NOR approach provides a higher abstraction, independent from specific ontology languages, and a lightweight and simple API. Compared to the APIs of BioPortal [4], Swoogle²⁵ [12], Watson²⁶ [21], ONKI SKOS and others, the NOR API focuses on a few basic methods that reflects the basic functionality of ontology repositories, e.g. concept search.

The Ontosearch2 [15] does a automatic complexity reduction of ontologies to ensure answering the ontology search queries within a specific time limit. This automatic approach however require using the OWL ontology language which is a limitation since many ontologies are not presented in that specific language. In contrast, the NOR approach is based on defining the normalized language and the simple API with

the goal of publishing both ontology repositories and informal ontological concept collections as uniform services.

Ontology Repositories such as BioPortal and Cupboard support publishing interlinked ontologies, but the ontologies have to be uploaded into a centralized service for a global search. On the other hand, the OOR [2] initiative intends to design an Ontology Repository framework that addresses the needs of all users, and includes an inter-repository content change protocol to keep the different OOR repositories up to date. In contrast to these, the NOR approach does not restrict the ontology publishers in where to publish the ontologies or what software to use. Instead, the ontologies can be published using an ontology service that is optimized for the specific ontology and the user's needs. If the organization wants to promote and make their ontologies available to the NOR users, they can implement the NOR API to make their repository compatible with other NOR repositories. If needed, the NOR API of a repository can be restricted to selected users or made publicly available for anybody.

Compared to the Open Knowledge Base Connectivity (OKBC) specification²⁷ and the agent communications languages FIPA-ACL²⁸ and KQML²⁹, which all can also be used to access ontological information, the NOR approach is more focused on the specific use-cases of finding ontologies and ontology concepts, and to get relevant information about them.

The OntoCAT is a programming interface to query multiple ontology repositories seamlessly from an application [10]. A wrapper is implemented for each supported ontology repository, such as the NCBO BioPortal. In comparison, to avoid wrappers, the NOR approach is based on defining a shared, unified representation for the ontology repositories. A well-known limitation of wrappers is that changes in the underlying representation often breaks the wrapper.

VI. DISCUSSION

This paper argues that ontology repositories should be made accessible using a shared API that would provide a simple but universal methods for accessing the repositories in a uniform way. In addition, the ontologies should be presented using a normalized concept representation.

The NOR approach has been evaluated with three case studies: The ONKI ontology repository case study demonstrates using the NOR approach for building an ontology service consisting of over 70 underlying back-ends with over 10 000 unique monthly users. The NCBO BioPortal and ONKI case study demonstrates using the NOR approach for creating a global search and browsing user-interface for accessing independent distributed ontology repositories. Finally, the SAHA metadata editor and the CultureSampo semantic portal case study demonstrates that the NOR approach can be used for accessing non-ontological concept collections.

The outcome of this work is that the NOR approach is feasible for providing a unified access to a multitude of

²¹<http://ontolog.cim3.net/cgi-bin/wiki.pl?OpenOntologyRepository>

²²<http://www.w3.org/TR/rdf-sparql-query/>

²³<http://www.w3.org/2001/sw/Europe/reports/thes/skosapi.html>

²⁴<http://owlapi.sourceforge.net/>

²⁵<http://swoogle.umbc.edu/>

²⁶<http://watson.kmi.open.ac.uk/>

²⁷<http://www.ai.sri.com/okbc/spec.html>

²⁸<http://www.fipa.org/repository/aclspecs.html>

²⁹<http://www.cs.umbc.edu/csee/research/kqml/>

ontology repositories. This makes it possible to provide for example global search and global browsing functionalities to a collection of separate underlying ontology repositories. At the same time, the NOR does not restrict the individual ontology repository providers from creating advanced ontology, business, and user specific implementations because the relation between the normalized representation and the native representation is kept intact.

The NOR approach allows the ontology user to find relevant concepts and ontology repositories in cases where the correct ontology repository is not known in advance or when many ontology repositories are used simultaneously. After finding the relevant repository, the user may access the underlying ontology repository for full-blown functionalities. For organizations that maintain an internal ontology repository, the NOR approach makes it possible to make simultaneous queries to repositories outside the organization. For the ontology publishers, implementing the NOR API increases the findability of the ontologies and therefore the benefits of publishing the ontology in the first place.

Future work includes developing further the API and its methods to support, for example, restricting queries to a specific ontology, specific subpart of the ontology or to a specific concept type. The normalized concept representations could be improved by introducing links between ontologies in the spirit of Linked Data. Such mappings between ontologies could be produced potentially automatically by creating a matching application on the top of the NOR compatible ontology repositories. NOR based metasearch would benefit from a ranking algorithm for ordering the results originating from different underlying ontology repositories. Finally, to evaluate the full potential of the approach, formal and informal ontology repositories should implement the NOR API.

Acknowledgements: This work is part of the National Semantic Web Ontology project in Finland³⁰ (FinnONTO, 2003-2012), funded mainly by the National Technology and Innovation Agency (Tekes) and a consortium of 38 organizations. We thank Osma Suominen, the Semantic Computing Research Group, and the OOR network for fruitful discussions.

REFERENCES

- [1] E. Hyvönen, K. Viljanen, J. Tuominen, and K. Seppälä, "Building a national semantic web ontology and ontology service infrastructure—the FinnONTO approach," in *Proceedings of the ESWC 2008, Tenerife, Spain*. Springer-Verlag, 2008.
- [2] K. Baclawski and T. Schneider, "The open ontology repository initiative: Requirements and research challenges," in *Proceedings of Workshop on Collaborative Construction, Management and Linking of Structured Knowledge at the ISWC 2009*, Washington DC., USA, October 2009.
- [3] M. d'Aquin and N. F. Noy, "Where to publish and find ontologies? A survey of ontology libraries," *Web Semantics: Science, Services and Agents on the World Wide Web*, vol. 11, pp. 96–111, Mar. 2012. [Online]. Available: <http://dx.doi.org/10.1016/j.websem.2011.08.005>
- [4] N. F. Noy, N. H. Shah, P. L. Whetzel, B. Dai, M. Dorf, N. Griffith, C. Jonquet, D. L. Rubin, M.-A. Storey, C. G. Chute, and M. A. Musen, "BioPortal: ontologies and integrated data resources at the click of a mouse," *Nucleic Acids Research*, vol. 37, no. Web Server issue, pp. 170–173, 2009.
- [5] K. Viljanen, J. Tuominen, and E. Hyvönen, "Ontology libraries for production use: The Finnish ontology library service ONKI," in *Proceedings of the ESWC 2009, Heraklion, Greece*. Springer-Verlag, 2009.
- [6] M. d'Aquin and H. Lewen, "Cupboard - a place to expose your ontologies to applications and the community," in *Proceedings of the ESWC 2009*. Heraklion, Greece: Springer-Verlag, June 2009, pp. 913–918.
- [7] T. R. Gruber, "A translation approach to portable ontology specifications," *Knowledge Acquisition*, vol. 5, no. 2, pp. 199–220, 1993.
- [8] G. Kobilarov, T. Scott, Y. Raimond, S. Oliver, C. Sizemore, M. Smethurst, C. Bizer, and R. Lee, "Media meets semantic web – how the bbc uses dbpedia and linked data to make connections," in *Proceedings of the ESWC 2009, Heraklion, Greece*. Springer-Verlag, 2009.
- [9] C. Bizer, J. Lehmann, G. Kobilarov, S. Auer, C. Becker, R. Cyganiak, and S. Hellmann, "Dbpedia a crystallization point for the web of data," *Journal of Web Semantics: Science, Services and Agents on the World Wide Web*, no. 7, p. 154165, 2009.
- [10] T. Adamusiak, T. Burdett, N. Kurbatova, K. J. van der Velde, N. Abeygunawardena, D. Antonakaki, M. Kapushesky, H. Parkinson, and M. Swertz, "Ontocat – simple ontology search and integration in java, r and rest/javascript," *BMC Bioinformatics*, vol. 12, no. 1, p. 218, 2011. [Online]. Available: <http://www.biomedcentral.com/1471-2105/12/218>
- [11] C. Bizer, R. Cyganiak, and T. Heath, "How to publish linked data on the web," <http://www4.wiwiw.fu-berlin.de/bizer/pub/LinkedDataTutorial/>, July 27 2007.
- [12] L. Ding, T. Finin, A. Joshi, R. Pan, R. S. Cost, Y. Peng, P. Reddivari, V. C. Doshi, and J. Sachs, "Swoogle: A Search and Metadata Engine for the Semantic Web," in *Proceedings of the Thirteenth ACM Conference on Information and Knowledge Management*. ACM Press, November 2004.
- [13] E. Oren, R. Delbru, M. Catasta, R. Cyganiak, H. Stenzhorn, and G. Tummarello, "Sindice.com: a document-oriented lookup index for open linked data." *IJMSO*, vol. 3, no. 1, pp. 37–52, 2008. [Online]. Available: <http://dblp.uni-trier.de/db/journals/ijms/o/ijms03.html>
- [14] Y. Qu and G. Cheng, "Falcons concept search: A practical search engine for web ontologies," *IEEE Transactions on Systems, Man, and Cybernetics, Part A*, vol. 41, no. 4, pp. 810–816, 2011.
- [15] E. Thomas, J. Z. Pan, and D. Sleeman, "ONTOSEARCH2: Searching Ontologies Semantically," in *Proceedings of the OWLED 2007 Workshop on OWL: Experiences and Directions*, 2007, online publication: <http://sunsite.informatik.rwth-aachen.de/Publications/CEUR-WS/Vol-258/>.
- [16] J. Tuominen, M. Frosterus, K. Viljanen, and E. Hyvönen, "ONKI SKOS server for publishing and utilizing SKOS vocabularies and ontologies as services," in *Proceedings of the ESWC 2009, Heraklion, Greece*. Springer-Verlag, 2009.
- [17] K. Viljanen, J. Tuominen, and E. Hyvönen, "Publishing and using ontologies as mash-up services," in *Proceedings of the 4th Workshop on Scripting for the Semantic Web (SFSW2008), 5th European Semantic Web Conference 2008 (ESWC 2008)*, June 1-5 2008.
- [18] E. Hyvönen, E. Mäkelä, T. Kauppinen, O. Alm, J. Kurki, T. Ruotsalo, K. Seppälä, J. Takala, K. Puputti, H. Kuittinen, K. Viljanen, J. Tuominen, T. Palonen, M. Frosterus, R. Sinkkilä, P. Paakkari, J. Laitio, and K. Nyberg, "CultureSampo – Finnish culture on the semantic web 2.0. Thematic perspectives for the end-user," in *Proceedings, Museums and the Web 2009, Indianapolis, USA*, April 15-18 2009.
- [19] J. Kurki and E. Hyvönen, "Collaborative metadata editor integrated with ontology services and faceted portals," in *Workshop on Ontology Repositories and Editors for the Semantic Web (ORES 2010), the Extended Semantic Web Conference ESWC 2010, Heraklion, Greece*. CEUR Workshop Proceedings, <http://ceur-ws.org/>, June 2010.
- [20] K. Viljanen, J. Tuominen, M. Salonoja, and E. Hyvönen, "Linked open ontology services," in *Workshop on Ontology Repositories and Editors for the Semantic Web (ORES 2010), the Extended Semantic Web Conference ESWC 2010*. CEUR Workshop Proceedings, <http://ceur-ws.org/>, June 2010.
- [21] M. d'Aquin, M. Sabou, E. Motta, S. Angeletou, L. Gridinoc, V. Lopez, and F. Zablith, "What can be done with the semantic web? an overview of watson-based applications," in *5th Workshop on Semantic Web Applications and Perspectives, SWAP 2008*, 2008.

³⁰<http://www.seco.tkk.fi/projects/finnonto/>