A Browser-based Tool for Collaborative Distributed Annotation for the Semantic Web

Onni Valkeapää
Helsinki University of Technology (TKK)
Semantic Computing Research Group (SeCo)
P.O. Box 5500, Otaniementie 17
FI-02015 TKK
http://www.seco.tkk.fi/
onni.valkeapaa@tkk.fi

Eero Hyvönen
Helsinki University of Technology (TKK) and
University of Helsinki
Semantic Computing Research Group (SeCo)
P.O. Box 5500, Otaniementie 17
FI-02015 TKK
http://www.seco.tkk.fi/
eero.hyvonen@tkk.fi

ABSTRACT
This paper presents a prototype of an ontology-based semantic annotation tool Saha. The tool eases the process of creating ontological descriptions of documents by providing a simple user interface that hides the complexity of ontologies from annotators. Saha is used with a web browser, and it supports collaborative distributed creation of metadata by centrally storing annotations, which can be viewed and edited by different annotators. Concepts defined in external ontologies can be imported and used in annotations by connecting Saha to ontology servers. The tool is being tested in practical semantic portal projects.

Categories and Subject Descriptors
H.3.1 [**Information storage and retrieval**]: Content Analysis and Indexing – *Indexing methods.*

General Terms
Design, Experimentation.

Keywords
Semantic Annotation, Ontologies, Annotation Schema.

1. INTRODUCTION
Provision of semantically rich, ontology-based metadata is one of the major challenges in developing the Semantic Web. In recent years, various annotation systems have been developed to face this challenge [14]. There is, however, a lack of systems that 1) can be easily used by annotators unfamiliar with technical side of the Semantic Web, and that 2) are able to support distributed creation of semantic metadata based on complex metadata annotation schemas (ontologies). In this paper, we present an annotation tool, Saha[1] [15], aiming to satisfy these needs. Saha is browser-based in order to support wide and distributed usage. It has simple user interface that hides complexity of ontologies from the annotator, and adapts automatically to different metadata schemas. Saha supports collaborative annotation of web-documents and it can utilize ontology services for sharing URIs and importing concepts defined in various external ontologies. The tool is targeted especially for creating metadata of web resources in semantic web portals. It is being applied to various applications within the National Semantic Web Ontology Project in Finland (FinnONTO)[2].

2. SAHA ANNOTATION SYSTEM

2.1 Design rationale and implementation
In order to develop an annotation system that would be easy to use and that would support the creation of semantically rich metadata, we identified four basic requirements for our system. These were also features that we felt were not supported well enough in many of the current annotation platforms:

1) The system should, as a rule, hide technical concepts related to markup languages and ontologies from its user. Typically, this means e.g. hiding URIs and complex class hierarchies from the annotators. This should not, however, be done at the expense of expressiveness of the annotations.
2) Annotations should be based on annotation schemas, which are ontologies that define the structure of annotations and guide annotators in their task. The system should form its user-interface automatically according to the annotation schema loaded in it.
3) The system should support collaborative distributed annotation, where the annotation process can be shared among different annotators at different locations.
4) In order to make the system platform-independent from the annotator's point of view, it should be implemented as a web application.

It has been widely argued that automation is needed in the annotation for the Semantic Web [1],[12],[14]. There are, however, limitations to what can be done automatically. These limitations usually lead to either missing or incorrect annotations (low recall/precision) [14]. For example, it is difficult for an automated system to recognize semantic relations between entities it has extracted from a document [3]. Due to the limitations related to automation, most of the current (semi)automatic annotation systems still need human intervention at some point in the annotation process [12]. In Saha, our primary goal has not been the automation of the annotation process, but rather to support the creation of

[1] http://www.seco.tkk.fi/applications/saha/

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

annotations that cannot be produced automatically. Although requiring a lot of work, such annotation can be seen as a collaborative effort, comparable to the creation of different kinds of Wikis[3].

The basic architecture of Saha is depicted in figure 1. It consists of the following functional parts:

1) **Saha application**, which is run on a web-server. It stores and distributes annotations and creates web-pages which form the user interface used in creating annotations.
2) **PostgreSQL-database**, which is used to store the Jena's ontology model containing schema and annotations.
3) **Annotators** using web browsers to interact with the system.
4) **The ONKI ontology-service**, which is used to fetch concepts defined in external ontologies and to share instances created by the annotators.
5) **Applications using the annotations** created with Saha. Annotations can be retrieved in RDF/XML using HTTP-GET

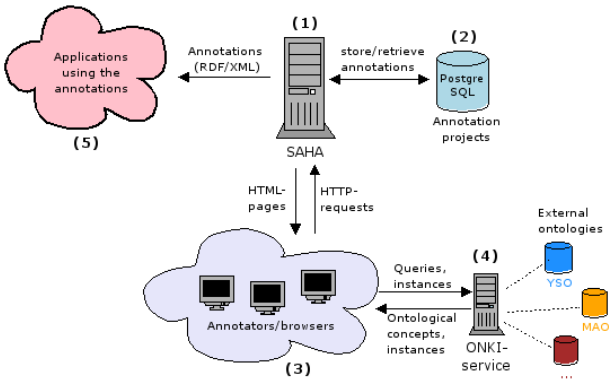

Figure 1. Architecture of Saha

Saha is a web application implemented using the Apache Cocoon[4] and Jena[5] frameworks. It is designed as a web application in order to impose as little requirements as possible to the end user's computational environment. To use Saha, all an annotator needs is an appropriate web browser with an Internet connection. Saha uses extensively techniques such as *Javascript* and *Ajax*[6] in order to provide annotator with simple and versatile user interface.

2.2 User interface

The user interface of Saha is comprised of two main pages. In the first page, depicted partly in figure 2, the annotator sees the annotation classes of an annotation schema and is able to search and open existing annotations or create a new one. The annotator can also create a new subclass for an annotation schema class in order to specify the class hierarchy. This is an optional feature and can be disabled. In order to keep the user interface as simple as possible, more elaborate ontology-editing features, such as creating new properties, are excluded. A new annotation is created by choosing a class and typing the URL of the document to be annotated. Existing annotations can be searched by their labels or by the documents they annotate.

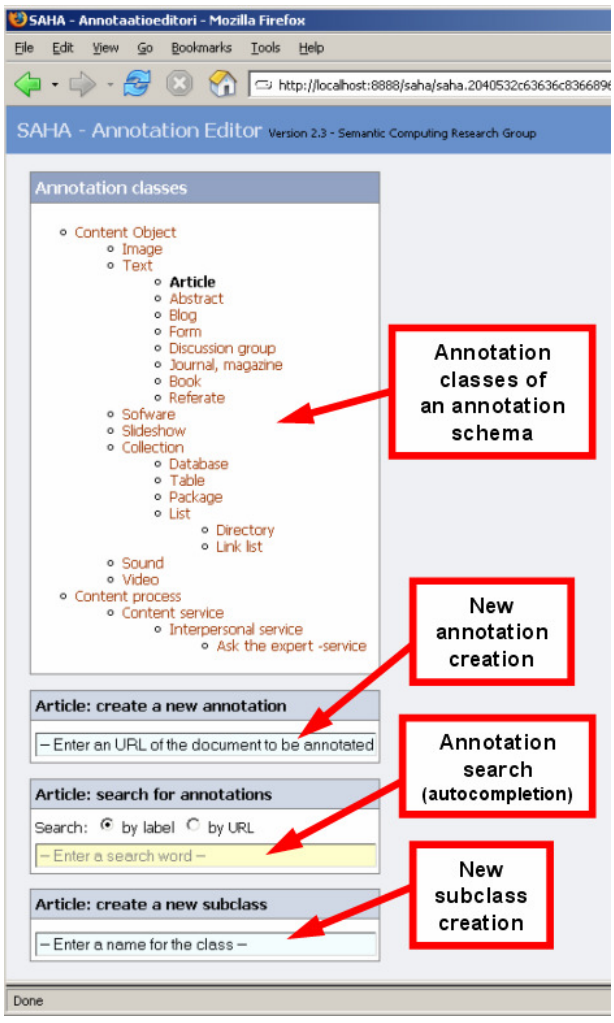

Figure 2. User interface of Saha: the class selection page

In the second page of Saha's user interface, depicted in figure 3, an annotator can edit annotations and view the documents to be annotated. With Saha, any kind of a document (HTML-page, PDF-document etc.) that can be referenced by a URI can be annotated. However, the web browser being used sets the limits to the kinds of documents that can be viewed in Saha's user

[3] http://en.wikipedia.org/wiki/Wiki

[4] http://cocoon.apache.org/

[5] http://jena.sourceforge.net/

[6] http://www.w3.org/2006/webapi/

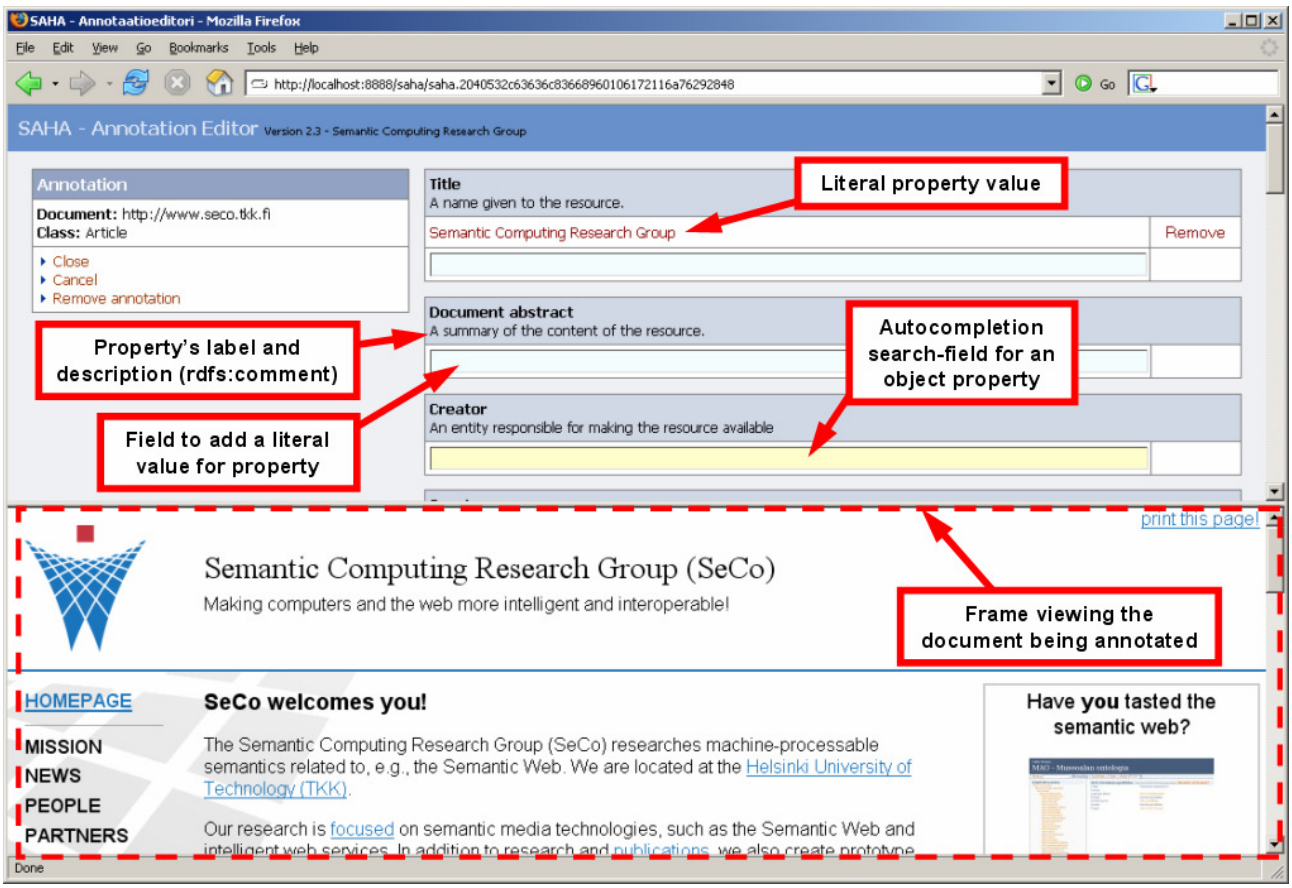

Figure 3. User interface of Saha: the annotation page

interface[7]. The annotation page shows properties of the selected annotation class in a simple form, which can be used to supply values for the properties. In figure 3, an annotation belonging to the class "Article" and annotating the document with the URL "http://www.seco.tkk.fi" is shown. The annotation has one literal value defined for the property "Title". The annotator can edit the value by clicking on it. Defining an object value, such as "Creator" shown in figure 3, is explained in detail in subsection 2.6.

2.3 Annotations in Saha

Annotations created in Saha are based on different metadata annotation schemas, which are defined in OWL. In typical use scenario, a schema describes some specific area of interest and is created by an administrator of annotation project. The purpose of an annotation schema is to define a description template for annotation construction [13]. The schema helps annotators to describe resources in a consistent way and it can be effectively used to construct a generic user-interface for the application.

In Saha, annotations are instantiated classes and properties of an annotation schema, which are linked to the document being described. The linking plays an essential role in annotations, because 1) an annotation is separate from the document it describes, and 2) the way linking is done affects the meaning of the relation between the annotation and the document. There are two ways to associate an annotation with a document in Saha. The first one makes the assertion that the document is an instance of one specific class defined in the schema. It can be thus used efficiently to classify documents. An example of this kind of an annotation is expressed in RDF/XML below:

```
<saha:article rdf:about="http://www.seco.tkk.fi/">
  <dc:title>Semantic Computing Research Group</dc:title>
</saha:article>
```

The second method associates an annotation with the document by using a named property, which is defined in the annotation schema. This idea is similar, e.g., to the usage the property "annotates" in Annotea [7]:

```
<saha:article rdf:ID="20060725104739">
  <saha:annotates rdf:resource="http://www.seco.tkk.fi/"/>
  <dc:title>Semantic Computing Research Group</dc:title>
</saha:article>
```

[7] The document being annotated is viewed in a frame. If the web browser cannot view the document due to unsupported format, a blank page is displayed.

An annotation created with Saha describes a document as a whole, in a sense that it is not being associated with any particular section, sentence or word inside the document. If we wanted to annotate certain parts of the document, we would have to use some fragment identifiers, such as XPointers[8], and could only annotate documents which support fragment identification. The reason why we are not using any identifiers is, that in our approach the main goal of annotations is to put a document in relation to other documents and to serve as a semantically described index, which can be used e.g. to categorize documents or to search a particular document among a group of documents. This is similar to the idea proposed in [9]. Our way of using annotations differs from the approach where annotations are used as additional pieces of information associated with some specific parts of a document and are shown to the reader. This kind of an approach is used e.g. in Annotea.

As stated earlier, an annotation in Saha is an instance of a schema's class that describes some document and is being linked to it using the document's URI. We make a distinction between the annotation of a document and the description of some other resource or concept that is somehow *related* to the document being annotated. In addition to containing classes and properties used to annotate documents, an annotation schema used with Saha can also contain classes and properties for describing resources that are not documents. In other words, an annotation schema can form a basis for the local knowledge base (KB) that contains descriptions of different kinds of resources that may or may not exist on the web. These descriptions or *KB instances* can be used as values of properties in annotations. The KB is refined and extended when new annotations are produced, as they require new KB instances to be created.

2.4 Annotation projects

Saha supports collaborative annotation and sharing of annotations through *annotation projects*. Each annotation schema loaded to Saha forms an annotation project, which can have multiple users as annotators. In practice, an annotation project is a Jena's ontology model stored in a database.

Each user of an annotation project sees all the annotations and KB instances of the project and can edit them, as well as create new ones. An annotation project can belong e.g. to an organization or some other group of people, that are producing annotations using a certain annotation schema. Even though the annotations and KB-instances created in one project cannot be directly edited and used in other projects, KB-instances can be exchanged between projects to allow semantic relations between them (see subsection 2.7).

2.5 Meta-schema

In addition to describing annotations' structure in an annotation schema, we also need a way to define *how* the schema is actually used during the annotation process. Although rules defining the use of a schema could in most cases be expressed using the schema itself, it is often useful to separate the schema design from its use [2]. In Saha, this is done using the *meta-schema*, which is a simple RDF-file that describes how a certain annotation schema is used in a particular annotation project. The meta-schema of Saha is used to define, among others, the following settings for an annotation project:

- The classes of annotation schema, which are shown on the class selection page (see figure 2) and which can thus be used to create new annotations.
- The property (or properties) of a class, whose value is used as rdfs:label of an instance of the class. These are typically literal properties, like the property "name" of the class "Person".
- The order of properties of a class, in which they are shown on the annotation page (see figure 3).
- The classes of the annotation schema whose instances are exported to an ontology service. This feature will be explained later in subsection 2.7.

The meta-schema plays an important role in improving the usability of the system. For example, when an annotator creates a new annotation, the rdfs:label can be automatically constructed for the annotation instance according to the meta-schema, saving the annotator from defining it manually. The meta-schema improves the usability of the system, but it may also enable the use of a same annotation schema in different annotation projects by allowing the schema to be used differently in each project. For example, there may be a different set of annotation classes shown in the class-selection page of each project. The use of the same annotation-schema in different projects enhances the interoperability of annotations as it decreases the need to develop distinct annotation schemas for different projects. In other words, when the same schema is used in different annotation projects, there is no need to specify mappings between annotations created in them.

2.6 Creating and using KB instances

An annotation schema's object properties may have two different kinds of values. They can be either concepts defined in an external ontology or, alternatively, KB instances. The rdfs:range can be used to define the types of instances that are allowed as values of a property. If rdfs:range is not defined, an annotator may choose any type of resource he or she wants to use. When a concept of an external ontology is to be used as a value of a certain property, the project's meta-schema must contain a mapping between the property and the ontology-server hosting the external ontology. According to this mapping, Saha can send the query used to find a concept to the right ontology-server.

When defining an instance value for a property, the annotator must first check if the annotation project's KB already contains an applicable instance to be used as a value. This is to prevent annotators from creating multiple KB instances that all refer to the same resource. If an appropriate instance cannot be found, the annotator can create a new one. Figure 4 illustrates the input-field of object property named "Creator", which has a class named "Person" in its range. An annotator has typed in "tom" in the input-field. The system has searched the KB for the instances of the class "Person" with an rdfs:label value that contains a (sub)string "tom". The search is done on the background using a technique similar to semantic autocompletion [5], a semantic extension for the idea of completing input search keywords

[8] http://www.w3.org/XML/Linking

online, proposed in Google Suggest[9]. The results of the KB search are shown in a menu appearing below the input-field after the search. The annotator may choose to pick one of the instances returned by the query, list all instances of the class "Person" or create a new one.

Figure 4. Instance-search using autocompletion

If the annotator chooses to create new instance, a dialog box is opened providing fields to supply values for instance's properties. Figure 5 illustrates a dialog box, which is used to create a new instance of the class "Person". The input form of the dialog box has the same functionality as the annotation page of Saha has. The instance is created and stored in the KB by closing the dialog box. After that, the instance will be available to be used in annotations.

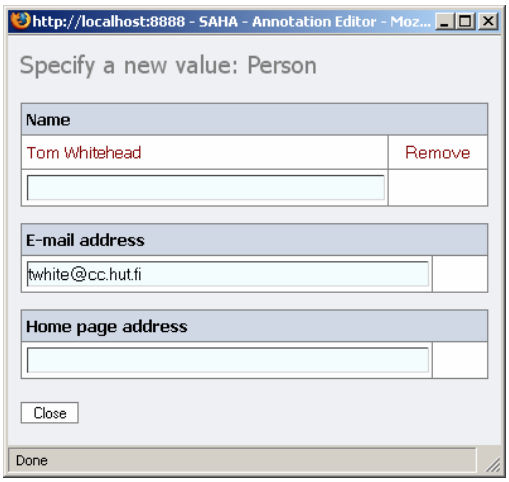

Figure 5. Dialog box for creating new KB-instance

The previous example showed how new instances can be created on the fly when annotating with Saha. The classes used to create new instances need not be as simple as the class "Person" presented here. Instead, they may well have some object-properties that require other instances as values and so forth. Saha can be thus used to describe diverse semantic relations and structures.

[9] http://labs.google.com/suggest/

2.7 Utilizing ontology services

An important feature of Saha is its ability to connect to the ONKI[10] ontology service framework developed in the FinnONTO project [10]. It allows annotators to import concepts defined in external ontologies and also to share KB-instances with other annotators that work on different annotation projects. This kind of instance sharing and use of shared ontologies is vital when pursuing the semantic interoperability between different Semantic Web systems.

Figure 6 illustrates an example of how the ONKI-service can be used when creating annotations in Saha. An annotator is defining a value for an object property and uses the ONKI-browser to find a concept defined in the ontology of an ONKI-service. In this case, the annotator is browsing the Finnish Upper Ontology YSO. When the annotator finds a concept he or she wants to use, it can be imported to Saha by clicking the link named "Fetch Concept". This will set the URI of the selected concept as the property's value in an annotation being created. Another way of finding and importing a concept from the ONKI-service is identical to the instance KB search presented in section 2.6. When using it, an annotator does not have to use the ONKI-browser to locate the concept, but is able to find it with semantic autocompletion.

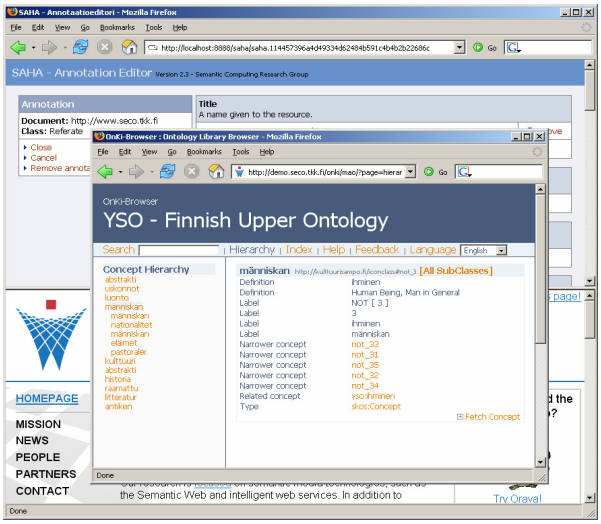

Figure 6. Using the ONKI-browser

In addition to providing a way to browse and use concepts defined in external ontologies such as YSO, ONKI-service can be used to share the instances created in different annotation projects. This is done by declaring an annotation schema's class "public" in the project's meta-schema. When the annotator creates an instance of public class, the instance's data will be sent to the ONKI-service. After this, the instance's URI can be used in other annotation projects using the same ONKI-service. The mechanism described here is practical, as it allows the creation and use of both public and private instances in projects.

[10] http://www.seco.tkk.fi/applications/onki/

3. DISCUSSION

Ontology-based semantic annotations are needed when building the Semantic Web. Although various annotation systems and methods have been developed, the question of how to effectively produce quality metadata still remains largely unanswered. Automated systems are of necessity when masses of documents are being annotated. On the other hand, when there is a need to express more complicated semantic structures and when the precision and quality of annotations are important factors, manual systems are still needed. This shows that different approaches to semantic annotation should not be seen as mutually exclusive, but rather completing each other. We have tried to tackle the problem of creating semantically rich annotations by developing an annotation system Saha that supports the distributed creation of metadata and that can be easily used by non-experts in the field of the Semantic Web.

3.1 Contributions and related work

A number of semantic annotation systems and tools exist today [12],[14]. These systems are primarily used to create and maintain semantic metadata descriptions of web pages.

Annotea [7] supports collaborative, RDF-based markup of web pages and distribution of annotations using annotation servers. Annotations created with Annotea can be regarded as semi-formal, since it does not support the use of ontological concepts in annotations. Instead, they are textual notes which are associated with certain sections of the documents they describe.

The Semantic Markup Tool [9] has user interface that is generated according to an annotation schema in a similar way as is done in Saha. It uses Information Extraction techniques to find different kinds of entities in documents and proposes them for values of the annotation's properties. The schemas it supports are relatively simple and it cannot be thus used to describe more complex semantic relations. The Ont-O-Mat [2], in turn, can be used to describe diverse semantic structures as well as to edit ontologies. It also has a support for automated annotation. The user interface of the Ont-O-Mat is not, however, very well suited for the annotators unfamiliar with concepts related to ontologies and the semantic annotation in general. Another example of the user interface of an annotation tool requiring understanding of the Semantic Web concepts can be found in SMORE [8].

Most of the current annotation systems, like the ones mentioned here, are applications that run locally on the annotator's computer. Because of this, the systems may not necessarily be platform-independent and must always be installed on the user's system, before the annotation can begin. In Saha, these problems are addressed by implementing the system as a web-application. By doing so, the system can be installed and maintained centrally and the requirements for the annotator's computational environment are minimal. The way Saha is designed and implemented also strongly supports the collaboration in annotation, making the sharing of annotations easy.

3.2 Applications

Saha is currently a working prototype. It is in trial use for the distributed content creation of the semantic health promotion portal Tervesuomi.fi[11] [4]. Here much of the content and metadata for the portal will be provided by health experts working at various health organizations in Finland. Saha has also been tested in metadata creation for the Opintie[12] portal, a follow-up version of the educational semantic portal Orava[13] [11] using Learning Object Metadata (LOM).

Initial feedback from end-users not involved in developing the software has been promising but further experimenting is still needed.

3.3 Future work

Future plans include using Saha to provide metadata for additional semantic portals such as CultureSampo[14], the next generation version of the MuseumFinland[15] portal [6]. We also aim to research the integration of the semiautomatic annotation framework Poka[16] with Saha.

ACKNOWLEDGEMENTS

This research is a part of the FinnONTO project funded mainly by the Finnish Funding Agency for Technology and Innovation (Tekes).

REFERENCES

[1] Corcho, O. (2006) *Ontology based document annotation: trends and open research problems*. International Journal of Metadata, Semantics and Ontologies, Vol. 1, No. 1, pp. 47-57.

[2] Handschuh, S. and Staab, S. (2002) *Authoring and annotation of web pages in CREAM*. Proceedings of the 11th international conference on World Wide Web, Honolulu, USA.

[3] Handschuh, S., Staab, S. and Ciravegna, F. (2002) *S-CREAM – Semi-automatic CREAtion of Metadata*, Proceedings of the 13th International Conference on Knowledge Engineering and Knowledge Management (EKAW 2002), Siguenza, Spain.

[4] Holi, M., Lindgren, P., Suominen, O., Viljanen, K. and Hyvönen, E. (2006) *TerveSuomi.fi – A Semantic Health Portal for Citizens*. Proceedings of the 1st Asian Semantic Web Conference (ASWC2006), Beijing, China, poster papers.

[5] Hyvönen, E. and Makelä, E. *Semantic autocompletion*. Proceedings of the 1st Asian Semantic Web Conference (ASWC2006), Beijing, China. Springer-Verlag, forth-coming.

[6] Hyvönen, E., Mäkelä, E., Salminen, M., Valo, A., Viljanen, K., Saarela, S., Junnila, M. and Kettula S. (2005)

[11] http://www.seco.tkk.fi/applications/tervesuomi/

[12] http://www.seco.tkk.fi/applications/opintie/

[13] Operational at http://demo.seco.tkk.fi/orava/

[14] http://www.seco.tkk.fi/applications/kulttuurisampo/

[15] Operational at http://www.museosuomi.fi/

[16] http://www.seco.tkk.fi/applications/poka/

MuseumFinland - Finnish Museums on the Semantic Web. Journal of Web Semantics, vol. 3, no. 2.

[7] Kahan, J., Koivunen, M.R., Prud'Hommeaux, E. and Swick R.R. (2001) *Annotea: An Open RDF Infrastructure for Shared Web Annotations*, Proceedings of the 10th International World Wide Web Conference (WWW10), Hong Kong, China.

[8] Kalyanpur, A., Hendler, J., Parsia, B. and Golbeck, J. (2005) *SMORE – Semantic Markup, Ontology, and RDF Editor*. Available at: http://www.mindswap.org/papers/SMORE.pdf

[9] Kettler, B., Starz, J., Miller, W. and Haglich, P. (2005) *A Template-based Markup Tool for Semantic Web Content.* 4th International Semantic Web Conference ISWC2005, Galway, Ireland. Lecture Notes in Computer Science 3729, Springer. pp. 446-460

[10] Komulainen, V., Valo, A. and Hyvönen, E. (2005) A *Tool for Collaborative Ontology Development for the Semantic Web*. Proceedings of International Conference on Dublin Core and Metadata Applications (DC 2005), Madrid, Spain.

[11] Känsälä,T. and Hyvönen, E. (2006) *A Semantic View-Based Portal Utilizing Learning Object Metadata*. Proceedings of the Workshop on Semantic Web Applications and Tools, the 1st Asian Semantic Web Conference (ASWC2006), Beijing, China. Forth-coming.

[12] Reeve, L. and Han, H. (2005) *Survey of Semantic Annotation Platforms*. Proceedings of the 2005 ACM Symposium on Applied Computing, Santa Fe, USA. ACM Press.

[13] Schreiber, G., Dubbeldam, B., Wielemaker, J.,Wielinga, B. (2001) Ontology-Based Photo Annotation. IEEE Intelligent Systems, 16, 3, pp. 66-74.

[14] Uren, V., Cimiano, P., Iria, J., Handschuh, S., Vargas-Vera, M., Motta, E. and Ciravegna, F. (2006) *Semantic annotation for knowledge management: Requirements and a survey of the state of the art*. Journal of Web Semantics, 4(1):14–28, January 2006.

[15] Valkeapää, O. and Hyvönen, E. (2006) *A Browser–based Semantic Annotation Tool for Distributed Content Creation*. Proceedings of the 1st Asian Semantic Web Conference (ASWC2006), Beijing, China, poster papers.