# Collaborative management of KYOTO Multilingual Knowledge Base: the Wikyoto Knowledge Editor

**Francesco Ronzano, Maurizio Tesconi, Salvatore Minutoli, Andrea Marchetti**
Institute of Informatics and Telematics (IIT) - CNR
Via G. Moruzzi, 1 – Pisa, Italy
{francesco.ronzano,maurizio.tesconi,salvatore.minutoli, andrea.marchetti}@iit.cnr.it

## Abstract

In this paper we introduce the Wikyoto Knowledge Editor, the wiki Web-based environment where the multilingual and multicultural community of KYOTO users interacts to maintain and extend, with respect to their particular domain of interest, the background knowledge resources of the KYOTO system, constituting the Multilingual Knowledge Base. KYOTO is a knowledge-driven system for fact mining from a multilingual collection of information sources concerning a specific domain of interest. Facts are mined from relevant documents that are linguistically and semantically annotated exploiting the Multilingual Knowledge Base, made of several language-specific WordNets all referred to a common Central Ontology.

## 1 Managing knowledge resources: the online collaboration paradigm

Nowadays, knowledge-intensive tasks like text mining, information extraction, semantic grounding of data as well as semantic integration of distinct datasets rely more and more on the background knowledge provided by properly structured knowledge resources such as ontologies or, in general, any semantic or linguistic network. Moreover, as a consequence of the great diffusion experimented by the Semantic Web along with all its related standards and methodologies, ontologies have also gained a central role as the preferred mean to provide shared conceptualizations over the Web in order to exchange and integrate knowledge among different communities of users.

In this context, knowledge resources need *to be properly structured and maintained so as to adequately and correctly model the required information*.

They often have to be tailored to a particular domain and kept up to date to reflect changes in the knowledge they describe; they need to be shared in one or among different communities of users that should reach consensus on the formalized knowledge.

Considering the process of creation and maintenance of knowledge resources, two distinct and often complementary categories of actors have to be actively involved: *knowledge engineers* and *domain experts*. The former ones are usually experienced in knowledge representation and formalization, but often they have little or no expertise in the domain being described. On the contrary, domain experts have a strong knowledge of the domain of interest, but they usually are not able to correctly deal with all the issues related to knowledge formalization.

During the last few years, to allow both of them to take part in the maintenance of a knowledge resource, the **online collaboration paradigm**, also referred to as the wiki paradigm, has been more and more adopted. Thanks to properly structured interfaces, many methodologies and tools, in the majority Web-based, have been proposed so as to collaboratively gather and harmonize all the contributions of the distinct involved actors.

In the remaining part of this Section we briefly describe some relevant example of tools and methodologies to collaboratively manage knowledge resources, summarizing, at the end, the core issues they deal with as well as the features they usually implement. The considered tools can be divided into two great categories:

- *Wiki tools to semantically structure information*: easily exploitable by users with little or no expertise in knowledge formalization, these tools usually help them to explicit structured knowledge starting from unstructured texts. As a consequence, the formalized knowledge can be exploited to improve information navigation and search. In order to be easily accessed by distributed user communities, these tools usually are provided with a Web interface. Three relevant examples are: Semantic *MediaWiki*, *OntoWiki* and *IkeWiki*.
- *Complex knowledge management environments*: they are usually exploited by experts in knowledge formalization in order to fully model information according to a specific and well known representational schema. Due to their usually complex knowledge editing patterns, these tools are mainly realized as standalone desktop applications, even if some of them can be accessed through a Web interface. Among them relevant examples are *Protégé*, the *NeOn Toolkit*, *SWOOP*, *UbisEditor* and *Ontoverse*.

A *Wiki tool to semantically structure information* is **Semantic MediaWiki**[1] (Kroetzsch et al., 2006): it represents a semantic extension of MdiaWiki[2], the wiki engine supporting the most important wiki projects like Wikipedia: it aims at extending Wikipedia contents with more structure and semantics so as to increase their consistency, their reusage and to facilitate their access exploiting semantic search patterns.

**OntoWiki**[3] (Auer et al., 2006) is a tool for collaborative knowledge editing. It is a Web-based ontology editor and a collaborative knowledge acquisition tool: it provides a graphic user interface for an easy and visual editing of RDF knowledge bases. OntoWiki allows defining instances with respect to ontology classes and properties, visualizing particular kinds of data through custom views, avoiding repetition of information and allowing also non expert users to easily understand knowledge organization.

[1] http://semantic-mediawiki.org/wiki/Semantic_MediaWiki
[2] http://www.mediawiki.org/
[3] http://ontowiki.net/Projects/OntoWiki

**IkeWiki[4]** (Shaffert, 2006) is another semantic wiki that allows users to describe pages and links between pages with semantic annotations, in a way compatible with MediaWiki. IkeWiki implements Semantic Web standards like RDF and OWL and supports different levels of knowledge formalization and different users expertise. The KiWi (Knowledge In a Wiki) Project[5], co-founded by the European Commission, starting also from the background provided by the IkeWiki experience, aims at analyzing the most adequate way to define and make semantic wikis usable by huge communities of users, in a user-centric environment.

Taking into account the examples of *Complex knowledge management environments*, we consider **Protégé[6]** (Tudorache and Noy, 2007; Abraham et al., 2008), mainly devoted to knowledge engineers. It is an open-source extensible Java frame work to edit ontologies that can be expressed also in RDF(S) and OWL. It is one of the most widely adopted ontology editing environments supported by a huge community of users. Based on a client-server architecture, it allows to the collaboratively edit ontologies. Moreover, recently *Web Protégé* (Tudorache et al., 2008), a Web Interface for Protégé has been developed for editing ontolgies through a Web browser.

Also the **NeOn Toolkit**[7], developed in the context of the NeOn (Networked Ontologies) Project[8], co-founded by the European Commission, is a rich ontology editing environment. It allows to edit F-logic, RDF(S) and OWL ontologies, thanks to different visualization facilities, supporting also the definition of rules. It is easily extensible exploiting a plug-in mechanism and currently a large number of plug-ins have been developed; they provide support to collaborative Web-based ontology editing as well as to other important tasks like ontology-related argumentation.

**SWOOP**[9] (Kalyanpur et al., 2005) is an ontology browser and editor based on OWL, developed at the Maryland Information and Network Dynamics Laboratory of the University of Maryland. It supports full editing features of OWL ontologies, multiple ontology usage and collaborative annotation over the same ontologies. It has

[4] http: http://ikewiki.salzburgresearch.at/
[5] http://www.kiwi-project.eu/
[6] http://protege.stanford.edu/
[7] http://neon-toolkit.org/
[8] http://www.neon-project.org/
[9] http://www.mindswap.org/2004/SWOOP/

been realized exploiting the Java Webstart technology in order to be accessed by a common Web browser and includes the possibility to rely upon a reasoner to validate the formalized knowledge.

**UbisEditor**[10] (Losky et al., 2009) is a collaborative Web ontology editing environment, developed in the context of the UbisWorld Project at the German Research Center for Artificial Intelligence. It allows users, through their Web browser, to navigate and edit hierarchies of ontological classes as well as to define instances or labels in different languages. Users can also exploit ontology ranking as well as define personalized ontology views.

Other examples of *Complex knowledge management environments* to build and enrich knowledge resources are **AceWiki** [11] (Khun, 2008) or the **CLOnE system** (Funk et al., 2007), both exploiting a controlled language to easily edit and refine ontologies and **Ontoverse**[12] (Bai and Zaloufa, 2007), allowing different actors to collaborate and concurrently build an ontology with a great focus on collaboration aspects and group awareness.

All the described tools underline how many issues need to be faced when we have to provide users with a useful environment to browse and edit knowledge resources: besides the *ease of interaction for the different kinds of users* with different levels of expertise, we have also to consider the *ease of accessing* to such a kind of environments, sometimes provided exploiting a client-server architecture and a browser-based Web interface. *Argumentation facilities* are needed as well as *concurrency controls* and *consistency checks* over the edited knowledge. *User-specific editing permissions* and *versioning support* are two other relevant features.

Starting from all the considerations, examples and analysis done, in the rest of this paper we present the Wikyoto Knowledge Editor, the wiki Web environment useful for editing and enriching the set of linguistic and ontological knowledge resources collected in the KYOTO Multilingual Knowledge Base: it constitutes the semantic reference to carry out the knowledge processing tasks of the KYOTO system, a complex knowledge mining architecture realized in the context of the homonym European FP7 Project. The description of the Knowledge Editor is based on and extends the presentation of the first developed prototype of the Knowledge Editor reported in (Marchetti et al., 2009).

## 2 Wikyoto Knowledge Editor, the wiki for the KYOTO Multilingual Knowledge Base

The Wikyoto Knowledge Editor represents the collaborative environment useful to edit and enrich the multilingual knowledge resources of the KYOTO system, referred to as the KYOTO Multilingual Knowledge Base. KYOTO is a complex knowledge-driven cross-language environment enabling communities of users to mine information form textual documents, sharing the collected facts: it is currently under development in the context of the homonym European FP7 Project[13], carried out also jointly with Asian partners. In KYOTO, most of the knowledge-based tasks are carried out by exploiting the background knowledge collected in the KYOTO Multilingual Knowledge Base: it is the core knowledge reference of the whole system. It is made of a set of linguistic and ontological resources. **The richer the Knowledge Base, the deeper and more effective KYOTO semantic analysis of data can be**.

Thanks to the support of the Wikyoto Knowledge Editor, the Multilingual Knowledge Base can be collaboratively maintained and enriched involving users with different levels of expertise, all belonging to the same community of interest. In particular, by accessing the Web interface of the Wikyoto Knowledge Editor, users can *create, as part of the Multilingual Knowledge Base, a set of customized domain-specific knowledge resources, referring to and extending the generic ones available in KYOTO*. Users can also browse different kinds of external resources like terminologies and thesauri getting useful hints to extend or refine the Multilingual Knowledge Base or defining mappings between these external resources and the same Multilingual Knowledge Base.

All the knowledge formalization activities carried out through the Knowledge Editor immediately affect the text mining and semantic processing capabilities and effectiveness of the whole KYOTO system that relies upon them.

[10] http://ubisworld.ai.cs.uni-sb.de/ontology/
[11] http://www.ontoverse.org/
[12] http://www.ontoverse.org/
[13] http://www.kyoto-project.eu/

In this paragraph, first of all we provide a global overview of the whole KYOTO system, so as to contextualize the Wikyoto Knowledge Editor (Section 2.1). Then we go deeper into details: in particular we describe the first release of the Wikyoto Knowledge Editor. It is useful to show many relevant core features of the wiki way to manage knowledge resources adopted in KYOTO. We present its Web-based architecture as well as the set of external resources it interacts with (Section 2.2). We describe its Web interface (Section 2.3) also by a simple example of usage (Section 2.4).

### 2.1 KYOTO: mining documents and sharing facts by means of the Multilingual Knowledge Base

In this Section we provide a brief overview of how the KYOTO system works. This description doesn't claim to be exhaustive since the main focus of this Section is to introduce the general context where the Wikyoto Knowledge Editor comes into play: to get further information about KYOTO, as well as more technical details about the different parts of the system, you can access the KYOTO Project Official Web Site, *http://www.kyoto-project.eu/*.

KYOTO architecture involves different interacting modules and data repositories so as to realize a complex set of knowledge-driven tasks and activities ranging from *text mining* to *terminology and facts extraction*, *semantic search* and *collaborative editing of knowledge resources* (see Figure 1).

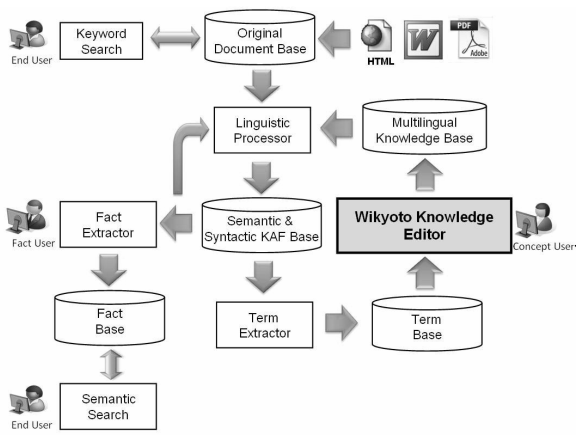


Figure 1. Global architecture of the KYOTO System: knowledge processing activities

The users of KYOTO are grouped by specific roles, depending on the purpose of their interaction with the system. In particular, as we can see from Figure 1, *End Users* mainly access KYOTO to search for information. On the contrary *Fact Users* and *Concept Users* have a stronger interaction with the system: the former ones support the definition of patterns to mine relevant facts from parsed documents, the latter ones directly deal with the Wikyoto Knowledge Editor.

All the documents of interest for KYOTO users (whatever is their format: HTML pages, MSWord or PDF, expressed in all the languages involved in the Project) are collected and indexed in the *Original Document Base*: End Users can perform keyword-based searches on their contents.

All those documents are linguistically and semantically annotated by language-specific *Linguistic Processors*. Semantic annotation is performed by accessing the language-specific knowledge retrieved from the *Multilingual Knowledge Base*. The annotation results are represented, in a language-independent way, thanks to the KYOTO Annotation Format (KAF) and stored in the *Syntactic & Semantic KAF Base*. Semantic annotation of documents can be repeated over time since the background knowledge exploited to perform this task, contained in the *Multilingual Knowledge Base*, is continuously updated by KYOTO Concept Users, by mean of the **Wikyoto Knowledge Editor**.

As previously said, the *Multilingual Knowledge Base* (see Figure 2) represents the semantic backbone for all the information mining and extraction tasks of the KYOTO system: it includes language-specific words and expressions, encoded in different WordNets[14], one for each language. Currently KYOTO includes the WordNets of seven European and Asian languages: English, Dutch, Italian, Spanish, Basque, Simplified Mandarin Chinese and Japanese. Those lexical resources are represented in a custom data format: KYOTO-LMF, derived from the Lexical Markup Framework (LMF, ISO/TC37). Proper mappings between the corresponding language-specific information contained in the WordNets are provided so as to support multilinguism. Moreover, all the WordNets are also mapped to a shared language-neutral ontology, the KYOTO Central Ontology: it is mainly intended to define a frame of reference to semantically characterize the entities described in the WordNets so as to strengthen the possibility to unify the particular and cultural-specific lexical patterns of each language. The Central Ontology, currently under refinement and extension, is mainly intended to

[14] http://wordnet.princeton.edu/

describe processes, relations and states in order to semantically specify the entities described in the WordNets.

WordNets are divided into a *fixed generic part* and one or more *domain specific extensions*.

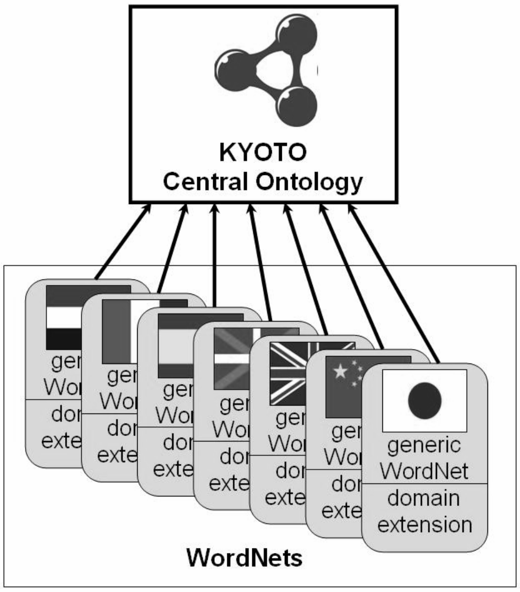


Figure 2. Structure of the KYOTO Multilingual Knowledge Base

The users of KYOTO are grouped Thanks to the **Wikyoto Knowledge Editor**, WordNets Domain extensions, that usually encode knowledge related to a specific domain of interest, can be created, maintained and enriched by KYOTO Concept Users, both domain experts and knowledge engineers; moreover proper mappings of WordNets to the KYOTO Central Ontology can be defined. These tasks are carried out also by exploiting the suggestions of new terms provided by the KYOTO terminological collection, contained in the *Term Base* as well as by accessing external resources like domain specific thesauri and DBpedia. All these issues will be better detailed in the following Sections of this Paragraph.

All KYOTO annotated documents collected in the *Syntactic & Semantic KAF Base* are processed by the *Fact Extractor*, called also Knowledge Yielding Robot or Kybot, storing and indexing the gathered knowledge in the *Fact Base*. Fact Users can define, by mean of the *Wikyoto Kybot Editor*, language-indipendent fact extraction patterns, referred to as Kybot Profiles.

End Users can perform semantic searches in the knowledge collected in the *Fact Base*, on the basis of natural language queries; they can retrieve facts of interest as well as relevant document excerpts.

## 2.2 The architecture of the Wikyoto Knowledge Editor

The Wikyoto Knowledge Editor has been realized as a *Web application* exploitable through a common Web browser, relying upon universally adopted Web technologies and standards like *(X)HTML*, *CSS* and *JavaScript*. To properly work the Wikyoto Knowledge Editor heavily interacts, through AJAX Web APIs, with a set of resources both internal and external to the system, as shown in Figure 3.

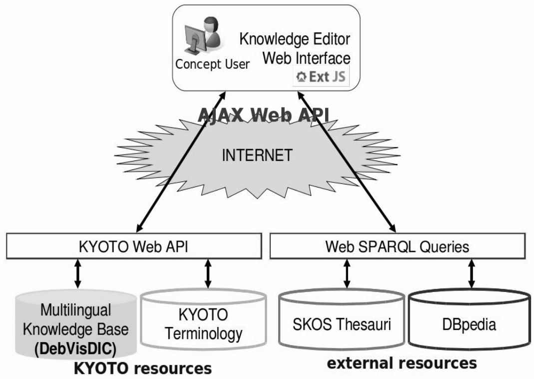


Figure 3. Architecture of the Wikyoto Knowledge Editor

In particular, the resources internal to KYOTO are:

- the **Multilingual Knowledge Base**: it is constituted by the Generic and Domain parts of the WordNets as well as by the KYOTO Central Ontology, all stored in and managed by the DebVisDic Server, a storage environment for semantic resources. *Domain WordNets represent the set of resources collaboratively edited by Concept Users through the Wikyoto Knoeldge Editor*.

- the **Term Base** or KYOTO terminology: it collects the terminology mined by the tybot from the set of KYOTO parsed documents (see Figure 1).

Both have been provided with a customized Web Application Programming Interface (Web API).

The Knowledge Editor exploits a set of external resources to provide useful suggestions to enrich or refine the Domain WordNets or simply to map the domain WordNets to. They include:

- **SKOS thesauri**: they are compliant to the model defined in the Simple Knowledge Organization System Specifications [15] of the World Wide Web Consortium (W3C). They are memo-

[15] http://www.w3.org/2004/02/skos/

rized in and accessed through the VIRTUOSO RDF triplestore[16]. In particular, they are queried through SPARQL[17], the W3C Query Language for RDF. Concept Users can browse these thesauri in order to search for useful suggestions to extend or refine Domain WordNets.

- **DBpedia[18]**: it is the Semantic Web version of Wikipedia. It can be queried through SPARQL and is mainly exploited in order to grasp and suggest textual definitions of synsets.

Also the set of external resources just listed is accessed by the Web Interface of the Knowledge Editor through SPARQL queries performed by REST Web API calls. In this context, the Ext.js[19] JavaScript library has been adopted as the main client-side programming facility and a set of custom JavaScript objects has been defined so as to realize a highly interactive and modular Web Interface in order to make easier knowledge editing tasks for Concept Users.

### 2.3 The Web Interface of the Wikyoto Knowledge Editor

By accessing the Web Interface of the Wikyoto Knowledge Editor, Concept Users can browse and edit the Domain WordNets of the seven languages of KYOTO. They can also navigate, integrate in and map to Domain WordNets, the set of knowledge structures belonging to the KYOTO Terminology as well as to external resources like SKOS thesauri and DBpedia. For the time being, the mappings of Domain WordNets to the KYOTO Central Ontology are still under definition and refinement: they will be integrated in the upcoming versions of the Knowledge Editor.

The interface, shown in Figure 4, is divided into two parts: on the left side there is the *Knowledge Resource Browser* while on the right side there is the *Domain WordNet Browser and Editor*.

The *Domain WordNet Browser and Editor* allows Concept Users to browse the Domain WordNet of a particular language and to extend it by encoding new lexical information relevant to the domain of interest. In the upper part of the *Domain WordNet Browser and Editor* box we can search for a synset typing a particular lemma and choosing among all the synsets referred through it (see the *Lemma Disambiguation Box* in Figure 4). Once a synset is chosen, the hierarchy of all its hyponyms is graphically shown in a tree view in the *Synset Hierarchy Box*. If we click over a node of this tree, corresponding to a specific Domain WordNet synset, all the information related to that particular synset are shown in the *Synset InfoBox*. In particular the definition of the synset is shown on the top of this box. The lower part of the *Synset InfoBox* is made of three tabs useful respectively to list the lemmas associated to a synset, deleting or adding new ones (*Lemma Tab*); to navigate or modify the synset-to synset relations, called internal relations, for instance hypernymy, holonymy, etc. (*Internal relations Tab*); to browse and navigate the set of external relations characterizing the specific synset (*External relations Tab*): external relations are for instance mappings of Domain WordNet synsets to Generic WordNet ones or to equivalent KYOTO terms, to concepts taken from SKOS thesauri and so on. On the top-right corner of the *Synset InfoBox* there is a set of *Synset Editing Buttons*; they are useful to modify the definition of the synset, to add a new synset from scratch or to delete the considered synset.

The *Knowledge Resource Browser,* visualized in the left side of the Interface provides a browsing interface to a set of read-only knowledge resources that can be exploited to enrich Domain WordNets or to simply define proper mapping relations from Domain WordNets synsets to entities belonging to these resources. In particular it is possible to browse and interact with three kinds of resources shown into three distinct tabs.

The **KYOTO Terms Tab** is useful to browse KYOTO term collections, representing a source of suggestions to create domain extensions of the WordNets: terms are extracted by the Tybot by mining the collection of KAF annotated documents. Terms are organized in specialization hierarchies; for instance, if we consider the English term '*bear*' we can access the following hierarchy where '*bear*' is the root:

**bear**
**I— polar bear**
**I— grizzly bear**
**I— brown bears**
**I— sun bear**
**I— ancient bears**

Each term can be characterized by:

- one or more possible ways to refer to it, called *term forms* (plurals, particular spelling or punctuation, etc.);
- one or more *document occurrences*;
- one or more *associated senses*, represented by synsets.

[16] http://virtuoso.openlinksw.com/dataspace/dav/wiki/Main/
[17] http://www.w3.org/TR/rdf-sparql-query/
[18] http://dbpedia.org/
[19] http://www.extjs.com/

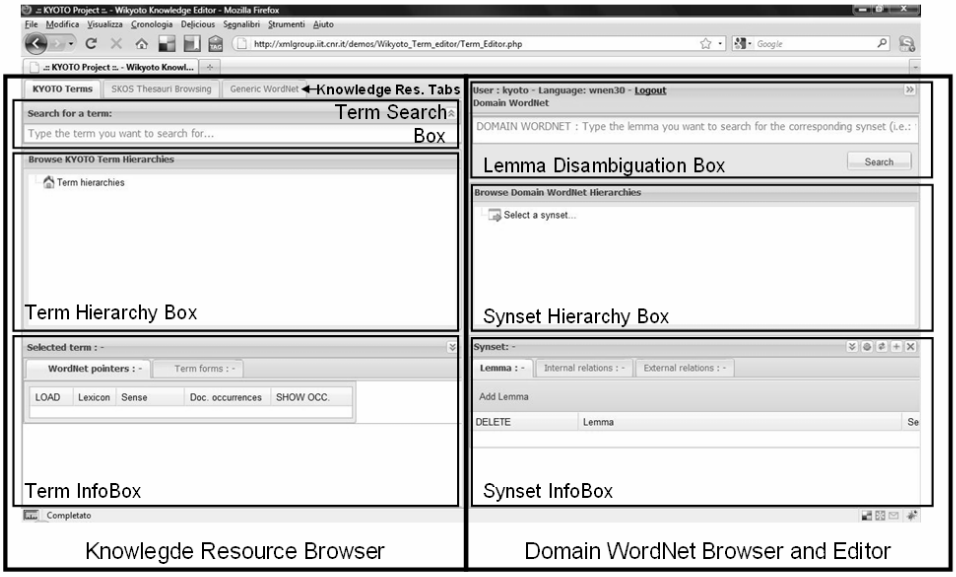


Figure 4. The Web Interface of the Wikyoto Knowledge Editor

As we can see from Figure 4, in the upper part of the KYOTO Terms Tab there is the *Term Search Box*, used to search for a term in the Term Base. Once a term is selected, it, along with the other terms of hierarchy it belongs to, if any, is visualized in a tree-view in the *Term Hierarchy Box*. Every time a term is selected from the opened hierarchies, all the information describing it is loaded in the *Term InfoBox* of the lower part of the interface: all the term forms as well as the WordNet synsets associated to a term are shown along with all its document occurrences.

Through the **SKOS Thesauri Tab**, Concept Users can search for a a particular concept in the set of SKOS thesauri available in KYOTO and browse, through a tree view, the related hierarchies of broader and narrower concepts, as well as the set of related concepts.

The **Generic WordNet Tab** allows Concept Users to browse the Generic WordNet of the language they have chosen once they authenticated to the Knowledge Editor, in a way similar to the *Domain WordNet Browser and Editor* just described, but without editing capabilities. By browsing the three knowledge resources just described, Concept Users can search for specific knowledge structures related to the information they want to express and formalize in the Domain WordNet. In particular, by simply dragging a node from tree of KYOTO terms, of concepts from SKOS thesauri or of Generic WordNet synset and dropping it over the intended Domain WordNet synset visualized in the *Synset Hierarchy Box*, Concept Users can easily:

- create a Domain WordNet synset from a KYOTO term or from a concept of a SKOS thesaurus (or create a hierarchy of Domain WordNet synsets from an entire terms/concepts hierarchy);
- define proper mapping relations between a Domain WordNet synset and one entity among a KYOTO term, a concept of a SKOS thesaurus or a Generic WordNet synset ('equivalence' and 'subsumption' relations are considered).

A practical example of usage of the Wikyoto Knowledge Editor is described in the following Section of this Paragraph. The current prototype of the Knowledge Editor is accessible at: *http://xmlgroup.iit.cnr.it/demos/Wikyoto_Term_Editor/*.

## 2.4 Enrich Domain WordNet: a practical Knowledge Editor usage example

In order to show a possible WordNet extension example by exploiting the Wikyoto Knowledge Editor, we consider the English language and the environmental domain, since the environment is the test-domain chosen for the KYOTO Project. Indeed two important environmental organizations are involved: the WWF[20] and the ECNC[21].

Let's suppose that we are environmentalist KYOTO Concept Users and we are interested in bears and, in particular, in bear species. After having accessed the Wikyoto Knowledge Editor, we can search for the term '*bear*' in the English Domain WordNet in order to see if the related synset has been added to and how it is

[20] http://www.wwf.org/

[21] http://www.ecnc.org/

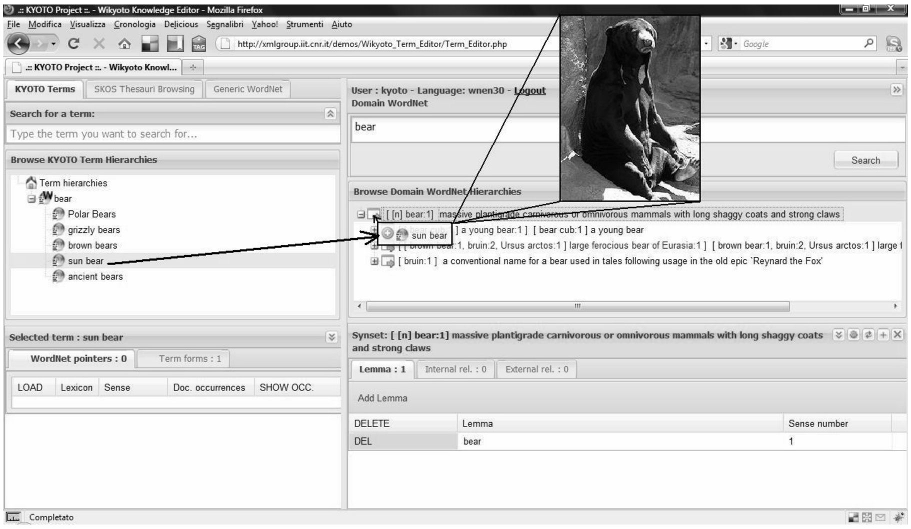


Figure 5. An example of Domain WordNet extension

semantically contextualized. Among all the different meanings, or better all the synsets associated to the word '*bear*', we load the one referring to a '*massive plantigrade carnivorous or omnivorous mammals with long shaggy coats and strong claws*'. As shown in the right side of Figure 5, we visualize the list of all the hyponyms of the bear synset, that are all the more specific concepts present in Domain WordNet usually referring to particular specie of bears like the '*brown bear*' or to other particular meanings.

We would like to check if there are other relevant species of bear that have still not been added to the Domain WordNet. To get new suggestions about other possible kinds of bears we search for the term '*bear*' in the KYOTO Terms Tab and load the related term hierarchy (see the left side of Figure 5). As we can notice, there are five terms that are '*bear*' descendants, probably representing different kinds of bears: they have been extracted by the Tybot from the collection of KYOTO parsed documents. We can immediately see that only the root term '*bear*' has been associated to at least one synset because of the little '*W*' shown on the icon on the left of this term.

We can notice that the term '*sun bear*' is a kind of bear that is not present as a synset in WordNet, but it could be relevant to describe environment related contents and to mine environmental information, thus we decide **to extend the English domain WordNet creating a new '*sun bear*' synset**.

We can simply drag the term '*sun bear*' from the related term hierarchy and drop it over the more general synset (the '*bear*' one) in the hyponymy hierarchy visualized in the *Synset Hierarchy Box*. Thus we can choose if we want simply to define a mapping from the '*bear*' synset to the term '*sun bear*' or to create, starting from the term '*sun bear*', a new child (hyponym) synset of the '*bear*' one; this new synset will be mapped to the term '*sun bear*'.

In this case we choose the latter option and we edit the core information describing the new domain synset we want to create. We can refine its lemma, specify its part of speech and provide a descriptive gloss for '*sun bear*'.

In order to search for the gloss of the synset to be refined or to access other descriptive data, from the Knowledge Editor we can query relevant external resources like, for instance, DBpedia, the Semantic Web version of Wikipedia. We thus can easily retrieve all the meanings for the word '*sun bear*'. We can read the short description (abstract) of each of them, choose and import the correct one to easily provide the gloss for our '*sun bear*' synset. Once completed all the operations, we can confirm the creation of the new domain synset that is added to the English domain WordNet: it is immediately visualized among all the other hyponyms of the '*bear*' synset.

Switching from the *KYOTO Terms Tab* to the *SKOS Thesauri Tab* we can also search for SKOS concepts having '*sun bear*' as label so as to further enrich Domain WordNets with new lemmas or mapping relations.

## 3 Conclusion and future work

Starting from a general overview of the main features of collaborative knowledge editing environments also by providing some real example of such a kind of tools, in this paper we have described the Wikyoto Knowledge Editor, the wiki Web environment for the KYOTO Multilingual Knowledge Base. We have introduced its architecture, providing also a practical usage example. The Knowledge Editor constitutes a core part of KYOTO, a complex knowledge mining system: it allows users to experiment right away the outcome of the collaborative modifications of KYOTO knowledge resources as far as concern text mining and fact extraction processes.

The development of the Wikyoto Knowledge Editor is currently ongoing. The next phases will focus on the implementation of a tracking mechanism for users' modifications to the Knowledge Base as well as the related possibility to manage versioning of changes. We plan to realize more fine-grained concurrency control mechanisms as well as the possibility to define and to properly manage mappings of Domain WordNets to the KYOTO Central Ontology.

In conclusion, we can say that the Wikyoto Knowledge Editor represents an important example of Web-based wiki tool to collaboratively manage a complex set of interlinked knowledge resources, in the context of a complex knowledge processing system like KYOTO: the Knowledge Editor faces many of the fundamental problems that we have pointed out analyzing the tools to collaboratively manage knowledge resources.

## References


Abraham S., Tudorache T., Noy N., and Musen M. A. 2008. *Customizable Workflow Support for Collaborative Ontology Development*. In proc. of the 4th International Workshop on Semantic Web Enabled Software Engineering in the 7th International Semantic Web Conference (ISWC08), Karlsruhe, Germany.

Auer S., Dietzold S., Lehmann J., and Riechert T. 2006. *Ontowiki - a tool forsocial, semantic collaboration*. In proc. of 5th International Semantic Web Conference (ISWC06), Athens, USA.

Bai F. and Zaoulfa E. J. 2007. *Interactive and collaborative ontology development*.

Funk A., Tablan V., Bontcheva K., Cunningham H., Davis B., and Handschuh S. 2007. *CLOnE: Controlled Language for Ontology Editing*. In proc. of The 6th International Semantic Web Conference, 2nd Asian Semantic Web Conference, ISWC 2007 + ASWC 2007, Busan, Korea.

Kalyanpur A., Parsia B., Sirin E., Cuenca Grau B., Hendler J. 2005. *Swoop: A 'Web' Ontology Editing Browser*. Journal of Web Semantics Vol 4(2), 2005.

Khun T. 2008. *AceWiki: Collaborative Ontology Management in Controlled Natural Language*. In proc. of the 3rd Semantic Wiki Workshop co-located with the 5th European Semantic Web Conference (ESWC07), Tenerife, Spain.

Kroetzsch M., Vrandecic D. and Voelkel D. 2006. Semantic MediaWiki. In proc. of the 5th International Semantic Web Conference (ISWC06), Athens.

Losky M., Heckmann D. and Kobayashi I. 2005. *UbisEditor 3.0: Collaborative Ontology Development on the Web*. In proc. of Web 3.0: Merging Semantic Web and Social Web, Workshop at Hypertext 2009, Torino, Italy.

Marchetti A., Minutoli S., Ronzano F. and Tesconi M. 2009. *Wikyoto Knoledge Editor: the collaborative environment to manage Kyoto Multilingual Knowledge Base*. In proc. Of the 6th International Conference on Knowledge Management (ICKM 09), Hong Kong, China.

Schaffert S. 2006. Ikewiki: A semanticwiki for collaborative knowledge management. In proc. of the 1st International Workshop on Semantic Technologies in Collaborative Applications (STICA 06), Manchester, U.K.

Tudorache T. and Noy N. 2007. *Collaborative Protégé*. In proc. of the Social and Collaborative Construction of Structured Knowledge Workshop at World Wide Web Conference (WWW2007), Banff, Canada.

Tudorache T., Venditti J. and Noy N. 2008. *Web-Protégé: A lightweight owl ontology editor for the web*. In proc. of the OWL experiences and directions, first International Workshop in the 7th International Semantic Web Conference (ISWC08), Karlsruhe, Germany.