UNIVERSITÀ DI PISA

Scuola di Dottorato in Ingegneria "Leonardo da Vinci"

Corso di Dottorato di Ricerca in
INGEGNERIA DELL'INFORMAZIONE

Tesi di Dottorato di Ricerca

Collaborative editing of knowledge resources for cross-lingual text mining

Autore:

Francesco Ronzano _____

Relatori:

Ing. Alessio Bechini _____

Dott. Andrea Marchetti _____

Anno 2011

ACKNOWLEDGEMENTS

The work presented in this thesis has been carried out at the Institute of Informatics and Telematics (IIT) of the National Research Council (CNR) of Pisa. I would like to acknowledge my advisors. In particular, I would like to thank Andrea Marchetti for his careful guidance and the friendly support provided. I acknowledge Piek Vossen and the members of the Computational Lexicology & Terminology Lab (CLTL) of the Vrije University for the many fruitful discussions and their support during my stay in Amsterdam. I would like to thank Maurizio Tesconi for his valuable collaborations together with all the members of the Web Applications for the Future Internet Group I have team up with. I would like to acknowledge all the people involved in KYOTO for the many valuable discussions and contributions hold during the course of the project.

I'm grateful to my parents Anna and Marcello, and my sister Nadia for their kind help during these years. I thank Diana for her lovely support. Even if I don't mention each one, I gratefully acknowledge all the people that directly or indirectly have supported me in this work.

ABSTRACT

The need to smoothly deal with textual documents expressed in different languages is increasingly becoming a relevant issue in modern text mining environments. Recently the research on this field has been considerably fostered by the necessity for Web users to easily search and browse the growing amount of heterogeneous multilingual contents available on-line as well as by the related spread of the Semantic Web. A common approach to cross-lingual text mining relies on the exploitation of sets of *properly structured multilingual knowledge resources*. The involvement of huge communities of users spread over different locations represents a valuable aid to create, enrich, and refine these knowledge resources. *Collaborative editing Web environments* are usually exploited to this purpose.

This thesis analyzes the features of several knowledge editing tools, both semantic wikis and ontology editors, and discusses the main challenges related to the design and development of this kind of tools. Subsequently, it presents the design, implementation, and evaluation of the Wikyoto Knowledge Editor, called also Wikyoto. Wikyoto is the collaborative editing Web environment that enables Web users lacking any knowledge engineering background to edit the multilingual network of knowledge resources exploited by KYOTO, a cross-lingual text mining system developed in the context of the KYOTO European Project.

To experiment real benefits from social editing of knowledge resources, it is important to provide common Web users with simplified and intuitive interfaces and interaction patterns. Users need to be motivated and properly driven so as to supply information useful for cross-lingual text mining. In addition, the management and coordination of their concurrent editing actions involve relevant technical issues.

In the design of Wikyoto, all these requirements have been considered together with the structure and the set of knowledge resources exploited by KYOTO. Wikyoto aims at enabling common Web users to formalize cross-lingual knowledge by exploiting *simplified language-driven interactions*. At the same time, Wikyoto generates the set of complex knowledge structures needed by computers to mine information from textual contents. The learning curve of Wikyoto has been kept as shallow as possible by hiding the complexity of the knowledge structures to the users. This goal has been pursued by both enhancing the simplicity and interactivity of knowledge editing patterns and by using natural language interviews to carry out the most complex knowledge editing tasks. In this context, TMEKO, a methodology useful to support users to easily formalize cross-lingual information by natural language interviews has been defined. The collaborative creation of knowledge resources has been evaluated in Wikyoto.

INDEX

LIST OF FIGURES AND TABLES

FIGURES

TABLES

I. INTRODUCTION

The exploitation of social massive contributions is becoming one of the most adopted patterns to collaboratively create distinct kinds of content over the Web. Currently, common Web users can access a huge set of Web applications to edit, publish on-line and discuss almost any kind of data of their interest. The whole set of new browser-based technologies, methodologies and interactions that have led the development of Web applications during the last few years has given a considerable boost to the adoption of on-line social content creation patterns. They support direct user-to-user communication and content-sharing by enhancing the interactivity of Web applications. All these trends are usually referred to as the *Web 2.0*.

Since the beginning of 2000's, the Web has been affected by a growing exploitation of semantic technologies in order to better structure contents and to help users to deal with the enormous amount of disparate information exposed on-line. Both deep and shallow patterns have been proposed to automate the management of Web data by semantics. Under the guidance of the World Wide Web Consortium, an elaborate infrastructure to create, publish and integrate on-line semantic metadata has been developed and formalized by defining a set of standards and best practices. These activities and trends are globally pointed out as the basic constituting themes of the *Semantic Web*.

In this scenario, the necessity to deal with and provide access to Web contents in many different languages cannot be underestimated. Even if English is still the most adopted language over the Web, during the last few years the amount of contents expressed in other languages has considerably grown. As a consequence, in order to limit the rise of on-line isolated linguistic islands of information, it is essential to take into account the need to manage multilingualism and thus to cope with a *Multilingual Web*.

This research involves aspects related to the three trends of on-line information management just mentioned: the *Web 2.0*, the *Semantic Web* and the *Multilingual Web*. The core topic *is the design and the implementation of Web environments to collaboratively edit multilingual knowledge resources exploited by cross-lingual text mining systems*. These systems aim at mining and integrating useful information from textual documents in different languages, like for instance Web contents. To carry out this task, it is possible to leverage on a set of properly structured multilingual knowledge resources that formalize linguistic and conceptual

data useful to automatically process texts. These knowledge resources need to be created, enriched and customized with respect to a specific domain of interest representing the informative target of the text-mining system. We investigate *how to carry out this task by involving common Web users enabling them to effectively contribute by proper collaborative Web applications*. In particular, we have developed Wikyoto, the collaborative Web tool to enable widespread social contributions in the editing and domain adaptation of the multilingual knowledge resources of KYOTO, a new generation cross-lingual text mining system. Our final aim has been to design and create a Web application that allows common Web users to easily extend and maintain KYOTO multilingual knowledge resources. In this way the effectiveness of cross-lingual text mining tasks in KYOTO can be improved by exploiting social contributions.

We have analyzed several collaborative editing Web environments for knowledge resources. Thus, we have considered the organization and the way KYOTO multilingual knowledge resources are exploited so as to understand the kind of formalized knowledge structures required by KYOTO to process texts in different languages.

We have realized that the language features that need to be modelled so as to support Natural Language Processing are quite different and sometimes complementary to those ones usually addressed in widespread formal models of knowledge resources like on-line ontologies. We have found out that the formalization of non-rigid concepts that can characterize instances for a limited amount of time with respect to their whole life is essential to mine relevant information from texts.

To enable cross-lingual text mining it is important to manage the specific traits of distinct languages described by their linguistic features, but at the same time, also the flow of information across languages needs to be supported. In the set of knowledge resources of KYOTO this aim is achieved by mapping language-specific knowledge structures to a language-independent ontology. Current editing tools for knowledge resources do not usually specifically support the possibility to gather linguistic information. These tools often explicitly manage multilingualism only by allowing the association of labels in different languages with concepts. In our approach each language can represent distinct concepts mapped to the same ontological class or related to multiple ontological entities by exploiting different patterns.

We have adopted a language-driven approach to knowledge editing in order to involve common Web users by requiring minimum efforts from them. We have tried to accomplish knowledge editing tasks starting from

the informative contents of natural language sources like texts and excerpts, but also domain terminologies. In this way, users feel more comfortable since they have to deal with the language as they use it in everyday life without handling complex knowledge structures. Also natural language interfaces based on user interviews have been exploited to simplify the fulfilment of complex knowledge editing tasks.

The language-specific knowledge resources of KYOTO are compliant to the WordNet model of linguistic knowledge. This model is inspired by psycholinguistic theories of human lexical memory and based on the notion of a synset, defined as the set of synonyms describing a concept. This model should be easier to understand for Web users if compared to the rigid set of formal constraints that characterize ontologies.

Considering the remarks just exposed, we can figure out Wikyoto as a balance between formal knowledge editing environment and lightweight ones. It aims at keeping knowledge editing tasks easy to common Web users like in lightweight environments. At the same time, Wikyoto supports users in formalizing a complete and rich set of knowledge and language features useful to improve the outcome of cross-lingual text mining without having to know and understand these foremost structures.

Wikyoto helps communities of users to formalize knowledge considering their own point of view on a domain of interest. The created knowledge resources are tailored to a specific field of interest and to a defined group of users, thus representing their informative needs. As a consequence, cross-lingual text mining applications that exploit these knowledge resources are in some way domain customized and better headed to extract information relevant to the users.

Wikyoto faces basic technical issues of Web 2.0 collaborative editing environments: client-server partitioning, client side logic support, data integration and management of concurrency and consistency are the most relevant ones.

I.I CONTRIBUTION AND SIGNIFICANCE

This research addresses the topic of collaborative editing of knowledge resources for cross-lingual text mining. The collaborative content creation paradigm has been investigated and applied to the set of knowledge resources exploited in KYOTO, a new generation cross-lingual text mining system.

Currently knowledge resources showing the same structures as those ones used in KYOTO, such as WordNets and ontologies, are widely adopted to support several semantic information processing tasks in the related research community. As a consequence, our focus on the collaborative editing of the KYOTO Multilingual Knowledge Base does not imply any loss of generality in this research. On the contrary, KYOTO provides a real application scenario where the collaboratively edited knowledge resources are exploited to perform cross-lingual text mining. Indeed in KYOTO the text processing and semantic search strongly rely on the knowledge formalized in the Multilingual Knowledge Base.

We have made the following investigations and contributions by this dissertation work:

- Analysis and comparison of the most adopted knowledge editing tools both semantic wikis and ontology editors;
- Analysis of *the linguistic features that need to be formalized in order to support cross-lingual text mining*, by considering the knowledge-based approach of KYOTO to mine multilingual textual contents;
- Design of intuitive *knowledge visualization patterns and language-driven user interactions useful to help non-expert users to easily edit the linguistic and ontological knowledge resources* constituting the KYOTO Multilingual Knowledge Base;
- Design and implementation of *Wikyoto*, a Web based collaborative knowledge editing system useful to *enable users with no experience in knowledge formalization to easily edit the knowledge resources for cross-lingual text mining of KYOTO*. In particular this activity includes:
    - the definition of the user interaction patterns;
    - the design of the Web application;
    - the identification and integration of the informative contents of multiple on-line data sources to support knowledge editing;
    - the implementation of the system by facing software concerns of Web 2.0 collaborative environments;

  - the incremental refinement of the system on the basis of users' feedback.

- As a more experimental component of Wikyoto, definition and implementation of *TMEKO, a methodology useful to support users to easily formalize cross-lingual information by natural language interviews*;
- *Evaluation of Wikyoto* by considering the collaborative creation of knowledge resources.

I.II OUTLINE

The contents of this thesis are organized into four Chapters.

Chapter 1: Knowledge resources: background knowledge to semantically structure Web contents

This chapter provides an introduction to knowledge resources considering their exploitation as background knowledge to support Web data semantics. Section 1.1 describes the recent Web information explosion responsible for on-line information overload and the related need to better structure Web contents by making explicit their semantics. Section 1.2 introduces Natural Language Processing (NLP) methodologies and the Semantic Web stressing their complementary usefulness in mining, structuring and integrating Web contents by means of semantics. Finally, in Section 1.3 knowledge resources are described since they represent the background knowledge necessary to support both the creation of semantic meta-data by means of text mining and NLP and the exploitation of these meta-data on a Web scale in the context of the Semantic Web. In particular computational lexicons like WordNet and ontologies are introduced.

Chapter 2: Editing knowledge resources: the wiki way

This chapter describes knowledge editing tools, with particular emphasis on those tools adopting the wiki paradigm. Section 2.1 motivates the usefulness of collaborative editing methodologies to manage knowledge resources. Section 2.2 provides detailed review of the most relevant environments to edit knowledge resources, considering both semantic wikis and ontology editors. In Section 2.3 a classification of tools according to a common set of descriptive criteria is presented. A comparison among the tools allows the definition of a core set of desirable features of a collaborative knowledge editor. To conclude, Section 2.4 discusses the importance of users' motivation in order to promote massive contribution in knowledge editing.

Chapter 3: Wikyoto Knowledge Editor: the collaborative Web environment to manage KYOTO knowledge resources

This chapter provides an overview of KYOTO and a detailed description of Wikyoto. Section 3.1 is devoted to analyze some important aspects of KYOTO, such as the knowledge-based approach adopted to perform cross-lingual text-mining, the exploited knowledge resources and the architecture of the system. Section 3.2 is focused on Wikyoto. The motivations, the design, the implementation, and the evaluation of the Wikyoto are

discussed in detail. Examples of knowledge editing task that can be performed in Wikyoto are provided. Finally, TMEKO, a more experimental component of Wikyoto, useful to support users to easily formalize cross-lingual information by natural language interviews is presented.

Chapter 4: Conclusions and perspectives

Final conclusions are presented. General consideration and perspectives concerning the work presented in this thesis are discussed.

1. KNOWLEDGE RESOURCES: BACKGROUND KNOWLEDGE TO SEMANTICALLY STRUCTURE WEB CONTENTS

During the last few years, the Web has progressively turned into the more widespread and pervasive global information sharing space. Currently, over the Internet huge amounts of contents can be easily exposed worldwide and people can access a growing number of on-line services. Users located all-over the world can easily interact, discuss, publish data and create communities of interest thanks to the visual, intuitive interfaces and the great interactivity characterizing many new Web applications. Even if English still represents the most used language to publish information over the Web, many other language communities are quickly growing. As a consequence, the spreading of information across different languages has become a distinguishing feature of the current Web.

In this varied, dynamic and rapidly evolving scenario, it is common for Web users to experience *information overload*. Users are flooded by huge quantities of contents that are often disconnected. Consequently they experience difficulties in browsing and dealing with different types of data in order to find the needed information.

From the beginning of the last decade, many initiatives have attempted to solve or at least reduce the information overload by improving the structure and organization of Web contents. Proper formalisms to represent the information published on-line have been proposed to make explicit data semantics. As a consequence, software agents are enabled to search, aggregate and integrate data from distinct Web sources. All these trends are usually referred to as the *Semantic Web*. The current Web is mainly made of human readable contents. The Semantic Web aims at building a Web of Data made of a huge heterogeneous network of automatically processable semantic descriptions of information, called semantic meta-data.

To realize the possibility to automatically retrieve, integrate and interpret semantic meta-data on a Web scale, it is essential to *foster the creation of semantic meta-data* from and in parallel to the huge amount of human readable contents currently available on-line, but also to *define proper formalisms, standards and best practices so as to represent semantic meta-data* and *to publish them over the Web*.

Semantic meta-data can be automatically created by exploiting *text mining approaches* by parsing unstructured or semi-structured human readable contents. *Natural Language Processing* (NLP) techniques are usually

adopted in order to make explicit the syntactic and semantic features of a text by annotating its contents. By relying on these annotations, extraction patterns can be applied so as to mine semantic metadata from texts. Depending on the features of the considered extraction patterns, NLP techniques performing shallow or deep text annotation may be necessary. As an alternative to text mining approaches, Web users can be directly involved in the definition of semantic meta-data together with the creation of the contents they publish on-line. Due to the considerable efforts that are usually required, the activity of manual definition of semantic metadata usually takes place in very specific situations. Some of the wiki environment for knowledge resources that are described in Chapter 2 support this process.

Regarding the definition of proper formalisms, standards and best practices, a set of guidelines to represent, publish and integrate on-line semantic meta-data has been issued in the context of the Semantic Web, under the guidance of the World Wide Web Consortium (W3C). A growing number of on-line data sources, ranging from public organizations to private institutions, are exposing their data as semantic datasets made of collections of semantic meta-data. Linked Data currently represents the most relevant initiative aimed at coordinating all these efforts so as to build a Web of interconnected semantic meta-data.

In any case the creation of semantic meta-data as well as their representation and integration can leverage some sort of background knowledge that supports both the mining of textual contents by means of NLP techniques and the homogeneous description and interconnection of distinct on-line semantic datasets. *Knowledge resources*, referred to also as *Knowledge Organization Systems* (KOS), constitute a specific type of background knowledge. Knowledge resources are collections of information describing some set of distinctive features of the entities that characterize a domain of interest. Each knowledge resource can provide information related to a general or a specific domain (i.e. environment, biology, genomics, etc.) with a proper level of data structuring. There are many different types of knowledge resources. This thesis focuses on *lexicons* and *ontologies*. Lexicons provide background knowledge to semantically interpret the meaning of textual contents by NLP techniques. Ontologies are usually exploited as a mean to homogenize and integrate semantic meta-data from distinct on-line sources.

The first section of this chapter analyzes in detail the distinguishing features of the recent Web information explosion responsible for information overload. The second section contains a description of both NLP

methodologies to perform text mining and the main set of knowledge representation standards and best practices that constitutes the basis of the Semantic Web. Their complementary usefulness in mining, structuring and integrating Web contents by means of semantics is stressed. Finally, the third section introduces knowledge resources as the background knowledge necessary to support both the creation of semantic meta-data by means of text mining and NLP and their exploitation on a Web scale in the context of the Semantic Web. In particular computational lexicons like WordNet and ontologies are considered and their relations are described.

1.1 WEB INFORMATION EXPLOSION: MASSIVE, HETEROGENEOUS, USER GENERATED, MULTILINGUAL CONTENTS

The Web has radically changed in the last decade, becoming a worldwide pervasive global information sharing space. The amount of Internet users and the size of Web contents have considerably increased. Furthermore the composition of the Web users' community and the typology of on-line contents have been greatly diversified, thus introducing a considerable amount of heterogeneity. Virtually, the increase and diversification of information available on-line should help users. In reality, the volume and heterogeneity of Web contents, if managed wrongly, can difficult the access to the right information.

From 2000 to 2009 the total number of Internet users has increased four times: at the end of 2009 Internet users constituted the 26,1% of the world population, even if with substantial differences between countries1. Impressive growing rates have also characterized the amount of the contents that are accessible on-line. It is really a difficult task to determine the size of Web contents and how they have increased with time because of their greatly heterogeneous organization, distribution and structuring as well as because of the different ways they can be generated. Some indicators of the growth of the Web can be the increase of the number of Web pages indexed by Google, and the number of active on-line servers. Concerning Google, in 1998 the search engine crawled and indexed contents from about 26 million of unique URIs. In 2008 Google claimed[2] to have reached one trillion of unique crawled URIs. On May 2010 Google index was estimated to include about 15 billion Web pages[3]. With respect

[1] Internet World Stats - http://www.internetworldstats.com/

[2] The Google Blog, 'We knew the Web was big…' http://googleblog.blogspot.com/2008/07/we-knew-web-was-big.html

[3] Daily statistics about the size of the World Wide Web - http://www.worldwidewebsize.com/

to the active on-line Web servers, between 2001 and 2010 its estimated number has grown from about 13 million up to about 100 million[4].

Furthermore, in the last years *the process of Web contents generation has actively involved a growing number of common Web users*. The Web has progressively turned into an increasingly interactive place where everyone can easily publish information, share and discuss topics of interest, communicate with other users, take part in communities. All the contents thus created and exposed on-line, usually referred to as *user generated contents*, are becoming more and more significant in the context of the Web. Indeed, during the last few years, Web resources of great relevance have been collaboratively built thanks to the spontaneous content editing efforts of communities of users. An important example is Wikipedia[5] where people contribute to enrich the greatest online multilingual encyclopaedia including in its English version more than 3,5 million articles[6] (December 2010). Also the whole set of blogs together with their interconnections, usually referred to as *blogosphere*, has experimented a considerable growth during the last few years. Since 2004 Technocrati[7], an Internet search engine specialized in indexing and searching contents from blogs, launched at the end of 2002, publishes every year 'The stat of the blogosphere' report[8]. In 2004 Technocrati was tracking about 4 million blogs. The number of tracked blogs has grown up to 57 million in 2007 reaching 112 million in 2008. The number of Web users that take part in social processes of on-line content creation is expected to increase during the next years[9]. As a proof of fact, it has been estimated that in 2009 44,6% of US Internet users published some content on-line, and this percentage is expected to grow up to 51,8% in 2013.

Another topic worth to mention is the increase of multilingualism on the Web. Many new language communities have considerably increased size over the Web during the last few years[10]. Even if the English native speakers still account for the greatest number of Web users (27.3% - 537 million), their predominance is vanishing. Chinese native speakers already account for 22.6% (445 million) of Web users and Spanish (7.8% - 153 million) and Japanese (5% - 99 million) are gaining ground (see Figure 1). Between 2000

[4] On-line Web Server Survey - http://news.netcraft.com/archives/category/web-server-survey/
[5] English Wikipedia - http://en.wikipedia.org/wiki/Main_Page
[6] Wikipedia statistics - http://stats.wikimedia.org/EN/TablesArticlesTotal.htm
[7] Technocrati - http://technorati.com/
[8] Technocrati, The state of the blogosphere - http://technorati.com/state-of-the-blogosphere/
[9] User generated contents: more popular than profitable - http://www.emarketer.com/Report.aspx?code=emarketer_2000549
[10] Internet World Statistics: http://www.internetworldstats.com/stats7.htm

and 2010 the number of English native speakers browsing the Web increased by 281%, but the percentage of non-English speakers rose even at a higher pace. For instance the number of Arabic native speakers increased by 2501%, that of Russians by 1825%, that of Chinese by 1277%, that of Portuguese by 989%, and that of Spanish by 743% (see Figure 2). Therefore the spreading of Web contents across different languages is becoming more and more a real and relevant issue.

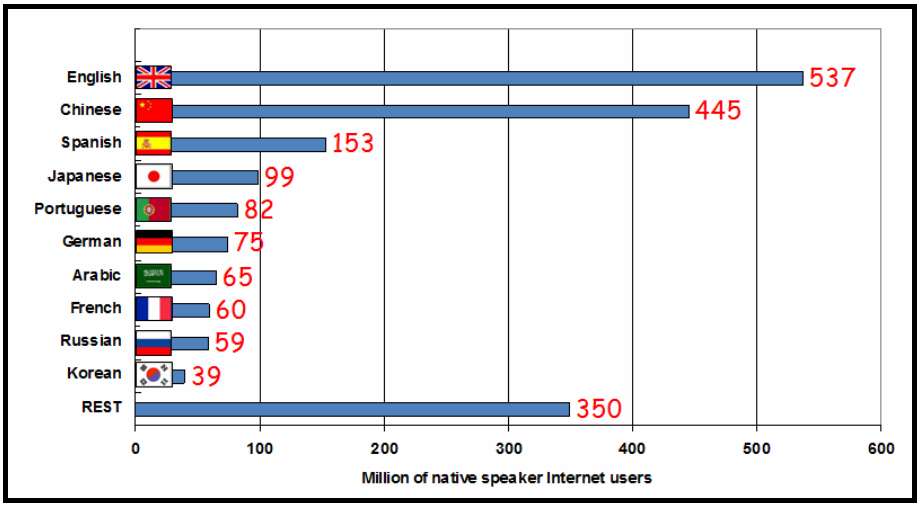


Figure 1: Languages spoken over the Web - June 30, 2010 - Source: Internet World Stats

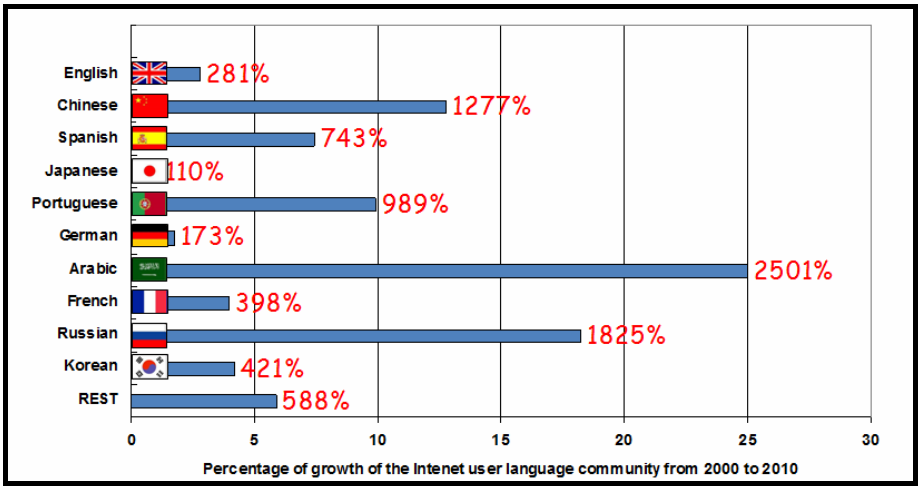


Figure 2: Growth of language communities between 2000 and 2010 - June 30, 2010 Source: Internet World Stats

The general trends concerning the exponential growth of information accessible on-line together with the great diversification of Web contents across languages, creation patterns, and typologies may be referred to as *Web information explosion*. The expression 'information explosion' was used for the first time in the 60's to point out the proliferation of newspapers and press agencies that characterized that period and thus the difficulty to deal with and access to all the informative contents that were available. Currently we are experiencing a similar situation concerning Web contents.

Web users are often overwhelmed by the possibility to access real-time huge amounts of distributed information. Consequently, they experience difficulties in searching, evaluating and selecting the most useful sources as well as in summarizing and combining many distinct contents. All these issues are usually referred to as *Web information overload* [1]. Web information overload occurs when the information we can consult on-line exceeds our ability to process it: thus the information we can access constitutes more a hindrance than a help, even though the information is potentially useful [2, 3]. Cognitive learning theories have identified by the expression '*working memory*', the ability of our brain to temporarily store the information we acquire from the outside as well as to manipulate these data in order to support complex cognitive tasks like comprehension, learning, and reasoning [4]. Working memory has a limited capacity that constitutes the bottleneck for our information acquisition and elaboration process. When we acquire too much information from the outside and we exceed this limit, we experience information overload. Information overload can have negative effects on Web users, and worsen their Web experience. When users are overwhelmed by information, they can find difficulties in assessing the trustworthiness and the completeness of contents. These problems, together with the feeling of lack of control over the situation, can delay the decision-making process.

1.2 NATURAL LANGUAGE PROCESSING AND THE SEMANTIC WEB: MINING, STRUCTURING AND INTEGRATING CONTENTS ON A WEB SCALE BY MEANS OF DATA SEMANTICS

In order to solve or at least to reduce the information overload experienced by Web users, the structure and organization of on-line contents need to be improved so as to enable automated data retrieval, aggregation as well as enhanced visualization patterns. In this context, the exploitation of *Web data semantic*s currently represents one of the most investigated strategies to deal with Web information overload. By making explicit and automatically processable the meaning of on-line contents, Web users can

be supported by software agents in collecting, interlinking and filtering on-line data on the basis of their informative needs. All these trends, referred to as the Semantic Web, aim at building, in parallel with the current Web of human readable data, a Web of automatically processable semantic meta-data interlinked across distributed sources by giving information a well defined meaning, better enabling computers and people to work in cooperation [5].

In order to realize this vision, it is essential to speed up the creation of huge amounts of semantic meta-data from unstructured Web contents as well as to define and adopt a shared set of standards and best practices to represent and publish on-line these meta-data.

In order to *create meta-data* a number of strategies can be exploited. *Internet users may be directly involved* in meta-data editing activities. However this task is often time-consuming and it is difficult to get massive involvement of Web users. Alternatively, meta-data can be extracted automatically from existing Web contents. Automated extraction often relies on the availability of *structured or semi-structured Web contents*. The structure of these data represents the main feature exploited in order to understand data semantics and thus to create semantic meta-data. Nevertheless a considerable portion of on-line contents is constituted by *unstructured natural language texts*. In this case more sophisticated meta-data extraction methods are required so as to properly understand their meaning. These methods range from shallow information extraction patterns to more complex approaches that exploit Natural Language Processing (NLP) techniques so as to mine textual contents.

In subsection 1.2.1, NLP is introduced as a mean to enable the automatic understanding of the structure and meaning of on-line texts. Considering a text mining application, a common chain of linguistic tools for NLP is presented. In subsection 1.2.2, Semantic Web standards and best practices to represent and share semantic meta-data are described. Linked Data, the most relevant initiative that promotes and regulates the publication of on-line semantic datasets is presented. Finally, subsection 1.2.3 discusses how NLP and the Semantic Web can cooperate to support Web data semantics.

1.2.1 Mining textual contents through Natural Language Processing

A considerable part of the information currently accessible on-line is expressed by means of natural language in the form of textual contents embedded in HTML or other kinds of mark-ups. As a consequence, in order to create semantic meta-data starting from Web contents, the exploitation of text mining techniques represents a valuable choice. These techniques

analyze textual contents and extract their semantic representations by automatically understanding their meaning. Thus users can search, browse, and integrate the information contained in the processed texts with greater ease.

In this context, Natural Language Processing approaches are often exploited in order to *make explicit the syntactic and semantic features of textual data by means of linguistic analysis* [6, 7]. These features can be directly exploited to improve the search and visualization of the informative contents of the considered texts. Otherwise, the outcome of the syntactic and semantic analysis can constitute the input of other text mining procedures useful to extract specific semantic representations of the information contained in the original texts.

The linguistic analysis of a text is typically structured in a *layered fashion*. Starting from the plain text, several linguistic tools sequentially analyze its contents. Each tool determines a set of descriptive features of the considered text, usually exploiting other descriptive features previously defined by the execution of other linguistic tools. Consequently, the whole process of linguistic analysis of a text can be thought as *the execution of a pipeline of linguistic tools*. In other words, by executing an ordered set of linguistic tools, the parts of a text (sentences, words, groups of words, etc.) can be recognized, classified, and interrelated. This process is called *text annotation*.

We present a brief overview of a set of common linguistic tools that could compose a linguistic pipeline. Then we describe a practical example of linguistic analysis of an English sentence by means of the considered set of linguistic tools (see Figure 3 and Figure 4).

1. Tokenizer (also known as lexical analyzer or word segmenter): starting from a plain text, identifies all the *tokens* (sequences of characters separated by a white spaces).
2. Paragraph and sentence splitter: sometimes relying on the results of the tokenizer, identifies the boundaries of *paragraphs*, and *sentences* inside each paragraph.
3. Part-of-speech tagger: labels each token identifying a word inside a sentence with the appropriate *part-of-speech* (POS) thus determining if it is a noun, a verb, an adverb, an adjective, etc. It is usually built exploiting the annotations produced by the tokenizer and the sentence splitter.

4. Lemmatizer: associates all the different inflected forms of a word to a single term usually referred to as the *root form* or *lemma*. This process is called stemming if it is carried out by a set of fairly simple heuristic rules. In addition proper vocabularies and some sort of morphological analysis can be used in order to lemmatize a word.

5. Named Entity recognizer: on the basis of the POS assigned to the words of the considered text, locates and classifies atomic elements into predefined categories (for instance persons, organizations, locations, expressions of time, quantities, etc). Named Entity recognizers usually rely on proper sets of rules as well as on list of words in order to identify and classifies Named Entities.

6. Parsers and grammar analyzer: usually relying on a set of grammar rules and on the POS assigned to the words of the considered text, a tree-based structure, called parse tree, is associated to each sentence so as to represent its grammatical organization. Linguistic parse trees can be structured on the basis of:

    a. phrase structure rules: when the words of each sentence are generally determined by a specific order. Words, characterized by their POS, are hierarchically grouped so as to identify the phrasal categories (Noun phrase, Verb phrase and Prepositional phrase) and the Determiners that will constitute the nodes of the parse tree.

    b. a dependency grammar: useful to represent the grammar of languages not characterized by a specific word order. In each sentence a head word is identified (usually the verb). The other words are interlinked to the head word or to each other by a defined set of relations referred o as functional dependencies (i.e. subject, object, complement, modifier, etc.). In this way the parse tree is determined.

7. Word Sense Disambiguator: determines the sense of the words inside a text by considering how each word is used in a particular context. There are many different approaches to perform Word Sense Disambiguation. Some of them exploit lower levels of text annotation like the information about the POS assigned to words.

Other examples of linguistic tools that can be included in a linguistic pipeline are the *Co-reference resolutor* that groups together inside a text, all the expressions that refer to the same thing (pronouns, abbreviations, acronyms, etc.) and the *Multi-Word Tagger* that detects and groups

together all the multiword expressions (i.e. code of conduct, flood alleviation, etc.).

The aforementioned linguistic tools often exploit some sort of *background knowledge* in order to support and improve the results of linguistic analysis tasks. Background knowledge can be constituted by collections of information describing specific linguistic features. This information is properly gathered and structured in order to enable or facilitate the mining process of linguistic tools. For instance lists of words, and dictionaries can be exploited as background knowledge by a lemmatizer or a Named Entity recognizer. Other knowledge resources such as semantic networks describing relevant linguistic features can be used to perform Word Sense Disambiguation.

Figure 3 shows a visual example of the annotation of the sentence "John eats his meal in the garden." by means of the set of linguistic tools previously described. Tokens are identified and the sentence boundaries are detected. The right POS is assigned to each token. The *Lemmatizer* identifies that the lemma or root form of the verb "eats" is "eat". The *Named Entity recognizer* identifies that the word "John" is a Named Entity represented by a person. Finally the *Word Sense Disambiguator* determines the sense of the words "eats", "meal" and "garden". The linguistic parse trees associated to this sentence are represented on the basis of both phrase structure rules (Figure 4a) and a dependency grammar (Figure 4b).

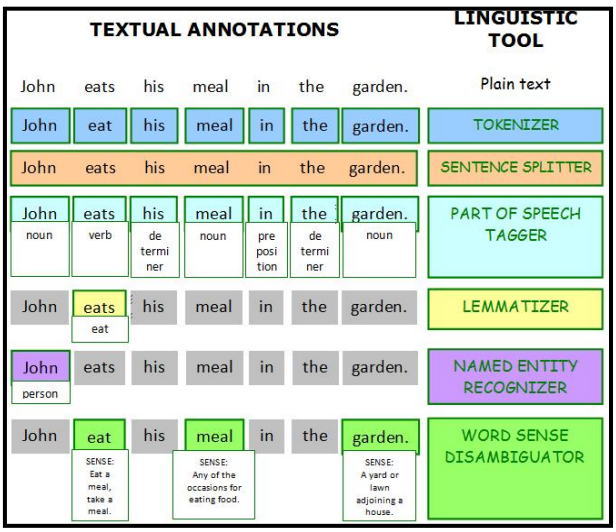


Figure 3: Example of linguistic analysis of a sentence by a pipeline of linguistic tools

Considering the same sentence, the execution of a *Co-reference resolutor* would determine that the pronoun "his" is referred to John or better that

both the words “John” and “his” refer to the same entity, thus have the same referent.

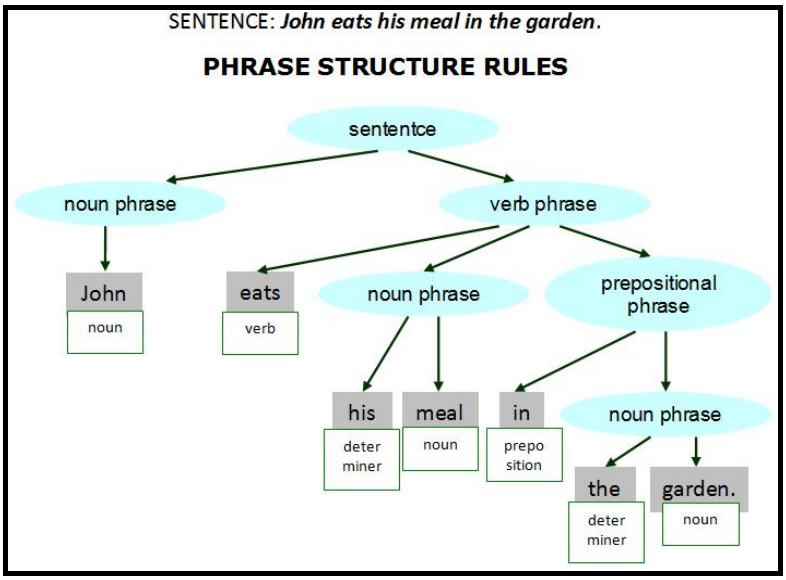


Figure 4a: Example of grammar analysis of a sentence based on phrase structure rules

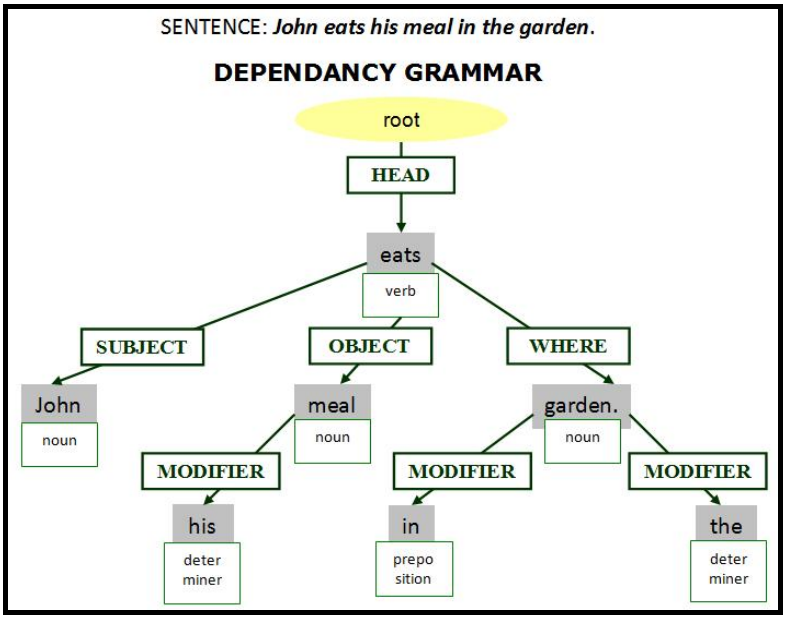


Figure 4b: Example of grammar analysis of a sentence based on dependency grammar

1.2.2 Semantically describing and interlinking data on a Web scale: the Semantic Web

To take advantage of the availability of semantic descriptions of Web contents, in addition to the creation of huge amounts of semantic meta-data, we need to define *how to represent these meta-data*, *how to publish them on-line*, and *how to interlink distinct semantic datasets so as to support automated data integration across multiple sources all over the Web*. All these issues are addressed by the Semantic Web ecosystem of standards and best practices that has been defined during the last decade under the guidance of the World Wide Web Consortium (W3C). This subsection offers an overview of the basic themes concerning the representation, publication and integration of on-line semantic datasets.

Two global issues need to be taken into account in order to realize the possibility to automatically semantically interpret and integrate contents on a Web scale:

- Since there are many distinct data sources all over the Web, exposing different kinds of contents, all of them *need to adopt the same shared identifier when they refer to the same entity in the semantic description of the contents they publish on-line, or at least they have to provide mappings between different identifiers pointing out the same entity*. Suppose there are two tourist information Web sites exposing semantic descriptions related to the city of Paris, the capital of France. If both Web sites reference Paris through the same shared identifier, a software agent can automatically retrieve, aggregate and relate the information from the two sources, 'understanding' that they both describe the same entity, Paris. This principle stands at the basis of what is usually referred to *as serendipity in semantic data integration over the Web* [8]. Therefore, it is possible to integrate and reuse the information contained in different heterogeneous on-line systems, devices and services without knowing anything about at design time, but only exploiting the support provided by the semantic descriptions of the exposed data. URIs are usually exploited as unambiguous entity identifiers to point out specific entities in the context of the whole Web.
- In order to fully exploit the advantages of the adoption of semantics to describe Web contents, it is important to *create semantic descriptions by referring to shared and formalized conceptualizations of the knowledge related to a specific domain of interest*: these conceptualizations are usually referred to as *ontologies*. Ontologies

describe a particular domain mainly in terms of the concepts that characterize it and the set of semantic relations among concepts by exploiting some formalism for instance the description logics. On the basis of the knowledge formalized in the ontologies it is possible to exploit automatic reasoning procedures over the semantic descriptions of Web contents. Thus for instance these descriptions can be verified with respect to the constraints imposed by the ontology or new information can be inferred from the one explicitly asserted in the ontology. Suppose we have an ontology describing animals and stating that *Siamese* and *Persian* are two kinds of *Cats*. Therefore both *Siamese* and *Persian* are two subclasses of the *Cats* class (see Figure 5). There is the LostPets Web site offering support to find cats and dogs that have been lost in London: the photos and the address of the proprietor of cats and dogs are published on-line so as to try to find them. In particular there are the photos of Fuffy, a Siamese cat and Miao a Persian one. If we need to search for all the lost cats, a semantic Web software agent will automatically infer from the knowledge formalized in the ontology that *Siamese* and *Persian* are kinds of *Cats* and thus both Fuffy and Miao will be included in the search results, even if their semantic descriptions state that they do not belong directly to the *Cats* class but to one of its subclasses. This is a simple example of how the knowledge formalized by means of an ontology can be exploited to semantically improve the understanding and retrieval of information.

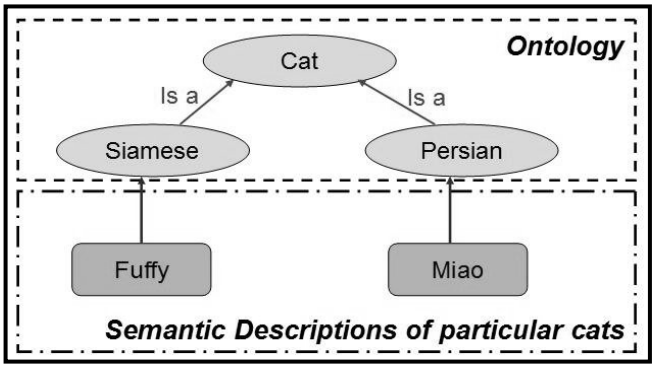


Figure 5: The animal ontology and the semantic description of particular cats

The rest of the subsection introduces RDF and OWL (the two main standard formats to represent and exchange semantic descriptions of contents) and the use of URIs to unambiguously identify entities over the Web. At the end of the subsection semantic search engines are introduced and Linked Data,

the most important initiative to foster the creation of semantic meta-data, is presented.

1.2.2.1 Data formats for semantic descriptions of on-line contents: RDF and OWL

A fundamental aspect that should be taken into account when dealing with semantic descriptions of Web contents is the adoption of standard formats to represent them. Data interoperability and integration on a semantic level is guaranteed:

- by exploiting *shared sets of URIs* to refer to entities that are common to distinct datasets or by adopting *shared ontologies* to semantically characterize Web contents exposed by distinct sources;
- by adopting *standardized data formats* that are useful to assure data interoperability on a syntactic level.

To this end during the last decade the W3C has developed and standardized the *Resource Description Framework*[11] (RDF). It constitutes the Semantic Web fundamental content representation meta-model. By adopting a triple-based model, it allows formulating semantic descriptions of Web contents. Each description is made of one or more machine-processable factual statements relating and describing URI-referenced entities. Each RDF statement, called also RDF triple, is usually made of three parts: a *subject* identifying a specific entity to describe, a *property* usually formalized inside an ontology, pointing out a specific feature of the subject of the RDF triple and an *object* containing the value of the property. For instance, a RDF statement can issue that 'Paris is located in France'. In this case Paris is the subject of the RDF triple, the property is "to be located in" and the object is France (see Figure 6). The greatest part of the Semantic Web data currently exposed on-line is represented by exploiting the RDF data model and is constituted by sets of RDF triples.

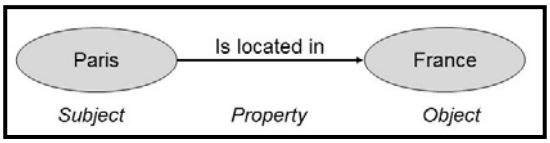


Figure 6: An example of a RDF statement/triple

In order to represent ontologies and thus to describe the structure and the properties of the entities concerning a specific domain, the W3C has

[11] Resource Description Framework W3C - http://www.w3.org/RDF/

standardized, in parallel to RDF, the *RDF Schema*[12] (RDFS). RDFS is a knowledge description language characterized by very elementary constructs and thus by a restricted set of expressive possibilities to describe a particular domain of interest. In RDFS it is possible to define hierarchies of classes together with properties that can characterize or relate these classes.

Since the expressive power of RDFS has proven insufficient in many contexts where more detailed descriptions of the conceptual structure of the domain of interest are necessary, in February 2004 the W3C issued the Recommendation of the *Web Ontology Language*[13] (OWL). Mainly based on description logics, OWL adds several expressive possibilities to RDFS so as to strongly structure ontologies. For example, OWL allows the definition of existential, union, disjunction and cardinality constraints among the classes of an ontology. Since the standardization of OWL, the need to extend the Web Ontology Language so as to add new expressive features that have been requested by users dealing with it has increased. New improved reasoning algorithms that can exploit these new features have experimented growing diffusion. As a consequence a new revised version of OWL, referred to as OWL 2[14], was defined and published as a W3C Recommendation in the last quarter of 2009. Summarizing, RDFS and OWL, even if with different expressive possibilities, both represent *means to define a reference model in order to express factual RDF statements*. More details about ontologies and OWL are explained in subsection 1.3.3.

Both the factual statements represented through sets of RDF triples and the description of ontologies expressed by the constructs of OWL can be serialized and exchanged by a specific XML syntax: indeed the W3C has defined, together with the standardization of RDF and OWL, proper XML serialization formats. RDF also has other compact notations different from XML, like *N-triple*[15] and *Turtle*[16].

The RDF semantic descriptions of Web contents are made of sets of RDF statements and constitute resources different from the related HTML pages that include the same information represented in a human-readable way. When a semantic software agent accesses the contents of a Web site it needs to retrieve their semantic descriptions to be able to automatically

[12] RDF Schema W3C - http://www.w3.org/TR/rdf-schema/
[13] Web Ontology Language W3C - http://www.w3.org/TR/owl-features/
[14] Web Ontolgy Language 2 W3C - http://www.w3.org/TR/owl2-primer/
[15] N-triple - http://www.dajobe.org/2001/06/ntriples/
[16] Turtle - http://www.dajobe.org/2004/01/turtle/

interpret them. Thus the same informative contents can be expressed in HTML and as a set of RDF triples. In order to avoid duplicated output formats for the same contents, different solutions have been proposed. The *RDF in attributes*[17] (RDFa), discussed in the context of the W3C Semantic Web Deployment Working Group, defines a set of XML attributes in order to extend XHTML so as to directly embed in it semantic metadata. Proper procedures are defined to enable software agents to automatically extract the related set of RDF triples from the extended XHTML syntax.

Other proposals to avoid the duplication of data formats are the *embedded RDF*[18] (eRDF) similar to RDFa and the *Gleaning Resource Descriptions from Dialects of Languages*[19] methodology (GRDLL) issued as a W3C Recommendation at the end of 2007. It defines a mark-up to specify that an XHTML document, or more in general a XML document contains information that can be represented as a set of RDF triples referring also an algorithm, typically constituted by an XSLT, to extract the RDF triples. In the definition of the HTML 5 the W3C is dealing with Microdata[20], a new flexible standardized way to incorporate semantic meta-data directly into HTML pages.

In order to issue particular queries over sets of RDF triples the W3C has developed *SPARQL*[21], the Query Language for RDF that has been standardized as a set of Recommendations at the beginning of 2008. Thanks to SPARQL proper generic patterns can be defined so as to search for matching sets of RDF triples also over distributed on-line RDF datasets. Currently, the SPARQL W3C Working Group is concerned with a new revised version of SPARQL[22], referred to as SPARQL 1.1.

1.2.2.2 The URI system: unambiguously refer and retrieve (semantic) descriptions of informative and non-informative resources all over the Web

In order to create and expose over the Web semantic descriptions of contents it is fundamental to unambiguously identify the entities that are involved. URIs are usually exploited as identifier because they are globally

[17] RDFa W3C - http://www.w3.org/TR/xhtml-rdfa-primer/
[18] eRDF - http://research.talis.com/2005/erdf/wiki/Main/RdfInHtml
[19] GRDDL W3C - http://www.w3.org/TR/2007/REC-grddl-20070911/
[20] HTML5 Microdata - http://www.whatwg.org/specs/web-apps/current-work/multipage/links.html#microdata
[21] SPARQL W3C - http://www.w3.org/TR/rdf-sparql-query/
[22] SPARQL 1.1 W3C - http://www.w3.org/TR/sparql11-query/

unique names without centralized management and they constitute the preferred mean to access information about a resource over the Web [9].

Usually a URI is exploited over the Web to access, or better to require a specific representation of an *information resource* that is constituted by a stream of bytes structured with respect to a particular data formats. For instance an information resource can be represented by a HTML document, a RDF set of triples serialized in XML, but it could be also an image or another multimedia file.

Besides information resources, in order to structure semantic descriptions of data we need to refer also to *non-information resources* that are real-world objects not directly accessible over the Web. Like information resources they can be unequivocally identified by a URI but in this case they are not directly connected to a specific representation accessible over the Web in terms of a stream of bytes. We can state that they are not directly dereferencable.

For instance if we consider the city of Paris we can identify Web URIs used to access information resources and thus to retrieve a specific one of their representation. For instance the URI *http://dbpedia.org/page/Paris* points out and is useful to retrieve a HTML document describing the city of Paris or the URI *http://dbpedia.org/data/Paris* points out and is useful to retrieve a RDF document serialized in XML containing a set of triples describing the city of Paris (see Figure 7). Both these URIs are exposed by DBpedia[23] [10], a collection of semantic information mined from Wikipedia.

If we want to unambiguously refer to Paris as the subject of an RDF triple describing some information concerning the capital of France, we are considering a real-world entity, a non-information resource. Also in this case we can point it out through a unique URI, *http://dbpedia.org/resource/Paris* but it will be not directly dereferencable and thus not directly connected to a specific representation.

Thus, summarizing, there are two kinds of URIs: those that identify *information resources* usually represented by a document over the Web and those that identify *non-information resources* that point out real-world objects.

In order to expose on-line semantic description of data, it is fundamental to establish, *given a URI, a global mechanism to find out what kind of resource it identifies*.

[23] Dbpedia - http://dbpedia.org/About

The W3C has defined guidelines to deal with both kinds of URIs[24]. The Web is used as a look-up mechanism with respect to these guidelines: by accessing on-line a URI it is possible to understand the kind of resource it identifies. When a *URI pointing out an information resource* is accessed, a document representing the current state of the resource is generated and sent to the client.

Figure 7: Two different representations of the information concerning the city of Paris in DBpedia, identified by two different information resources URIs

Two different techniques can be exploited to *mint the URIs of non-information resources*. These techniques, together with the HTTP protocol

[24] Cool URIs for the Semantic Web W3C - http://www.w3.org/TR/cooluris/

enable clients to get a Web document describing the real-world object referred by the non-information resource URI:

- *hash URIs*: all non-information URIs contain a fragment portion, separated by the rest of the URI through the hash symbol (*#*) in order to identify a particular entity. For instance, the URI *http://www.bestcities.com/france* can point out a document containing a set of RDF triples describing three famous tourist cities in France: Paris, Nantes and Grenoble. We can structure the non-information resource URIs identifying the cities by adding specific hash parts:

  http://www.bestcities.com/france#Paris,
  http://www.bestcities.com/france@#Nantes,
  http://www.bestcities.com/france#Grenoble.

- These three new URIs can be used as city identifier in the RDF triples of the document published at *http://www.bestcities.com/france*. When we access over the Web one of these three new URIs, the hash part is stripped out as recommended in the HTTP protocol and the RDF document *http://www.bestcities.com/france* describing Paris, Nantes and Grenoble is retrieved.

- *URI forwarding*: all non-information resource URIs are minted freely. When a non-information URI pointing out a specific real-world object is accessed on-line, the server replies with a redirection to a URI of an information resource represented by a document describing the real-world object. For instance, the non-information resource URI *http://dbpedia.org/resource/Paris* can be adopted to refer to the city of Paris. When this URI is accessed, the server will redirect us to the URI *http://dbpedia.org/page/Paris* that points out and is useful to retrieve a HTML document describing the city of Paris.

In both cases, the *HTTP content negotiation mechanism* can be exploited as a possible way to let the client specify what kind of representation of a specific information resource is needed. If we consider a Semantic Web browser (client), probably it would ask for a RDF representation of a document in terms of a set of triples rather than a human readable HTML one.

Regardless of the choice of hash URIs or the adoption of the URI forwarding mechanism, there are some general rules to take into consideration when we have to choose URIs for non-information resources[25]. First of all URIs

[25] Linked Data Tutorial - http://www4.wiwiss.fu-berlin.de/bizer/pub/LinkedDataTutorial/

have to be *under a HTTP name-space under our control*. They should be *short*, *readable* and *not implementation-dependant*. For instance *http://mysite.com/home.php#me* is not implementation-independent since with great probability it says that the document accessed is generated by some sort of PHP script. They should be *stable* and *persistent*; the URI exploited to identify a real-world object should possibly not change with time.

Summarizing, non-information resource URIs are exploited over the Web in order to unambiguously reference entities that can be real-world objects like cities (Paris, Pisa, etc.), persons (Alice, Bob, etc.), animals (Fuffy, Miao, etc.). Non-information resource URIs are also exploited to point out the concepts of an ontology, referred also to as classes, as well as to unambiguously identify the properties that relate these classes.

When semantic descriptions of Web contents are created, *different URIs can be exploited to identify the same real-world object thus the same non-information resource*: these URIs are referred to as *alias*.

For instance the following two URIs: *http://dbpedia.org/resource/Berlin* and *http://sws.geonames.org/2950159/* both refer to the city of Berlin in the semantic metadata exposed on-line respectively by *DBpedia* and *Geonames*[26], an extensive geographic database. Both *DBpedia* and *Geonames* provide different and complementary information related to Berlin.

In order to successfully merge and integrate the data by means of a semantic Web software agent, it should be explicitly stated that both the two URIs refer to the same city. The most common way to provide this kind of equivalence mappings between couples of URIs is through proper RDF triples by exploiting the OWL sameAS property identified by the hash URI *http://www.w3.org/2002/07/owl#sameAs* (see Figure 8).

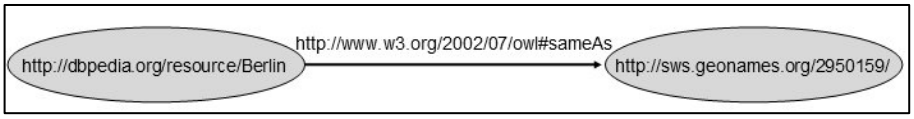


Figure 8: Mapping two equivalent non-informative resources URIs through the OWL sameAs property

In order to easily allow as much integration as possible between semantic descriptions of Web contents exposed by different sources it is fundamental for each source:

[26] Geonames - http://www.geonames.org/

- to *provide mappings of its URIs to other URI aliases defined by different sources*;
- to *reuse stable URIs defined under other Web domains to identify the real-world entities referenced in the semantic descriptions*. For instance in the previous example, it could have been exploited the *DBpedia* URI *http://dbpedia.org/resource/Berlin* in order to refer to the city of Berlin inside the semantic datasets of *Geonames*.

In the same way, to define new ontologies, it is particularly relevant to *try to reuse as much as possible non-information URIs pointing out classes and properties already defined by other ontologies under different Web domains*. Therefore it will be easier to integrate the information described by different ontologies without having to formalize again ontological classes or properties as well as without having to deal with a considerable number of mappings of equivalent URIs.

1.2.2.3 Semantic Web Search Engines: searching for semantic data over the Web

The semantics of Web contents is represented and exposed over the Web by means of sets of RDF statements as well as through ontologies mainly formalized by exploiting OWL. In order to carry out searches inside the semantic data published over the Web, during the last few years several search engines for semantic contents have been proposed. They crawl and index semantic Web documents, both ontologies and collections of RDF triples. Starting from keywords, they allow searching for URIs identifying real-world objects, classes and properties of an ontology or semantic Web documents usually composed by sets of RDF statements. Different parameters can be specified in order to customize the searches. Three examples of Semantic Web search engines - *Watson*, *Sindice* and *Falcons* - are presented below.

Watson[27], developed by the Knowledge Media Institute (KMi) of the Open University in Milton Keynes in the UK, allows searching for the URI of ontological classes, properties and individuals inside the set of indexed semantic documents. The URI as well as a set of descriptive meta-data are provided for each document.

Sindice[28], developed jointly by the Digital Enterprise Research Institute of Galway in Ireland, Fondazione Bruno Kassler and Open Link Software,

[27] Watson Semantic Web Search Engine - http://kmi-web05.open.ac.uk/WatsonWUI/
[28] Sindice Semantic Web Search Engine - http://sindice.com/

crawls and indexes semantic Web data represented as documents containing sets of RDF triples but also as HTML documents including semantic meta-data by exploiting RDFa or Microformats. Users can search for URIs of classes, properties and individuals starting from keywords: for each result the originating Semantic Web document is cached and can be visualized as a list of triples or as a graph. The list of ontologies referenced the document is accessible as well.

Falcons[29], developed by the Institute of Web Science (IWS) of the Southeast University in China, is another relevant example of semantic Web search engine. Its index in September 2009 included almost 23 million semantic descriptions of Web contents represented in RDF and serialized in XML and almost 13 thousand ontologies.

Watson, *Sindice* and *Falcons* provide a set of Web API in order to easily integrate their search functionalities to support other applications based on the semantic data published over the Web.

Search engines for semantic data are particularly useful to retrieve widely adopted URIs used to refer classes, properties or individuals. This information can be reused to create on-line semantic descriptions of data

1.2.2.4 Linked Data: a Web of interlinked distributed semantic datasets

Linked Data[30] currently represents the most relevant initiative aiming at the practical building of a Web of Data with respect to the Semantic Web vision [11]. Linked Data is devoted to promote the creation and on-line publication of interlinked semantic datasets. Linked Data also defines a set of best practices to expose, share and connect pieces of data, information, and knowledge over the Web, like for instance the best practices to manage Semantic Web URIs. Many public and private institutions, initiatives and research projects have exposed on-line their data as Linked Data semantic datasets, thus contributing to the creation of the Web of Data. The core hub of this net of interconnected data is constituted by DBpedia [12]: it is a semantic representation of the structured information contained in Wikipedia in terms of RDF triples.

The number of on-line datasets available as Linked Data, globally referred to as the Linked Data cloud, has considerably grown during the last few years, turning the Linked Data initiative into the catalyst of the creation of

[29] Falcons Semantic Web Search Engine - http://iws.seu.edu.cn/services/falcons/conceptsearch/index.jsp
[30] Linked Data - http://linkeddata.org/

the largest repository of free on-line structured and interconnected contents. In May 2007 the Linked Data cloud was made of about 1 billion of RDF triples and there were about 120.000 links between 12 RDF datasources. By September 2010 (see Figure 9) this had grown to 25 billion of RDF triples coming from 203 datasources, interlinked by around 395 million RDF links.

Considering the growing number of initiatives and projects involving Linked Data datasets or exposing their contents as Linked Data, this trend is expected to be kept during the next years, thus increasing more and more the usefulness and the possibilities of exploitation of this enormous amount of interlinked semantic information.

Several tools to support the editing, browsing, and storage of Linked Data RDF datasets are available[31]. Methodologies and techniques to store, distribute, and aggregate Linked Data currently represent an active research field. Also techniques to browse and visualize this huge amount of semantic information are currently investigated so as to make it easily accessible and exploitable by common Web users.

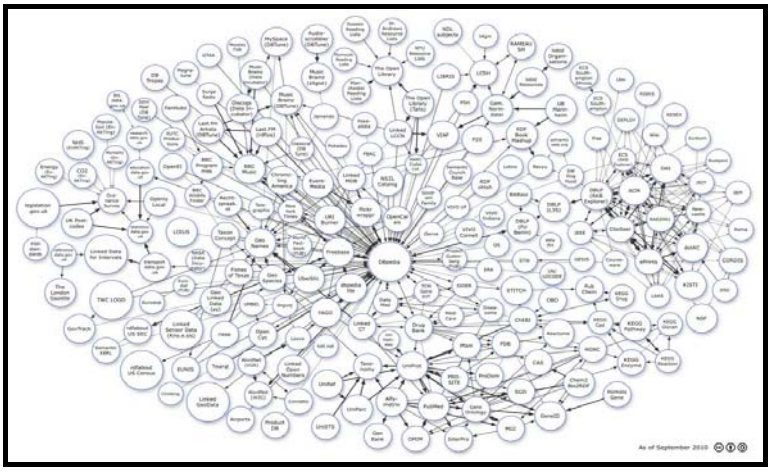


Figure 9: The LinkedData cloud as of September 22, 2010 - Authors: Anjeve, Richard Cyganiak

1.2.3 Natural Language Processing underpins the Semantic Web

Natural language processing has proven to be a valid support to automatically add machine processable semantics to Web contents and, more in general, to ease several other activities connected to the semantic management of information on a Web scale [13, 14].

[31] Linked Data tools - http://linkeddata.org/tools

In order to foster the creation of a considerable amount of semantic meta-data linked to relevant on-line semantic datasets, during the last few years text mining techniques have been adapted and applied to Web contents. In particular, in the context of the LinkedData Project, DBpedia currently constitutes the most relevant collection of semantic descriptions of real-world objects, widely exploited and referred so as to semantically characterize Web contents.

Procedures for keyword extraction, Named Entity recognition, but also more advanced Natural Language Processing techniques have been often used to mine Web documents. To produce semantic meta-data that are linked to other relevant semantic datasets, terms from these documents have been often disambiguated thanks to the association of the URI of the referred DBpedia real-world objects usually described by a page of Wikipedia. Open Calais [15], for example, parses documents and points out entities, facts and events. When possible, entities are linked to relevant semantic datasets like DBpedia, Freebase, and GeoNames. Wikify [16] performs keyword extraction from Web pages, and disambiguates mined terms linking each of them to the referred Wikipedia entity.

Natural Language Processing approaches have been also applied in ontology editing by means of controlled languages or also to automatically enrich or populate an ontology with new instances by mining textual contents.

1.3 KNOWLEDGE RESOURCES: BACKGROUND KNOWLEDGE TO SUPPORT WEB DATA SEMANTICS

The mining of textual contents by means of NLP techniques so as to create semantic-metadata as well as the representation and integration of these meta-data on a Web scale may rely on some kind of background knowledge. In this context, background knowledge would be *any kind of resource that is properly created, structured and exploited so as to enable or facilitate the execution of one or more tasks related to knowledge-based data management*. Knowledge resources represent a kind of background knowledge since they organize mankind knowledge in order to support a wide range of tasks like automated content analysis, data management, knowledge representation, and information retrieval. Knowledge resources are also referred to as Knowledge Organization Systems (KOS) because they aim at organizing information by making explicit relevant features of the underlying semantic structure. Each knowledge resource can provide information related to a general or a specific domain (i.e. environment, biology, genomics, etc.) with a proper level of data structuring.

Subsection 1.3.1 offers an overview of the possibility to organize and structure information to provide background knowledge by introducing different kinds of knowledge resources. Two of them, lexicons and ontologies, are described in detail in subsection 1.3.2 and 1.3.3 respectively. By contrast in Section 1.3.4 the differences and the complementary aspects of lexicons and ontologies are discussed, and the possibility to unify or interconnect both knowledge resources is considered.

1.3.1 Taxonomy of knowledge resources

Knowledge resources also referred to as Knowledge Organization Systems (KOS), include *all the types of schemes for organizing knowledge and promoting knowledge management* [17, 18]. The structure and kind of the information represented by means of a knowledge resource may differ depending on both the specific purpose of the knowledge resource and its exploitation context. Taking into consideration the level of knowledge structure, Knowledge Organization Systems can be grouped into three broad categories (see Figure 10):

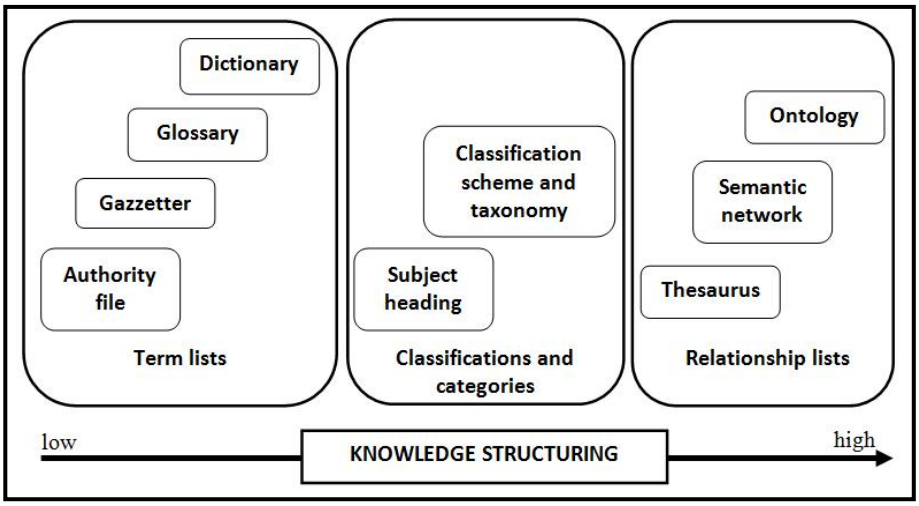


Figure 10: Taxonomy of knowledge resources

- Term lists: lists of words and phrases often better specified by means of a definition. Some examples are:
  - *authority file*: list of terms that are exploited to control several alternate names that could be given to an entity or to specify the domain value for a particular field (i.e. list of countries, individuals, organizations, etc.).

  - *glossary*: list of terms together with their definitions, usually concerning a specific domain of interest.
  - *gazzetter*: a geographical reference, used to find information about places and place names usually associated with other descriptive data like geographical coordinates.
  - *dictionary*: alphabetical list of words with definitions with a more general scope, and providing more information about every term than a glossary

- Classifications and categories: schemata that organize knowledge on the basis of a predefined set of subjects or topics sometimes arranged in a hierarchical structure. Some examples are:
  - *subject heading*: a controlled vocabulary made of terms useful to represent subjects in a specific domain with the purpose of indexing contents.
  - *classification scheme or* taxonomy: a classification of entities related to a specific domain, arranged in a hierarchical structure. It is typically organized by super-type/sub-type relationships, also called generalization-specialization relationships.
- Relationship lists: representations of relationships between terms and the concept they represent. Some examples are:
  - *thesaurus*: a collection of words interlinked mainly by four kinds of relationships: broader term, narrower term, synonym and related term.
  - *semantic network*: exploited in the context of Natural Language Processing, is a collection of concepts and terms structured as a network. Concepts can be related by means of a set of semantic relations more complex than a thesaurus, like part-whole, cause-effect and so on. Computational lexicons like WordNet are examples of semantic networks.
  - *ontology*: a conceptual model describing a specific domain of interest useful to represent complex relationships among the considered objects, including rules and axioms missing from semantic networks. Ontological descriptions stand at the basis of the paradigm to represent semantic meta-data over the Web, representing a core element of the Semantic Web.

In this thesis, focus is put on computational lexicons that represent a particular kind of semantic networks and ontologies because they constitute knowledge resources of great relevance to automate the creation and interoperability of semantic meta-data over the Web.

1.3.2 Computational lexicons: mining semantics from texts

Computational lexicons represent a particular kind of knowledge resources *modelling the features of a language in order to support the linguistic analysis of textual contents*. Computational lexicons are usually structured as semantic networks and thus constituted by a set of concepts interconnected by means of semantic relations. Computational lexicons are mainly exploited for the automated understanding of the meaning of texts by means of Natural Language Processing techniques. For instance Word Sense Disambiguation procedures can leverage computational lexicons in order to understand the meaning of a word in a specific context by automatically choosing the most appropriate sense.

There are many models of computational lexicon that formalize distinct features of a language. Every model is usually identified by the name of a particular lexicon compliant to that model. Two relevant examples of models of computational lexicons are:

WordNet model, organized as a semantic network where each node constitutes a meaning identified by a set of synonyms and each arc represents a semantic relation connecting a pair of meanings. The WordNet model is described in greater detail below.

FrameNet model, structured as a collection of semantic frames [19]. Each semantic frame describes a specific real-world situation or context by means of a set of words representing relevant concepts as well as by defining the semantic roles characterizing the entities involved in that context. Thus a semantic frame may be figured out as a coherent structure composed of a set of related concepts that are interlinked such that without knowledge of all of them, one does not have complete knowledge of any one[32].

For instance, a semantic frame characterizing a *commerce scenario* would include the following set of lexical units: *buyer*, *commerce*, *cost*, *goods*, *price*, *purchaser*, *retailer*, *seller*, and *vendor*. Each lexical unit represents a specific meaning of a word in the semantic frame scenario. The same frame would be also described by the following set of semantic roles: a *buyer*, *goods* that are sold, *money* useful to buy goods, and a *seller*. When a text is

[32] Frame Semantics, Wikipedia - http://en.wikipedia.org/wiki/Frame_semantics_%28linguistics%29

analyzed, once identified one or more semantic frames that can be exploited to interpret the meaning of a sentence, the words of that sentence are described by means of the semantic roles of the considered frame. In this way the semantics of all the entities taking part in the real-world scenario described by the frame can be specified.

The FrameNet model of computational lexicons derives his name from the FrameNet project[33], started in 1998 at the International Computer Science Institute (ICSI) in Berkeley. The FrameNet project is devoted to create knowledge resources made of collections of semantic frames. The English FrameNet knowledge resource developed and maintained by the ICSI includes more than 1000 semantic frames representative of a wide range of semantic domains. By adopting the FrameNet model, a considerable number of knowledge resources for languages other than English have been created so as to support frame-based linguistic analysis of textual contents in multiple languages.

Summarizing, computational lexicons constitute a particular typology of knowledge resource. They provide linguistic tools with the background knowledge useful to support the automated understanding of the meaning of a text. Each computational lexicon usually describes the features of a specific language. Textual contents in different languages can be mined if the computational lexicons describing the considered languages are available.

A collection of texts spread across different languages can be semantically described by relying on the related set of computational lexicons. The informative contents of these texts can be represented by means of semantic meta-data and thus made accessible across multiple languages. This approach to deal with multilingual textual information is referred to as *cross-lingual text mining*. Cross-lingual analysis of textual contents is becoming more and more relevant since the information currently available over the Web is becoming more and more constituted by multilingual contents.

The remaining part of this subsection introduces WordNet, one of the most used computational lexicons. There is a detailed description of WordNet data model followed by a discussion about issues related to the interoperability across WordNet of different languages

[33] FrameNet - http://framenet.icsi.berkeley.edu/

1.3.2.1 WordNet

Currently the most relevant and exploited computational lexicon is WordNet[34]. It is a lexical reference system that explicitly represents many different characteristics of the human linguistic knowledge. It was conceived in 1985 by a group of research of the Princeton University, on the basis of psycholinguistic theories concerning human memory. Since then its contents, terms coverage, and relations have been continuously enriched. The structure of the language representation model has also been improved and better defined [20].

WordNet language structuring is based on the fundamental distinction between:

- *lexical form*, the way used to represent a single word as a sequence of characters (string);
- *meaning*, that is a specific concept; it can be identified by means of one or more different lexical forms.

The many-to-many relations between meanings and lexical forms can be represented by a *lexical matrix*, a sort of table in which every row corresponds to a particular meaning and every column to a specific lexical form. If a lexical form represents different meanings, it is a *polysemous* lexicon. For example the lexical form 'car' can represent two different meanings: a four wheels vehicle and the machine where passengers ride up and down. On the other end, every meaning can be expressed through different lexical forms that are called *synonyms*. For instance, the lexical forms 'machine' and 'car' can both refer to a four wheels vehicle.

In Figure 11 an example of lexical matrix is represented, underlying the occurrence of synonymy and polysemy.

[34] WordNet - http://wordnet.princeton.edu/

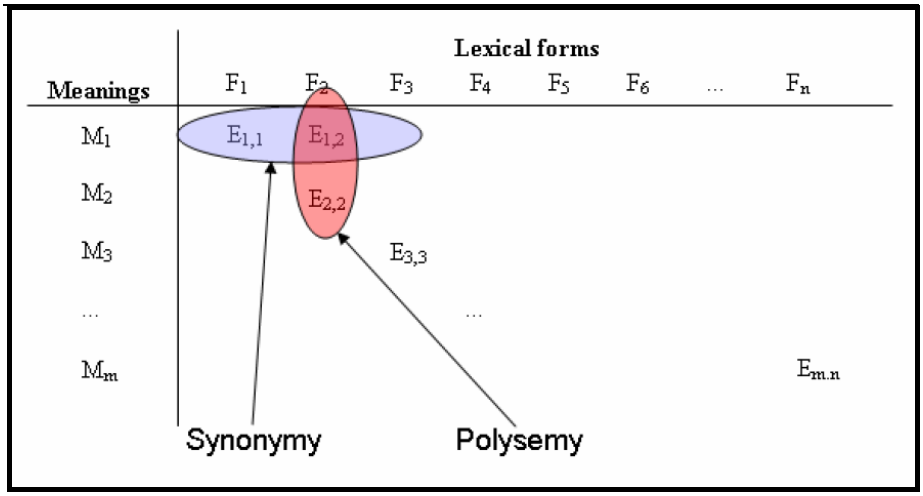


Figure 11: WordNet lexical matrix: meanings and lexical forms

Starting from the lexical matrix representation we can describe the basilar entity that constitutes the core of WordNet: the *synset*. A synset represents a specific meaning or concept and is identified by the set of synonym lexical forms used to refer to that particular meaning. For example the lexical forms 'car', 'auto', 'automobile', 'machine' and 'motorcar' constitute the synset that define the concept of four wheels vehicle.

Intersecting a meaning and a lexical form a *WordSense* is identified. A WordSense is the association of a lexical form to a particular meaning identified by a synset. Thus a WordSense determines one of the different concepts referable using that lexical form. Every element of the lexical matrix in Figure 11 represents a WordSense.

Some considerations can be made from abovementioned features of WordNet:

- the number of WordSenses generated by a synset is equal to the number of lexical forms covered by the synset;
- every WordSense is associated exactly to a single synset;
- every WordSense includes a single lexical form;
- every lexical form can belong to one or more WordSenses and thus can be associated to one or more synsets.

In Figure 12 there is a graphical representation of the cardinality of the relation between synsets, WordSenses, and lexical forms.

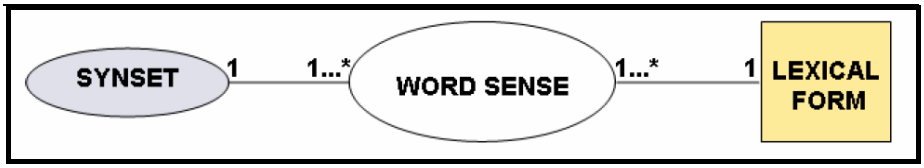


Figure 12: Cardinality of the relations between synsets, WordSenses, and lexical forms

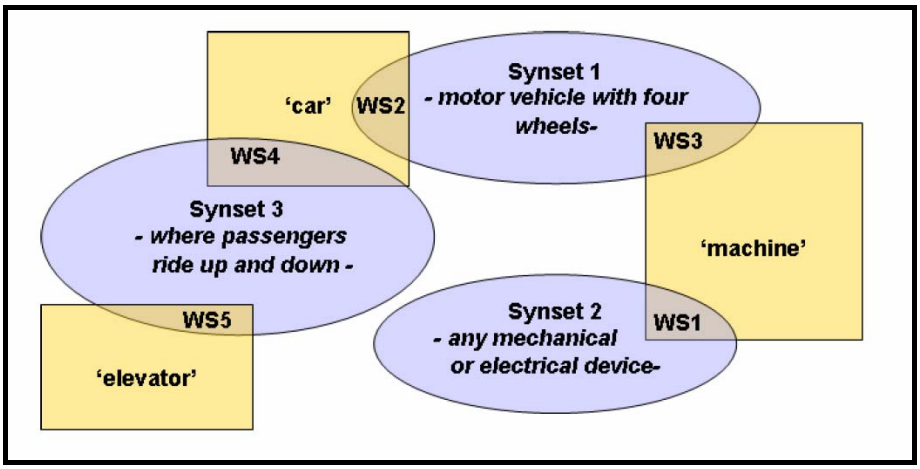


Figure 13: Examples of different synsets sharing lexical forms

Figure 13 shows an example of different synsets sharing lexical forms. Each synset is identified by the description of its meaning, called *gloss*. The abbreviation WS stands for WordSense. Every WordSense represents the intersection between a lexical form (rectangle) and a meaning (circle).

WordNet considers four parts of speech: nouns, verbs, adjectives and adverbs. Every synset is associated to a particular part of speech.

Two different sets of relations have been defined in order to represent the associations that characterize WordNet:

- *lexical relations*, between two or more WordSenses; Among the lexical relations relevant examples are:
  - *Synonymy*, connecting all the WordSenses that refer to the same meaning thus constituting a synset;
  - *Antinomy*, connecting two WordSenses referring to opposite meanings, for instance 'natural object' and 'artefact';
  - *SeeAlso*, linking a WordSense to one or more other WordSenses that can provide further descriptive information. For example the verb 'breathe' can be connected to the verbs 'breath in' and 'breath out';

- *semantic relations*, between two synsets. Some significant semantic relations are:
  - *Hypernymy / hyponymy*, respectively representing relations of generalization / specialization between synsets. For instance the concept of 'station wagon' is a specialization or an hyponym of the concept of 'car' and, inversely, the concept of 'car' is a generalization or an hypernym of the concept of 'station wagon';
  - *Meronymy / holonymy*, used to represent part-whole associations between concepts. A 'cell' is a part or meronym of an 'organism' and, inversely, an 'organism' is an holonym of a 'cell' meaning that an 'organism' is composed by cells;
  - *Entailment*, linking two verbal synsets. The former implies the latter and the latter can't be executed if the former is not. For instance the verb 'walk' entails the verb 'step';
  - *Attribute*, linking a nominal synset with one or more adjectival synsets that express possible characteristics of that name. The name 'measure' can be connected with the adjectives 'standard' and 'non standard';
  - *Similarity*, linking two adjectival synsets with similar meaning. For instance 'wet' with 'moist' or 'dry' with 'arid';

Every relation can be symmetric and/or transitive and can be characterized by restrictions regarding the parts of speech that it connects.

At present, the Princeton English WordNet version 3.0 (see Table 1) includes 117659 concepts (or distinct synsets).

POS	Words	Synsets	Word-sense pairs
Noun	117798	82115	146312
Verb	11529	13767	25047
Adjective	21479	18156	30002
Adverb	4481	3621	5580
Total	155287	117659	206941

Table 1: Number of Words, Senses, and WordSenses by POS in the Princeton English WordNet 3.0

There are several interfaces to query WordNet and different available data formats to export its contents. WordNet can be accessed by means of standalone desktop applications thanks to an appropriate graphical

browsing interface. WordNet can be also queried directly over the Web exploiting an HTML form-based interface[35].

WordNet data collection is available to download as a Prolog database (referred to as Prolog distribution) properly structured and organized across different files. The WordNet SQL Builder[36] is a Java utility to generate a SQL database directly from the Prolog files.

The Web Ontology Language (OWL) constitutes a convenient data model to formalize the representation of networks of semantic meta-data as RDF datasets. Since WordNet is a semantic network composed of a collection of synsets interconnected by different kinds of relations, it can be conveniently represented by means of a RDF graph. Moreover representing WordNet in OWL/RDF can ease its exploitation and integration with other semantic datasets, enabling greater interoperability and making no assumptions about a particular application domain. Because of all these considerations, in 2006 the WordNet Task Force of the W3C Semantic Web Best Practices Working Group published a Working Draft titled RDF/OWL Representation of WordNet[37]. This Working Draft describes how to represent WordNet as a RDF dataset. A proper OWL ontology has been defined in order to formally specify the general structure of WordNet, or better to formalize its data model. An URI assignment policy has been described so as to unambiguously identify all the entities constituting a WordNet of a specific language by means of URIs. The English Princeton WordNet 2.0 has been published on-line as an RDF dataset[38]. In 2010, the Computer Science Department of the Vrije University of Amsterdam published on-line the RDF dataset related to the version 3.0 of the English Princeton WordNet[39], together with the list of mappings between synsets of the version 2.0 and synsets of the version 3.0. Both the RDF/OWL representation of WordNet created by the W3C and the most recent one created by the Vrije University of Amsterdam are integrated with the semantic datasets of the Linked Data initiative. WordNet synsets are directly or indirectly mapped to the concepts of DBpedia that represents the core semantic reference of the Web of Data.

On the basis of the WordNet data model several other semantic resources have been created from scratch or extending and enriching existing WordNets. BabelNet [21] is an example of semantic network sharing the

[35] WordNet Web Browser - http://wordnetweb.princeton.edu/perl/webwn
[36] WordNet SQL Builder Web Site - http://wnsqlbuilder.sourceforge.net/
[37] W3C Working Draft RDF/OWL Representation of WordNet - http://www.w3.org/TR/wordnet-rdf/
[38] Princeton WordNet 2.0 in RDF - http://www.w3.org/2006/03/wn/wn20/
[39] Princeton WordNet 3.0 in RDF - http://semanticweb.cs.vu.nl/lod/wn30/

same data model of WordNet. BabelNet has enriched the version 3.0 of the English Princeton WordNet with new synsets and multilingual lemma by means of the analysis of Wikipedia contents as well as by exploiting external corpora and online translation services like Google Translate. BabelNet contains almost 3 million concepts and an average of 6,7 lemma of different languages associated to each concept.

The traditional applications of WordNet are mainly related to support several kinds of automated linguistic analysis [22, 23]. Word Sense Disambiguation can be enabled thanks to the huge amount of lexical relations included in WordNet. Also information extraction procedures, related to the automatic identification of selected types of entities, relations, or events in free texts can benefit from the huge amount of interconnected data that WordNet provides. In automated question answering WordNet lexical contents are usually exploited to interpret the meaning of a user defined question so as to determine how to find the best-matching answer. Many processes of automatic characterization and indexing of textual contents like documents or Web pages use WordNet relations to evaluate the similarity between different texts. In this way texts can be also clustered so as to simplify their retrieval procedures.

WordNet and the management of multilingual contents

WordNet data model has been adopted as a reference to create computational lexicons of different languages. During the last twenty years several WordNets for languages other than English have been built. Currently the Global WordNet Association[40], constituted to foster interoperability between WordNets of distinct languages, comprehends 64 WordNets covering 51 languages [24].

In this scenario the management of the interoperability between WordNets of different languages constitutes a relevant issue so as to enable their exploitation to perform cross-lingual text mining tasks. In 1996 the EuroWordNet project[41] started. EuroWordNet produced a collection of mapping data between WordNets of different languages. Each mapping is obtained through the Inter-Lingual Index (ILI). The ILI is a collection of many entries. Each entry identifies a cross-lingual concept by means of a short definition and the reference to the corresponding synset in the English version of WordNet.

All the WordNets of languages different than English in order to support cross-linguism, should map their synsets to the entries of the ILI by means

[40] Global WordNet Association Web Site - http://www.globalwordnet.org/
[41] Euro WordNet Web Site - http://www.illc.uva.nl/EuroWordNet/

of four kinds of mapping relations (EQ SYNONYM, EQ NEAR SYNONYM, EQ HAS HYPERONYM, and EQ HAS HYPONYM). The ILI mapping methods allow independently developing every single WordNet enabling at the same time the interoperability with other WordNets of different languages by exploiting the mappings of the synsets to the corresponding ILI records.

Currently the Euro WordNet project is continued through the Global WordNet association (GWA). GWA was founded in 2000 and represents the most relevant initiative that aims at supporting multilingual interoperability between WordNets. GWA is a non-profit organization that wants to collect WordNets of different languages in order to promote the development of methodologies, standard procedures and shared representations to support their interactions.

In the GWA the notion of Base Concepts (BCs) is exploited so as to reach maximum overlap and interoperability among WordNets in different languages. BCs are the most representative WordNets synsets, since they have a high position in the WordNet semantic hierarchy and are characterized by many relations with other synsets. A set of 1024 Base Concepts has been extracted from the version 1.5 of the English Princeton WordNet and mapped to the concepts of the Suggested Upper Merged Ontology (SUMO) [25]. WordNets of other language have been invited to start their building process from the set of BCs or to map their synset to the BCs so as to support cross-linguism.

1.3.3 Ontologies: representing and reasoning about data

The term ontology has been originally coined in philosophy to refer to basic existential issues and then has been adopted also by the artificial intelligence (AI) and the knowledge management research.

An ontology is a typology of knowledge resource representing 'a formal, explicit specification of a shared conceptualization' [26]. An ontology is:

- a *conceptualization*, indeed it represents a conceptual model of a specific domain;
- *formal* because it must be machine-understandable and processable;
- *explicit* because it needs to be defined in an unambiguous way and shared because it must be commonly accepted by a community of users that refer to it.

Ontologies are typically composed of three kinds of entities [27]:

- a set of *concepts* that characterize the considered domain;

- a set of *relations* between those concepts;
- a set of *instances* of particular entities along with their specific properties.

Ontologies can be expressed adopting different formalisms or description languages. A formalism is a collection of various constructs useful to support the formal description of a particular domain of interest. When choosing a formalism, it is important to determine the right trade-off between two main opposite needs: its *expressive power* and its *complexity of reasoning*.

The *expressive power* is the richness of different available constructs that could be exploited to describe a particular domain of interest (i.e. the possibility to precisely define the properties of every concept or relation or to express more or less complex constraints). Closely related to the expressive power issue, other important properties that should be considered when we deal with a formalism concern the possibility to entail new knowledge from the information explicitly stated by means of the same formalism. These properties are:

- the *correctness of entailment procedure*: the impossibility to draw false entailed conclusions;
- the *completeness of entailment procedure*: the ability to draw all correct conclusions;
- the *decidability of entailment problem*: the existence of an algorithm which compute the entailed knowledge in a finite number of steps.

The *complexity of reasoning* can be figured out as the variable amount of computing resources needed to obtain new entailed knowledge through a specific reasoning algorithm. This algorithm applies a specific set of reasoning rules or procedures to the knowledge described by means of a formalism. An increase in the expressive power of a formalism will usually mean an improvement of its description possibilities of a particular domain, but also an increase in the complexity of reasoning.

Thus the ease and directness of use of the formalism will decrease, making more difficult to take advantages of it. Moreover, by increasing the expressive power of a formalism, it could happen that the formalism loses its decidability or its completeness and correctness.

Depending on the specificity of the domain, ontologies can be classified considering the *generality of the conceptualization behind them*. In

particular, three kinds of ontologies can be picked out moving from a general to a very specific conceptualization of the domain of interest:

- Upper level ontologies represent a general model of the world. They usually are vague, not much specific. They could be adopted in many different application areas. Examples of lightweight upper level ontologies are the Suggested Upper Merged Ontology (SUMO) [25] and the Descriptive Ontology for Linguistic and Knowledge Engineering (DOLCE) [28];
- Domain ontologies conceptualize a particular domain. Domain ontologies are strictly related to a specific context and are usually reusable, but sometimes also conflicting. An example of domain ontology is MeSH [29] which describes concepts related to the medical domain;
- Application and task ontologies are extremely specific conceptualizations, usually related to a particular application or used to provide support to a defined task. Generally, they aren't reusable. For instance an ontology of this kind could be adopted to express the delivery date of a letter or the confirmation of an order of a particular product sold by a commercial company.

Relying on the semantics introduced by the use of ontologies, we make knowledge explicit, formal, thus machine-understandable and processable. For this reason ontologies represent the main typology of knowledge resource currently exploited on-line in the context of the Semantic Web. Web Ontologies constitute shared conceptual representations of the knowledge characterizing a domain of interest, useful to semantically describe Web contents by means of semantic meta-data, being thus the enabling factor of both distributed data integration patterns and automated reasoning procedures based on the machine-understandable representation of the semantics of Web contents.

1.3.3.1 OWL: the DL formalism to define Web Ontologies

Description logics (DL) are a family of knowledge representation formalisms. DLs have an important role in the context of the Semantic Web because they represent the formal basis of the Web Ontology Language (OWL)[42], the semantic knowledge description language standardized by the W3C, universally adopted to specify ontologies over the Web.

[42] W3C OWL Working Group - http://www.w3.org/2007/OWL/wiki/OWL_Working_Group

DL knowledge bases

The DLs are decidable subsets of the First Order Logic (FOL). There are different kinds of description logics characterized by different degrees of expressive power and then distinct computational complexity. When deciding which family of description logic to use, the central idea to keep in mind is to obtain the needed expressive power, keeping the possibility of doing reasoning under control so as to obtain as much efficiency as possible [30].

The main two building blocks of description logics are *concepts* and *roles*. A concept is the characterization of a set of individuals while a rule is a kind of relation that could hold between two individuals. The different description logic families are distinguished by different sets of concepts and rules constructors that are the distinct expressive means available to construct concepts and rules descriptions.

By exploiting the concept and rule constructors, a description logic knowledge base can be defined. Description logic knowledge bases are constituted by two major components: the terminological box and the assertional box:

- the *Terminological Box* (T-Box or schema): set of statements describing the structure of the domain of interest in terms of the definitions of concepts and the relations between concepts (roles);
- the *Assertional Box* (A-Box or world description): set of axioms describing the concrete data, the specific individuals of the considered domain exploiting the concepts and the rules defined in the T-Box.

The A-Box is usually characterized by two fundamental assumptions: the *Unique Name Assumption* (UNA) and the *Open World Assumption* (OWA). While the UNA states it is assumed that two individuals identified by different names are distinct, the OWA states that one cannot assume that the knowledge in the knowledge base is complete; in other words, if some assertion isn't contained in the knowledge base and cannot be formally inferred, we cannot make any assumption about its truth or falseness.

Figure 14 provides an example of a description logic knowledge base where an A-Box and a T-Box related to a company employees domain are defined.

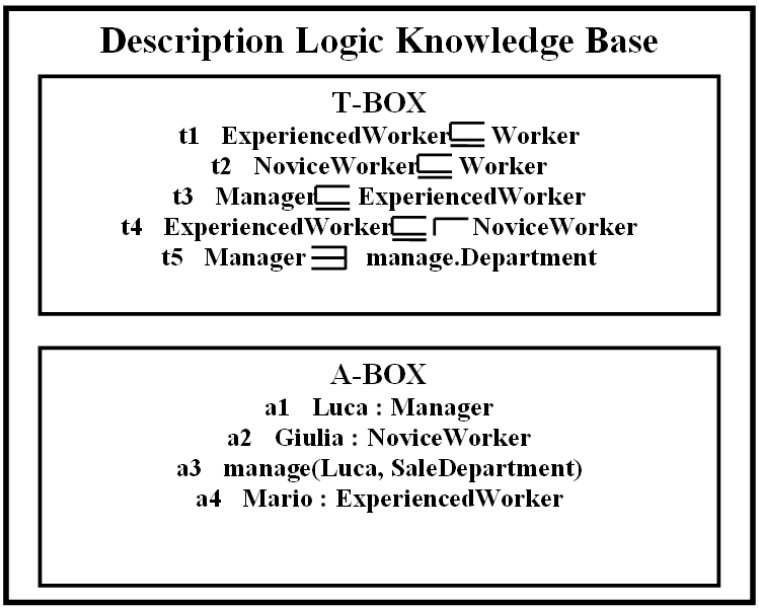


Figure 14: Example of description logic knowledge base

In the T-Box the concepts or classes: Worker, ExperiencedWorker, NoviceWorker, Manager and Department and the property manage are defined by using two concept constructors: the negation (t4) and the existential quantifier (t5). The possibilities of the concepts subsumption constructor are exploited to state that: every Experienced Worker is a Worker (t1); every Novel Worker is a Worker (t2); every Manager is an Experienced Worker (t3). In this a sort of concepts (or classes) hierarchy is defined. The disjunction construct is exploited to state that the classes ExperiencedWorker and NoviceWorker are disjoint (t4) and the existential quantification construct (t5) to state that every manager must manage at least one department.

In the A-Box there is a description of the concrete data, regarding three people: Giulia, Luca and Mario. Giulia is a Novice Worker (a2), Mario an Experienced Worker (a4) and Luca a Manager (a1) who manages the Sale Department (a3). There is consistency between the general formal descriptive assertions expressed through the T-Box and the real data contained in the A-Box.

Reasoning procedures based on DL

Starting from the knowledge contained in a description logic knowledge base, we can realize different *automated reasoning services* [31]. Reasoning services are algorithmic procedures that verify some particular assumption

relying on the derivation of new inferred knowledge from the knowledge explicitly asserted in the knowledge base. In this case, they exploit the expressive power of the description logic language, adopting a set of specific inference rules. For instance from the knowledge base represented in Figure 14: Luca is a Manager (a1), every Manager is an Experienced Worker (t3) and every Experienced Worker is a Worker (t1). As a consequence, it can be inferred that Luca is a Worker or that Luca is an Experienced Worker. These assertions are both implicit and can be derived by exploiting the description logic semantic content from the set of assertions explicitly stated. In particular the transitivity that characterizes subsumption hierarchies is exploited to perform this specific inference.

The classical reasoning services that can be realized processing a description logic knowledge base are:

- *Concept satisfability*: it checks if a class (or concept) can have any instances. If a class is unsatisfiable the whole ontology is inconsistent;
- *Consistency checking*: it checks if the A-Box is consistent with the respective T-Box or, in other words, that there are no contradictory facts;
- *Classification*: definition of the complete classes hierarchy in order to determine which classes are subsumed or subsume other classes;
- *Realization*: definition of the most specific class a given individual belongs to; it can be verified only after the execution of the classification.

All the reasoning services can be expressed in terms of more or less complex issues of concept satisfability. The algorithm adopted to verify the satisfability of a given concept is the tableaux algorithm that uses a set of reasoning inference rules and tries all possibilities to prove that the considered concept is satisfiable. This algorithm is sound (or correct: it always draws the correct result) and complete (if it fails to verify the satisfability of a concept it is unsatisfiable).

The Web Ontology Language

The Web Ontology Language (OWL) is the description language globally adopted over the Web to represent ontologies. OWL expressive power is based on a particular Description Logic family. OWL includes three sublanguages characterized by a greater expressivity:

- *OWL Lite* is the less expressive OWL sublanguage. OWL Lite allows defining classes and properties that can be organized by means of

subsumption hierarchies. The domain and range of properties can be specified together with the equivalence between pairs of classes and properties. Also instances of classes and properties can be defined together with equivalence or difference constraints among them. Basic property features like restrictions and cardinality constraints can be specified.

- *OWL DL* has the same expressive power of OWL Lite enriched with the possibility to define enumerated classes and restrictions over the value that a property can assume. OWL DL allows specifying also disjointness, union, intersection and complement between classes as well as more articulated cardinality constraints over properties.
- *OWL Full* lacks many expressivity constraints proper of OWL DL, thus it represents the most powerful OWL sublanguage. OWL Full has been designed in order to preserve compatibility with RDF Schema, but it is no more decidable, thus it is no more possible to apply reasoning procedures over OWL Full ontologies.

Nowadays the great part of OWL ontologies over the Web is expressed using OWL DL sublanguage. OWL Lite is not so less expressive to justify its adoption in terms of better reasoning performances. OWL Full is not decidable and thus standard automatic reasoning techniques cannot be applied.

The first version of OWL, referred to as OWL 1 has been standardized by the World Wide Web Consortium (W3C) as a series of Recommendations in February 2004. Since that date a growing set of new requirements were discussed in order to extend OWL 1. This need was a consequence of the development of new reasoning algorithms and of the feedbacks coming from the many experiences done concerning the design of OWL ontologies. Therefore a new revised version of OWL, referred to as OWL 2[43], was defined and published as a W3C Recommendation in the last quarter of 2009. It extends OWL 1 with new expressive constructs as well as new possibilities to define constraints over the formalized knowledge by a more compact syntax.

Below there is a simple example of OWL DL ontology, serialized through the standard OWL XML syntax[44], concerning the description of a family.

[43] Web Ontolgy Language 2 W3C - http://www.w3.org/TR/owl2-primer/
[44] OWL XML Presentation Syntax - http://www.w3.org/TR/owl-xmlsyntax/

```
<?xml version=¨1.0¨?>

<rdf:RDF xmlns:rdf=¨http://www.w3.org/1999/02/22-rdf-syntax-ns#¨

xmlns:xsd=¨http://www.w3.org/2001/XMLSchema#¨

xmlns:rdfs=¨http://www.w3.org/2000/01/rdf-schema#¨

xmlns:owl=¨http://www.w3.org/2002/07/owl#¨

xmlns=¨http://www.mylocation.it/myontology.owl#¨

xml:base=¨http://www.mylocation.it/myontology.owl¨>


<!-- This OWL element specifies the metadata that characterize the
ontology; in this case the empty attribute rdf:about points out that
the URI of the whole ontology is those used to refer the file that
contains it over the Web -->


<owl:Ontology rdf:about=¨ ¨/>

<!-- Definition of the class Person -->

<owl:Class rdf:ID=¨Person¨/>

<!-- Definition of the class Man which is a subclass of the class
Person and its set of instances is disjoint from those of the class
Woman -->

<owl:Class rdf:ID=¨Man¨>

<rdfs:subClassOf rdf:resource=¨#Person¨/>

<owl:disjointWith rdf:resource=¨#Woman¨/>

</owl:Class>


<!-- Definition of the class Woman which is a subclass of the class
Person and its set of instances is disjoint from those of the class
Man -->

<owl:Class rdf:ID=¨Woman¨>

<rdfs:subClassOf rdf:resource=¨#Person¨/>

<owl:disjointWith rdf:resource=¨#Man¨/>

</owl:Class>


<!-- Definition of the class Father as a subclass of the class Man,
stating that every instance of the class father must be the subject of
at least one RDF-triple characterized by the property hasChild -->

<owl:Class rdf:ID=¨Father¨>

<rdfs:subClassOf rdf:resource=¨Man¨/>

<owl:Restriction owl:minCardinality=¨1¨>

<owl:onProperty rdf:resource=¨#hasChild¨/>

</owl:Restriction>

</owl:Class>
```

```
<!-- Definition of the property hasChild which must have as subject
and as object an element/instance of the class Person; its inverse
property is hasParent -->
<owl:ObjectProperty rdf:ID=¨hasChild¨>
<rdfs:domain rdf:resource=¨#Person¨/>
<rdfs:range rdf:resource=¨#Person¨/>
<owl:inverseOf>
<owl:ObjectProperty rdf:about=¨#hasParent¨/>
</owl:inverseOf>
</owl:ObjectProperty>

</rdf:RDF>
```

Considering DL knowledge bases, we can specify by means of OWL ontologies the set of statements describing a domain of interest in terms of the definitions of concepts (Person, Man, Woman, Father) and the relations between concepts (hasChild). All these entities will constitute the *Terminological Box*. The factual knowledge, describing real world entities together with their relations (*Assertional Box*) is usually specified by means of RDF statements[45].

In Figure 15, we graphically represent a simplified version of the family OWL ontology just described including four classes organized in a subsumption hierarchy and one property. Then we specify that "John has child Juliana" by means of three RDF triples that state:

- John is an instance of the class *Father*
- Juliana is an instance of the class *Woman*
- John has child Juliana.

This set of RDF triples represents factual knowledge (A-Box) describing real world entities (John and Juliana instances respectively of the ontological classes *Father* and *Woman*) together with their relationship (hasChild instance of the *hasChild* ontological property) with respect to the domain model (T-Box) specified by means of the family OWL ontology.

[45] W3C Resource Description Framework - http://www.w3.org/RDF/

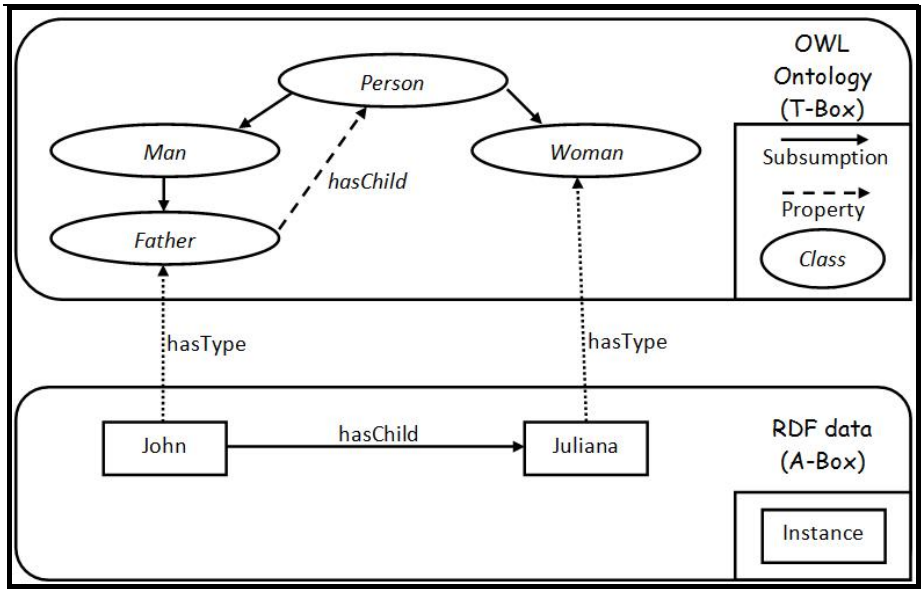


Figure 15: OWL ontological knowledge and RDF factual knowledge

2. EDITING KNOWLEDGE RESOURCES: THE WIKI WAY

Currently the exploitation of processable semantic descriptions of Web contents represents one of the mainstream approaches to support Web users in dealing with the growing, distributed, and heterogeneous amount of information available on-line. By accessing semantic descriptions of Web data, software agents can search, aggregate and integrate information spread across distinct on-line sources on the basis of users' informative needs.

In this context, the rise of the Semantic Web has promoted the definition of shared standards and best practices to represent, interconnect, and publish on-line semantic descriptions of data, also known as semantic metadata. The Resource Description Framework (RDF) currently constitutes the worldwide language to represent semantic meta-data over the Web by means of RDF triples. The Web Ontology Language (OWL) has turned into the global standard to define shared on-line ontologies. OWL ontologies formalize the knowledge related to a specific domain, thus supporting logical grounding and interoperability across semantic RDF datasets published on-line by distinct sources.

All the activities concerning both the creation of semantic descriptions of Web contents and above all the definition and maintenance of shared knowledge resources usually require the contributions of a considerable number of actors with different expertise, ranging from common Web users to domain experts and knowledge engineers. Therefore, *during the last few years several collaborative editing tools, often Web-based, have been proposed so as to support knowledge editing activities*. The wiki paradigm has been largely adopted to enable the collaborative editing of knowledge resources, like OWL ontologies or in general, any semantic or linguistic network.

Section 2.1 motivates the usefulness of collaborative editing methodologies to manage knowledge resources. Section 2.2 is a detailed review of the most relevant environments to edit knowledge resources, with particular emphasis on those tools adopting the wiki paradigm. Subsequently, section 2.3 discloses a classification of tools according to a common set of descriptive criteria. A comparison among the tools allows the definition of a core set of desirable features of a collaborative knowledge editor. Finally, section 2.4 emphasizes the importance of users' motivation in order to promote massive contribution in knowledge editing.

2.1 THE WIKI PARADIGM APPLIED TO KNOWLEDGE RESOURCES

The wiki paradigm is an expression adopted to refer to the collaborative editing of a specific typology of contents by exploiting a kind of Web-based tools called wiki environments. The core feature of the wiki paradigm is *the social aspect of content creation*. A wiki environment indeed enables distributed communities of users to actively contribute to the editing activities of shared contents. The users of this kind of tools are pushed to create and share contents because they feel they are contributing to a community effort useful to produce knowledge the whole community can take advantage of.

The term wiki was used with this meaning for the first time in 1995 when Ward Cunningham launched on-line the first wiki environment[46]. The initiative was intended to foster the exchange of ideas between programmers by giving them the possibility to collaboratively edit the contents of a set of shared Web pages. The term Wiki is the Hawaiian language word for fast, and it was chosen to refer to the possibility to rapidly edit Web contents. The concept of WikiWikiWeb pages was introduced by Cunningham to describe Web pages whose contents can be edited on-the-fly by their users.

The wiki environment created by Cunningham represents the first example of Web tool for social content creation. Ever since many other wiki tools have been developed so as to support the collaborative editing of different kinds of contents inside open or closed user communities. The most famous and widespread successor of the first Wiki environment is *MediaWiki*[47], a popular Web-based wiki software application. MediaWiki has been developed since 2003 by the WikiMedia Foundation[48], a non-profit charitable organization that operates also several on-line collaborative wiki projects, among them *Wikipedia*[49]. Based on the MediaWiki platform, Wikipedia currently constitutes the most relevant collaboratively edited encyclopaedia including in its English version more than 3,5 million of articles[50] (December 2010). Other examples of relevant wiki projects sponsored by the WikiMedia Foundation are *Wiktionary*[51], a multilingual Web-based free content dictionary, *Wikiquote*[52], a vast reference of

[46] The first Wiki Wiki Web environment - http://c2.com/cgi/wiki
[47] MediaWiki Web Site - http://www.mediawiki.org/wiki/MediaWiki
[48] MediaWiki Foundation Web Site - http://wikimediafoundation.org/wiki/Home
[49] English Wikipedia Web Site - http://en.wikipedia.org/wiki/Main_Page
[50] Wikipedia statistics - http://stats.wikimedia.org/EN/TablesArticlesTotal.htm
[51] Wiktionary Web Site - http://www.wiktionary.org/
[52] Wikiquote Web Site - http://www.wikiquote.org/

quotations from prominent people, books, films, and proverbs, *Wikibooks*[53], a collection of free content textbooks and annotated texts that anyone can edit, and *Wikispecies*[54], a comprehensive free content catalogue of all species (see Figure 16).

Figure 16: Some examples of wiki tools based on the MediaWiki software application

Several wiki environments have been developed to support the collaboration in small communities of users by enabling them to share and collaboratively edit specific kinds of contents. For instance *GoogleDocs*[55] allows groups of users to share and collaboratively edit textual documents, spreadsheets and presentations by exploiting their Web browser, *MindMeinster*[56] supports the Web-based collaborative editing of mind maps, and *Central Desktop*[57] enables groups of users to share a workspace similar to their desktop.

Summarizing, some distinguishing features common to most of the wiki environments are:

- the ease of access through a common Web browser;

[53] Wikibooks - http://www.wikibooks.org/
[54] Wikispecies - http://species.wikimedia.org/wiki/Main_Page
[55] Google Docs - https://docs.google.com/
[56] MindMeinster - http://www.mindmeister.com/
[57] Central Desktop - http://www.centraldesktop.com/

- the immediate update of changes that are visible to all the members of the wiki community;
- the simple content editing syntax;
- the possibility to track changes and to rollback modifications;
- the support for argumentation over edited contents;
- the possibility to interlink and interrelate the edited contents.

Semantic technologies have experienced a great expansion over the Web in the last few years. As a consequence, Knowledge resources have been more and more exploited as knowledge references in a distributed Web context. The contents of knowledge resources are characterized by *continuous changes* because the knowledge they formalize is usually in constant evolution. In order to be kept up to date, knowledge resources need *editing contributions from editors that are spread over different locations all over the world* and have *different levels of expertise* ranging from knowledge engineers to domain experts.

Considering this scenario the adoption of wiki environment represents a valuable approach to edit and maintain knowledge resources. In wiki environments the editing activities can be performed over the Web, often through a Web browser. Therefore distributed communities of users can contribute without location or time constraints and can cooperate to keep the formalized knowledge up to date. Intuitive, visual Web interfaces can be exploited in order to enable users without any knowledge engineering background to edit the set of complex knowledge structures that characterize knowledge resources like OWL ontologies.

The most relevant kinds of wiki environments to edit knowledge resources are reviewed below.

2.2 ENVIRONMENTS TO EDIT KNOWLEDGE RESOURCES

The aim of this section is to present the most relevant examples of environments to edit knowledge resources by describing their core features, functional aspects, and implementation perspectives. The environments have been grouped into two categories: semantic wikis and ontology editors. Semantic wikis are wiki environments in which contents are structured by means of semantic annotations. Ontology editors are applications that enable users to edit ontologies. Ontology editors can exploit a graphical interface to browse and edit ontologies or rely on a controlled language. Thanks to the adoption of a controlled language, the

knowledge formalized in an ontology can be edited by means of natural language interactions.

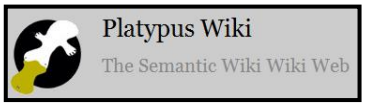
2.2.1 Wiki editors of textual contents enriched with semantic annotations

In order to better structure Web contents, semantic wiki environments allow formalizing their meanings through RDF statements. RDF statements are specified by annotating textual contents or by means of customized Web interfaces. SAVVY wiki represents the only exception. SAVVY wiki is a semantic wiki environment that allows structuring its contents by exploiting special knowledge structures.

All the wiki environments described are accessible by a Web interface.

Platypus Wiki

Web Link: http://platypuswiki.sourceforge.net/



Platypus Wiki [32] is one of the first examples of semantic wiki. It has been developed since the end of 2003 as an open source project. The last version is dated 2004.

Platypus Wiki includes RDF meta-data inside each wiki-page in order to better structure its contents. Each wiki-page describes a knowledge entity and is pointed out by an URL. A proper URL syntax is defined in order to retrieve the HTML textual contents or the set of RDF triples associated to the wiki-page. RDF triples related to a wiki-page are displayed in a proper set of boxes included in the Web interface layout of Platypus Wiki. In particular, there is a box showing all the RDF triples that have the resource represented by the wiki-page as their subject and a box displaying all the RDF triples that have the resource as their object. A third box is included in the Web interface to show the list of RDF triples that have the resource represented by the current wiki-page as their subject but have also a literal as their object. Users can edit the textual contents of wiki-pages as well as the set of RDF triples connected to the same wiki-page. On the basis of the label assigned to each wiki-page in Platypus Wiki, links to other wiki-pages are automatically included in the textual contents when the same label occurs. Basic versioning features are implemented. No consistency checks over the RDF data are performed.

Platypus Wiki has been implemented in Java, as a Web application relying on the Apache Tomcat servlet container by also exploiting Java Server Pages (JSP).

Semantic MediaWiki

Web Link: http://semantic-mediawiki.org/wiki/Semantic_MediaWiki

Semantic MediaWiki [33, 34] is an extension of MediaWiki[58], the famous wiki-engine exploited by many important Wiki projects, like Wikipedia. Semantic MediaWiki enables users to better specify the structure of the edited wiki contents by exploiting semantic technologies in order to realize new improved possibilities to browse, aggregate and retrieve information.

Semantic MediaWiki represents one of the most popular open-source Semantic Wiki projects. It is currently used in more than 200 public active wikis around the world[59] and in an unknown number of private contexts. Semantic MediaWiki was initially developed as a research project by the Institute of Applied Informatics and Formal Description Methods (AIFB) of the University of Karlsruhe. Since its first release, on September 2005, many people, companies and initiatives have contributed to the project.

Semantic MediaWiki allows users to semantically characterize the links between wiki contents. Each link is described by a property that makes explicit the meaning of the connection between the wiki-page containing the link and the target of the same link. Possible targets of a link can be other wiki-pages or data values of some type (string, number, date, etc.).

For instance, in the wiki-page describing the city of Rome we can create a link to the wiki-page describing the state of Italy, stating that the property characterizing that link is “isTheCapitalOf”. By means of “Category” links, each wiki-page can be connected to one or more Categories, usually representing OWL classes. In this way the wiki-page is declared to be an instance of the class. In Semantic MediaWiki, the Wiki Markup Language (WikiML) has been properly extended so as to support the definition of semantic links.

Wiki-pages can be visualized by means of additional widgets like the fact box showing the explicit semantic information contained in a specific page. The classical Wikipedia templates have been turned into semantic templates where each row is associated with a specific semantic property exploitable also to define semantic links.

[58] http://www.mediawiki.org/wiki/MediaWiki
[59] http://smw.referata.com/wiki/Special:BrowseData/Sites?Status=Active

Wiki-pages, categories and semantic links can be mapped to external OWL ontologies so as to support the reuse of knowledge definitions. Complex search patterns over semantically annotated pages can be defined by exploiting WikiQL, a proprietary query language useful to take advantage of the semantic structuring of contents in Semantic MediaWiki. Wiki-pages as well as snippets of contents can be dynamically generated by defining specific WikiQL queries.

The consistency and the reuse of contents are fostered in Semantic MediaWiki due to its structuring of information. Data can be exported in many different formats like RDF, CVS, and JSON. The version control and rollback mechanism is the same as the one exploited in MediaWiki.

Semantic MediaWiki, like Semantic MediWiki, is written in PHP and exploits an SQL database to support data persistency. More than 40 extensions to Semantic MediaWiki have been developed by several contributors. They aim at improving data browsing, editing and integration. Semantic Drilldown[60] is an extension that provides a faceted browser to navigate the contents of Semantic MediaWiki, enabling users to select the information needed by means of a set of filters. Extensions like LinkedWiki[61], RDFIO[62], SPARQLextension[63] and Triple Store Connector[64] enable users to store Semantic MediaWiki data in a triple-store and to query them by exploiting SPARQL.

IKEWiki and the KiWi project

Web Link: http://www.kiwi-community.eu/dashboard.action
http://www.kiwi-project.eu/

IkeWiki [35, 36] is a wiki environment that allows users to semantically enrich the contents of wiki-pages in order to improve their browsing and searching experience. IkeWiki has been developed since 2005 by the Knowledge and Media Technologies Department at the Salzburg Research Centre, a non-profit research organisation of the State of Salzburg.

IkeWiki extends the visualization features of wiki articles thanks to the possibility to define and graphically represent the metadata describing each

[60] http://www.mediawiki.org/wiki/Extension:Semantic_Drilldown
[61] http://www.mediawiki.org/wiki/Extension:LinkedWiki
[62] http://www.mediawiki.org/wiki/Extension:RDFIO
[63] http://www.mediawiki.org/wiki/Extension:SparqlExtension
[64] http://smwforum.ontoprise.com/smwforum/index.php/Help:TripleStoreConnector_Basic

one of them. Each article can be associated to one or more ontological classes and characterized by a set of custom metadata. Articles' link to other articles can be typed so as to make explicit their semantics. The addition of semantic information to wiki articles is very easy for IkeWiki users, thanks to editing facilities like interactive link typing.

IkeWiKi is compatible with MediaWiki and thus with the great amount of information available in Wikipedia. IkeWiki adopts Semantic Web standards like RDF and OWL and supports different levels of knowledge formalization and distinct kinds of users' expertise.

IkeWiki is implemented as a Java application storing the texts of wiki-pages in a SQL repository. An RDF store is exploited to save the semantic metadata associated to each wiki-page, represented by RDF triples. During the rendering process, textual contents are properly mixed with semantic information and can be visualized to the user by exploiting different data formats, including XHTML. Specific data retrieval and editing features are available in the generated XHTML pages thanks to the support of AJAX interactions. The choice to keep textual contents separated from semantic information is essential to allow an easy access and querying of the semantic information, realized through a SPARQL engine.

IkeWiki is deployed thanks to a Tomcat server. The RDF triples used to semantically annotate each page are managed thanks to the Jena framework in an in-memory RDF triple store.

Starting also from the experience related to the development of IkeWiki, the KiWi (Knowledge In a Wiki) FR7 STREP project [37] started in 2008. KiWi is a 3-years collaboration between universities and industrial partners, lead by the Knowledge and Media Technologies Department of Salzburg Research Centre. KiWi aims at defining methodologies and environments where wiki patterns for content editing can be integrated with Semantic Web intelligence and methods allowing users to easily interact and refine data, in a global participation context. The KiWi project has among its goals the analysis of the most adequate way to define and make semantic wikis usable by huge communities of users, in a user-centric environment, bringing together application experts and knowledge engineers.

At the end of 2010, the KiWi consortium released the version 1.0 of the KiWi system. It is a Java-based platform composed of a set of modules devoted to manage specific features of a collaborative semantic content management system. Among the components of the KiWi system there are modules useful to index data, manage RDF triples and OWL ontologies, log

users' activities, support argumentation, manage vocabularies, extract information and semantically query datasets.

OntoWiki

Web Link: http://ontowiki.net/Projects/OntoWiki
http://code.google.com/p/ontowiki/

OntoWiki [38, 39] is a Web application useful to easily browse and edit RDF knowledge-bases by means of an interactive user interface. OntoWiki has been developed by the Agile Knowledge Engineering and Semantic Web (AKSW) Group at the University of Leipzig since 2005. OntoWiki represents one of the most relevant examples of Semantic Web collaborative environment. Unlike text-based Semantic Wikis, OntoWiki do not extend the Wiki Markup Language in order to explicit the semantics of data. On the contrary every data is represented and stored by means of RDF triples.

Many different ways to visualize and navigate RDF triples like hierarchical and faceted browsing are available in OntoWiki. OntoWiki allows full-term searches, but also the definition of complex search filters thus taking advantage of information structuring. All these different navigation features are always translated into SPARQL queries over the OntoWiki RDF datasets. In addition, custom data view can be defined to show for instance geographically located data, calendars or to allow browsing particular resources like SKOS Vocabularies and FOAF profiles.

The authoring of RDF contents can be performed by exploiting specific Web forms. Thanks to the adoption of RDFa annotations, RDF triples can also be embedded directly in OntoWiki Web pages. In this way, the RDFauthor tool [40] can be exploited in order to enable in-place editing functionalities. RDFauthor is a Javascript tool that allows managing and editing the RDF statements included in RDFa annotated Web contents. Visual widgets can be exploited to define custom views of the data to be edited by considering if a particular RDF property is a datatype or an object one as well as by evaluating the domain and range restrictions of the same property. RDFauthor can store the modifications carried out over RDF triples by exploiting the SPARQL Update language.

Predefined patterns can be applied so as to support the evolution of ontologies and other knowledge models in parallel with the evolution of the real knowledge expressed by means of RDF statements.

OntoWiki can show and retrieve data from external sources by exploiting the Linked Data paradigm. Moreover, it implements semantic Pingback client and server features [41] so as to support the automated creation and propagation of typed links between RDF datasets.

OntoWiki provides support for changes tracking and detailed users logging, annotations, contents' rating, and statistics about contents popularity. OntoWiki exploits the PHP server-side scripting language. Data can be stored by exploiting an SQL database management system or an RDF triple store.

Recently OntoWiki Mobile [42] has been developed as a mobile Web application implemented in HTML5 and Javascript. It allows navigating and editing OntoWiki RDF knowledge base on mobile devices by enhanced faceted browsing functionalities. Thanks to HTML5 local data storage features, OntoWiki Mobile enables users to edit portion of RDF knowledge bases without an Internet connection. Proper server-side conflict resolution protocols are exploited to merge the edited contents with the main knowledge base.

Rizhome

Web Link: not available

Rizhome Wiki [43, 44] is a Web-based wiki environment to manage RDF resources. Rizhome Wiki has been developed since 2003 as an open-source project. The most recent version has been released in 2005.

Rizhome allows editing the entire structure of a Web site as RDF contents by exploiting ZML, a plain-text XML formatting language introducing wiki-like mark-up inside textual contents so as to specify their semantic structure as RDF statements. Rizhome relies on a Web-interface that allows editing the raw ZML specification of each page. The management of the ZML specifications of the different wiki pages as well as their storage in terms of RDF triples is realized by a Web server written in Python.

SweetWiki

Web Link: http://semanticweb.org/wiki/SweetWiki

SweetWiki [45] is a Web-based collaborative contents editor that exploits semantic information structuring in order to better organize and manage contents. SweetWiki has been developed since the end of 2005 by the Acacia Group of the INRIA laboratory in France. SweetWiki exploits an ontological model

that describes the different entities that characterize a wiki (wiki-pages, internal links, authors, and keywords). In this way it is more efficient to search through the information included in the SweetWiki, and it is possible to create aggregated views. Thanks to this ontological model of wiki contents, a sort of schema to support interoperability of the contents among different wiki engines is defined. Any wiki syntax to be embedded in textual contents is adopted in order to edit the structured contents included in wiki-pages. To this purpose a WYSIWYG HTML editor is exploited.

Each wiki-page can be semantically tagged by means of one or more semantic keywords that are ontologically structured hierarchies of concepts, each one referred by a proper set of labels. By doing so, wiki-pages can be semantically classified by topic. In order to reuse as much as possible existing keywords, auto-complete functionalities are provided. The structure of keyword-based classifications of concepts can be refined and improved by experts exploiting an external ontology editor.

The RDF triples describing each wiki-page are embedded in the HTML syntax by means of RDFa. SweetWiki has been implemented in Java as a Web-based application relying on the Tomcat servlet container, the Java Server Pages and exploiting the CORESE engine[65] and the SeWeSe Library[66] Java tools for all the semantic operations.

Maariwa

Web Link: not available

Maariwa [46] is a Web-based semantic wiki that wants to enable non-expert users to edit both wiki-pages by enriching them with semantic metadata as well as the underlying ontological model. Maariwa has been developed by the Friedrich-Schiller-Universität Jena. Each page of this semantic wiki is characterized by two areas: a box devoted to edit natural language texts, and the metadata area that displays the semantic annotations giving users the possibility to edit them. The annotated snippets of texts are underlined with proper colours. The annotations of each page can be visualized as RDF triples and exported as RDF/XML. The Maariwa Web interface allows the import and the modification of the ontologies that will be used to annotate textual contents. Maariwa adopts an ontology meta-model based on a subset of OWL-lite expressivity. Maariwa implements the MarQL Semantic Query Language. MarQL has a

[65] http://www-sop.inria.fr/teams/edelweiss/wiki/wakka.php?wiki=Corese
[66] http://www-sop.inria.fr/teams/edelweiss/wiki/wakka.php?wiki=Sewese

more compact syntax if compared to SPARQL but is less flexible. Thanks to MarQL, users can search among the set of annotated wiki-pages by specifying elementary constraints also over ontological elements such as classes, attributes and relationships. Maariwa has been implemented in Java.

SAVVY Wiki

Web Link: not available

The Semantic Association Various Viewpoint sYstem (SAVVY) Wiki [47] is a wiki that aims at organizing fragmented information. SAVVY wiki has been developed by the National Institute of Information and Communications Technology (NICT) of Kyoto.

SAVVY wiki enables users in grouping sparse data into homogeneous views referred to as subject pages. Each subject page can contain contents taken from other wikis or generic Web contents. Subject pages can be browsed by means of two views: the arrangement view and the surrounding view. The arrangement view shows all the information fragments collected in the considered subject page. By contrast, the surrounding view allows users browsing all the information fragments linked to the ones included in the considered subject page but belonging to different subject pages.

2.2.2 Ontology editors

The knowledge editors reviewed in this subsection deal mainly with OWL ontologies. They are divided into two groups depending on how they interact with users. The first group includes all the environments that exploit some sort of graphical interface. The second group includes three examples of ontology editors and ontology editing methodologies based on the exploitation of a controlled language are described in the second group.

2.2.2.1 Ontology editors based on a graphical interface

All the tools described in this subsection rely on a graphical interface that enables users to edit and browse ontological contents. Most of them are desktop applications because they exploit complex visualization patterns in order to support knowledge browsing and editing actions that otherwise would be difficult to implement through a Web interface. Protégé and CODA are accessible through a Web interface.

Collaborative Protégé and Web Protégé

Web Link: http://protege.stanford.edu/

Protégé [48] is a free open-source ontology editor and knowledge-base framework, developed by the Stanford Centre for Biomedical Informatics Research at the Stanford University School of Medicine. Protégé is one of the most widely adopted knowledge base editing environments. It is supported by a huge community of users including more than 160.000 members. Protégé allows editing ontologies that can be modelled as frame-based ontologies or Semantic Web ontologies. Considering the latter modelling scheme, both RDF(S) and OWL ontologies are supported. From the version 4, also OWL 2 ontologies can be edited in Protégé.

Protégé is based on a client-server architecture that enables multiple users to simultaneously browse and edit the same ontology. A set of features devoted to support collaborative editing of ontologies has been implemented: this collaboration-aware version of the system is usually referred to as Collaborative Protégé. All users' modifications to the ontologies edited in Protégé are modelled and can be commented by exploiting an appropriate annotation ontology so as to enable users' argumentation. Parts of an ontology as well as ontology modification actions can be annotated. Discussion threads about specific ontology modifications as well as the possibility for users to vote ontology modifications have been implemented. It is also possible to search for and specify filters over contents to retrieve annotations.

In Protégé, workflows [49] for collaborative ontology development can be defined, thus specifying possible user interaction patterns and user access rights. Ontology editing workflows can be specified by exploiting a proper ontological model. A proper software infrastructure to instantiate and manage ontology editing workflows has also been defined.

Protégé has been implemented in Java. A set of API is available both to deal with knowledge-base contents and to manage the collaboration features. A server module enables the access to Protégé features by means of a set of Remote Method Invocations exploited by the Java-based desktop interface. In order to store data, different formats can be exploited by Protégé: textual files, relational databases, RDF, OWL and other XML representations compliant to specific XML Schemas. Protégé core features and interface can be extended by means of plug-ins. Currently there are more than 50

Protégé plug-ins[67] covering a wide set of new features ranging from Semantic Web to Data Formats Issues, Terminology Management and Natural Language Processing.

Relying on the Protégé framework, a browser-based interface, called Web Protégé [50, 51], has been developed. In Web Protégé, ontology editing tasks can be easily performed through a Web browser, without the need to install any additional application. Web Protégé has been realized relying upon the Google Web Toolkit[68] (GWT). GWT is a framework that allows defining in Java the structure of a Web Interface: the framework then is able to generate the HTML/Javascript code of the interface as well as to manage the client-server interactions. The interface of Web Protégé consists of a set of portlets that are specific browsing and editing views over ontological contents (hierarchical views, info views, etc.). Every portlet can be characterized by a certain number of interactions with other portlets. The possibility to easily create, interconnect and customize portlets with respect to the users' ontology visualization needs is essential. The Web Protégé interface implements a limited set of ontology editing features with respect to the desktop interface.

Ontostudio

Web Link: http://www.ontoprise.de/en/home/products/ontostudio/

OntoStudio [52] is a widespread commercial ontology management tool. OntoStudio has been developed by Ontoprise, an independent software vendor based in Germany dealing with applications for semantic knowledge management at enterprise level. OntoStudio is a plug-in of Eclipse, a widely exploited Java-based multi-language software development environment. Thanks to OntoStudio it is possible to exploit a graphical interface for the creation and maintenance of ontologies. Navigation is supported by both tree-based and graph-based views of the ontological entities.

OntoStudio supports globally adopted Semantic Web knowledge representation languages like RDF(S) and OWL as well as the definition of rules by the Rule Interchange Format (RIF). In particular, OntoStudio includes a graphical interface to support rule editing.

SPARQL as well as ObjectLogic can be exploited so as to define queries to semantic datasets. Validation and consistency checking can be performed

[67] http://protegewiki.stanford.edu/wiki/Protege_Plugin_Library
[68] http://code.google.com/intl/it-IT/webtoolkit/

also by specifying of proper consistency rules. Different ontologies can be mapped thanks to the exploitation of a proper visual interface. Knowledge models from external sources like relational databases, spreadsheets, emails as well as from the folder structure of the file system can be imported and mapped to ontologies. The OntoStudio collaborative server provides basic features for the collaborative management of ontologies is an open source licensed version of OntoStudio.

The NeOn Toolkit and the NeOn Project

Web Link: http://neon-toolkit.org/wiki/Main_Page
http://www.neon-project.org/

The NeOn Toolkit is an ontology development environment built on the top of Eclipse[69], a widely exploited Java-based multi-language software development environment composed of an integrated development environment (IDE) and an extensible plug-in system. The core functionalities of the NeOn Toolkit are based on the OntoStudio ontology editor. The NeOn Toolkit has been developed in the context of the NeOn Project, a 4-years initiative co-founded by the European Commission in March 2006, involving 14 European partners. The NeOn project aimed at providing methodological and tool support for the development and the management of the evolution of networked ontologies so as to enable their collaborative development and the contextual adaptation of semantic technologies. Issues related to ontology design patterns, networked ontology design and evolution models as well as ontology localization have been addressed by the NeOn Project. The NeOn Toolkit represents the software legacy of the NeOn Project. The project ended in March 2010. From November 2010, the management and distribution of the NeOn Toolkit is coordinated by the NeOn Technologies Foundation (NTF)[70], a not-for-profit organization.

The NeOn Toolkit supports the different actions that need to be performed to manage the lifecycle of ontologies. By exploiting NeOn, it is possible to carry out tasks ranging from the creation, browsing and refinement of ontologies to the integration of information coming from different knowledge resources, the definitions of mappings between distinct ontologies or the modelling of knowledge by means of rules. The NeOn

[69] http://www.eclipse.org/
[70] http://www.neon-foundation.org/

Toolkit supports the management of ontologies expressed in different languages like F-logic and OWL, including the recent OWL 2 specifications.

Ontology browsing and editing is simplified thanks to the possibility to exploit different perspectives that are particular visualizations of the knowledge formalized in an ontology. Each perspective is optimized for a specific task; for instance the Schema perspective is useful to create, edit and delete ontology objects (concepts, attributes, relations, instances, rules and queries) and the Mapping perspective is optimized for the definition of correspondences between different parts of an ontology. There is also an explicit support for ontology debugging.

Facilities to visualize the evolution of parts of an ontology throughout the editing actions are provided. The definition of ontology rules is supported by both a textual and graphical rule editor and debugger.

In addition to these core features, a consistent number of plug-ins has been developed in order to extend with specific sets of functionalities the NeOn Toolkit. For instance, RaDON (Repair and Diagnoses of Ontology Networks) is a NoOn plug-in that identifies sources of ontology inconsistencies allowing users to automatically or manually repair them on the basis of the results of the diagnosis. CICERO is an argumentation tool to support the collaboration and interaction of both domain experts and ontology engineers. CICERO is implemented as a server storing discussions related to ontological elements or to phases of the ontology development process. LabelTranslator is a plug-in developed to localize ontologies. The localization of an ontology is the translation of its labels in other languages. LabelTranslator exploits the support of external resources as well as to eventually connect the ontological entities to entries of lexical resources. A lot of other plug-ins have been created dealing with visualization, editing, validation, information search and terminology creation from a particular ontology. A comprehensive list and description of these plug-ins can be found in the Neon Toolkit Wiki[71], a wiki devoted to collect all the plug-in related information.

Ontoverse

Web Link: http://www.ontoverse.org/

Ontoverse [53, 54] is a Web-based platform for collaborative ontology design and editing. Ontoverse has been developed by the Heinrich-Heine-University

[71] http://neon-toolkit.org/wiki/Neon_Plugins

Düsseldorf Institute for Computer Science. Ontoverse allows different actors to design and edit ontologies with a great focus on collaboration aspects and group awareness. Every user in Ontoverse can take part in one or more ontology editing projects with distinct roles and permissions. Before starting the actual modelling phase of an ontology, textual wiki pages can be created and collaboratively edited in order to support the definition of ontology design requirements.

The ontology editing interface includes tree-views to browse and edit classes, properties and individuals. For each one of these ontological entities a proper feature editing form is visualized.

Ontoverse implements a fine grained concurrency control. Users can lock parts of an ontology in order to prevent conflicts during their concurrent editing sessions. The interface dynamically shows the parts of an ontology being edited by other users, giving the possibility to interact with them through instant messaging tools if needed. In addition to history tracking and version control, in Ontoverse private workspaces can be defined for a single user or shared between specific groups of users to edit parts of an ontology. Once terminated the editing process, it is possible to merge the outcomes with the complete public version of the same ontology.

Ontoverse interface has been implemented as a Web application. The Java applet technology has been adopted to support ontology visualization and collaborative editing.

TopBraid Composer

Web Link: http://www.topquadrant.com/products/TB_Composer.html

TopBraid Composer is a graphical development environment to model knowledge and integrate data in compliance with Semantic Web information representation standards and best practices. TopBraid Composer is a commercial product developed by TopQuadrant, an international company dealing with semantic knowledge management. TopBraid Composer has been developed in Java as a plug-in on the top of Eclipse, a widely exploited Java-based multi-language software development environment. TopBraid is fully compliant with World Wide Web Consortium Semantic Web standards. RDF-S and OWL ontologies can be created and managed and RDF datasets can be built and queried by exploiting the SPARLQ query language. Also rule based knowledge can be defined by exploiting SPIN, the SPARQL Inference Notation as well as SWRL, the Semantic Web Rule Language.

TopBraid Composer adopts tree-based views to visualize the contents of ontologies and exploits customizable forms in order to edit ontological entities like classes, properties and instances. Alternative graph-based views are also available to visualize ontological contents.

Data can be imported and exported in multiple formats like XMLSchema and RDBMS schemas. Google and Wikipedia can be queried in order to better model knowledge. Inferences and consistency checking over the edited ontologies can be performed by exploiting different OWL-DL inference engines like Pellet, Jena and OWLIM. Collaborative ontology editing features are supported by the CVS check-in/check-out mechanism. Users who need to edit a shared ontology perform their changes over a local copy, then committing the modifications to a shared repository. Changes over ontologies are logged and roll-back actions can be performed.

TopBraid is available to users in a Free Version with limited knowledge editing support and two commercial editions (Standard and Maestro).

CODA

Web Link: http://ubisworld.ai.cs.uni-sb.de/ontology/
http://www.ubisworld.org/

CODA [55, 56] is a collaborative Web-based ontology browsing and editing environment, formerly referred to as UbisEditor. CODA has been developed in the context of the UbisWorld Project at the German Research Centre for Artificial Intelligence. UbisWorld aims at defining tools to support the modelling of real world contexts by ontological formalizations.

CODA allows users, through their Web browser, to navigate and edit hierarchies of ontological classes. Properties and roles of each class are displayed as tree nodes. If a node of a class hierarchy is characterized by a great number of children, a set of intermediate stub nodes is created in order to group the children and better support their visualization and navigation. It is possible to associate multilingual labels to ontological classes.

A user-right management system as well as five-stare rating features for ontological entities have been included in CODA. Currently there is no support for concurrency control and change management. The UbisWorld ontology, built in the context of the same project, can be browsed on-line by exploiting CODA.

CODA has been implemented as a Rich Web Application relying on the DXHTML[72] Javascript library.

SWOOP

Web Link: http://code.google.com/p/swoop/
http://www.mindswap.org/2004/SWOOP/

SWOOP [57] is one of the first examples of feature rich OWL ontology browser and editors. SWOOP was developed at the Maryland Information and Network Dynamics Laboratory of the University of Maryland. The last version of the editor, SWOOP2.3 was released in 2006. It is not under active development anymore. In 2006 the last version of the editor was released, SWOOP 2.3.

SWOOP supports full editing features of OWL ontologies including access to multiple ontologies, collaborative annotations and tracking of changes. In order to simplify the browsing and editing of ontological claims, SWOOP user interface adopts a layout similar to a classical Web browser. An address bar that specifies the location inside the ontology is present together with a main box to show and edit the features of the part of the ontology visualized. There are also buttons to explore the ontology browsing history.

SWOOP was implemented in JAVA by exploiting the Java WebStart technology in order to be accessed by a common Web browser. A plug-in mechanism was defined to easily define and integrate extensions. Among the few plug-ins available, there is one that enables the exploitation of the Pellet reasoner to validate the edited knowledge.

2.2.2.2 Ontology editors and ontology editing methodologies based on controlled languages

Controlled languages can be exploited to express formal ontological assertions by means of natural language statements. This subsection introduces three kinds of controlled language: the Attempto Controlled Language, the CLOnE Controlled Language, and GINO. In particular, the Attempto Controlled Language has been integrated in a wiki environment to support ontology editing tasks. The CLOnE Controlled Language has been exploited to describe the contents of OWL ontologies through natural language generation procedures as well as to edit the same ontologies. This

[72] http://dhtmlx.com/

ontology editing approach is referred to as RoundTrip Ontology Authoring. GINO is the third example of controlled language presented.

The Attempto Controlled Language and ACEwiki

Web Link: http://attempto.ifi.uzh.ch/acewiki/
http://attempto.ifi.uzh.ch/site/

ACEwiki [58, 59] is an example of Wiki controlled language ontology editing system. It has been developed at the Department of Informatics of the University of Zurich. ACEwiki distinguishing feature is the exploitation of a controlled language, the Attempto Controlled English (ACE) [60] to enable users to edit and refine a ontology. The possibility to exploit natural language interactions allows also non expert users to actively contribute in knowledge formalization tasks without the need to have any experience in complex knowledge representation languages.

ACE is a subset of the spoken English, characterized by specific syntax restrictions and a set of interpretation rules useful to translate and formalize ACE natural language sentences into first-order logic representations. Proper one-to-one mappings are defined between ontological entities like individuals, concepts, and properties and linguistic entities like proper names, nouns, of-constructs, and verbs.

Thanks to ACE, users with no background in logic can define formal knowledge in AceWiki. ACE includes a predictive editor of ACE sentences that helps users by showing at any point of the editing process all the syntactical expressions that can be used. Each time a piece of knowledge is added to the system through Controlled English sentences, it is verified against a reasoner. As a consequence, possible inconsistencies with respect to the knowledge already formalized are shown to the user in order to allow reformulations.

The ACE controlled language can be exploited to express sentences describing ontological claims as well as rules. When we have to describe OWL ontologies by means of ACE sentences, since the expressivity of ACE first-order logic is greater than the expressivity of OWL, only the sentences that can be formalized into OWL claims are marked as acquired by the system and thus included in the knowledge base.

In order to make formal rules understandable by everybody, ACErules has been defined. ACErules is a rule system exploiting ACE to specify rules and facts. Ace has been exploited as the controlled language of the FP6 network

of excellence REWERSE[73] (Reasoning on the Web with Rules and Semantics). The ACE parser has been written in Prolog.

The ACEwiki tool has been recently extended so as to support of users' argumentation. In this context ACEwiki has been adopted as an environment to enable policy-making tasks by tracking discussions and formally representing, processing and interlining their natural language contents.

CLOnE and the RoundTrip Ontology Authoring

Web Link: not available

The Controlled Language for Ontology Editing (CLOnE) [61] is as a subset of the English language that allows users to model knowledge and design information by specifying natural language inputs, without having to know complex knowledge representation languages like RDF-S and OWL. CLOnE has been implemented by the English University of Sheffield and the Irish Digital Enterprise Research Institute of Galway.

CLOnE is developed by relying on both standard and customized tools of the GATE platform[74] for Information Extraction and Natural Language Processing. Controlled English natural language sentences are split, tokenized, POS-tagged and lemmatized. Thus GATE specific gazzetters and JAPE rules are applied so as to interpret and translate them as OWL formal claims.

CLOnE has also been used for Natural Language Generation (NLG): it consists in the creation of CLOnE-compliant natural language descriptions of the knowledge formalized by means of an ontology. Rules and templates to generate natural language sentences from OWL ontologies are specified thanks to properly structured XML files.

By exploiting CLOnE NLG features, the RoundTrip Ontology Authoring (ROA) [62] procedure has been defined. The CLOnE-compliant natural language description is generated from starting from the ontology to be authored. The users can edit the generated set of sentences and if needed they can add others. The new CLOnE ontology description is parsed and converted in a new updated version of the ontology. Since users do not need to learn the rules of the controlled language, non-expert users can also easily edit the ontologies. Users can immediately start analyzing and understanding the contents of an ontology since it is expressed in natural language by

[73] http://rewerse.net/
[74] http://gate.ac.uk/

exploiting CLOnE. They can modify and enrich the ontology by adding new sentences on basis of the structure of the sentences already present.

The exploitation of CLOnE-based ROA has been compared to the adoption of common ontology editors like Protégé, in a repeated-measures task-based evaluation. Non-expert users from both research and industry have found most direct and productive to edit ontologies by exploiting a controlled language rather then an ontology editor like Protégé, especially if initial and draft versions of an ontology are considered.

GINO

Web Link: not available

The Guided Input Natural language Ontology editor (GINO) [63] is a controlled language useful to query and edit ontologies and knowledge bases. GINO has been developed by the Dynamic and Distributed Information Systems Group of the University of Zurich. GINO user interface allows querying and editing any OWL ontology by exploiting natural language sentences. The sentences that can be parsed and interpreted by GINO are structured on the basis of a simple static sentence-structure-grammar and are represented by the Backus-Naur-Form notation[75]. The grammar is composed of a set of parsing rules. Rules specify the general structure of the correct sentences. Rules are ontology-independent. An initial set of 120 ontology-independent English rules has been defined in GINO.

Thanks to a grammar compiler module, the ontology-independent grammar rules are extended with ontology-specific rules generated by parsing the OWL ontology that needs to be edited. As a consequence, the complete set of all the grammatically correct natural language sentences that can be formulated by users is defined.

While a user types a sentence, all the possible correct completions are shown by means of popup lists so as to ensure the correctness of the sentence structure with respect to the grammar of GINO. Proper tree-views of classes, individuals and properties of the edited ontology are shown so as to simplify its management. Queries to the ontology are issued in SPARQL. GINO has been implemented in JAVA by exploiting the JENA[76] ontology API to manage and modify OWL ontologies.

[75] http://en.wikipedia.org/wiki/Backus%E2%80%93Naur_Form
[76] http://jena.sourceforge.net/ontology/

2.3 COMPARING KNOWLEDGE EDITORS

In this section the main features of semantic wikis and ontology editor are compared. On the basis of this comparison, the fundamental issues characterizing the design and exploitation of collaborative knowledge editors are summarized.

2.3.1 Analysis of semantic wikis

Semantic wikis enable users to edit in parallel textual contents and some sort of related semantic data structure. All of them are Web-based applications. With the exception of SAVVY wiki, they somehow exploit RDF statements to semantically describe information.

Two approaches are mainly used to *edit semantic meta-data*. The first approach is based on the extension of the Wiki Mark-up Language with new syntactic constructs that support the explicit definition of semantic meta-data. For instance Semantic MediaWiki allows creating internal semantic links by naming the property that relates two wiki-pages. The second approach exploits proper form-based Web interfaces to support users in editing the RDF statements that characterize each wiki-page. The latter solution is adopted by Platypus wiki, IkeWiki and Maariwa. OntoWiki embraces a different methodology to semantically structure information. Unlike others semantic wiki, in OntoWiki every data is represented and stored by means of RDF triples. Different views are available to edit specific typologies of semantic contents.

The *interface* of semantic wikis is usually structured so as to show and browse the semantic information that describes each wiki-page. The layout of these wikis usually includes one or more boxes to show the RDF statements associated to the visualized wiki-page. For instance Platypus wiki displays all the RDF statements that have as subject or as object the considered page inside two areas of the interface surrounding the main textual contents box. Other semantic wikis like Maariwa, inside each wiki-page highlight the portion of textual contents that is described by RDF statements. Part of the considered semantic wikis exploits also OWL ontologies to describe each wiki-page by means of one or more classes. Therefore the set of associated OWL classes is displayed in the layout of wiki-pages. Similarly Semantic MediaWiki supports the connection of wiki categories to ontological classes so as to exploit ontological knowledge to describe wiki contents.

Information search in semantic wikis is usually enhanced by faceted-browsing interfaces that allow users to take advantage of the semantic structure of wiki contents. OntoWiki exploits SPARQL queries to carry out

user searches. The support for SPARQL queries can be enabled in MediaWiki by exploiting a specific extension. Other semantic wikis have defined custom query languages like WikiQL in Semantic MediaWiki and MarQL in Maariwa.

The implementation of semantic wikis has involved a broad range of technologies and programming languages. Platypus wiki, IkeWiki, Maariwa and SweetWiki have been all implemented in Java, Semantic MediaWiki and OntoWiki in PHP, Rizhome in Python and SAVVY wiki in Ruby. Other important implementation aspects are storage, and support for content versioning. Regarding storage, Data are usually stored by exploiting an SQL database, and OntoWiki and SweetWiki use an RDF triple store. Support for content versioning and rollback of modifications are present in the majority of the considered semantic wikis. In Semantic MediaWiki new functionalities can be added by means of software extensions.

2.3.2 Analysis of ontology editors

Since OWL is the ontology representation language globally adopted over the Web, it is not surprising that Most of the considered editors support the management of *OWL ontologies.* Protégé and the NeOn toolkit can also deal with frame-based ontologies.

OWL ontologies are usually characterized by complex knowledge structures. In order to ease the browsing of their contents, specific visualization patterns are exploited. Since browsing the subsumption hierarchies of its classes and properties is one of the most effective ways to get a sense of how an ontology is structured, hierarchies of entities are a widely used to visualize the contents of an ontology. For this reason, the greatest part of the graphical ontology editors adopts *tree-based views* to represent subsumption hierarchies of classes and properties. Part of the ontology editors analyzed supports drag & drop actions to easily rearrange the nodes of these hierarchies. Usually the visualization and editing of the features of ontological classes and properties is performed through proper forms (see Figure 17 for an example in Protégé). Some ontology editor also exploits other visualization patterns such as *graph-based views* to show the contents of an ontology. In graph-based views, ontological classes are represented by graph nodes and properly interconnected by different kinds of relations and properties. The choice of appropriate visualization patterns for ontological contents is a fundamental issue in order to ease the understanding of knowledge structures thus facilitating the task of ontology editing both for knowledge engineers and for users with less experience in knowledge formalization.

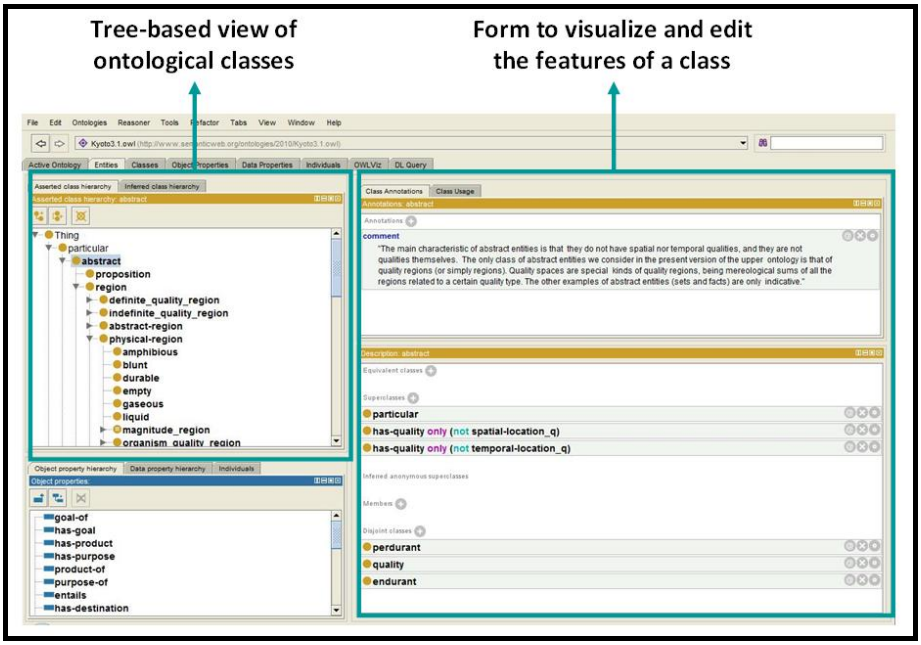


Figure 17: Tree-based view of ontological classes and the class features editing form in Protégé

Most of the considered ontology editors have been implemented as *desktop applications* by exploiting graphical libraries to visualize ontological contents, and only two of them, Web Protégé and CODA, can be accessed by a *browser-based interface*. Web Protégé supports only part of the ontology browsing and editing features of its desktop-based alternative interface. CODA allows performing basic editing actions over ontologies exclusively by means of a Web-based interface. The implementation of an ontology editor as a desktop application has been often preferred because it offers greater ease and flexibility in visualizing the contents and managing knowledge editing patterns. However, the possibility to edit an ontology by means of a web-browser is one of the factors that can enable more extensive user participation in the knowledge editing process since every users through his browser can contribute to the refinement of the knowledge formalized in an ontology.

Collaboration features are supported by the greatest part of the analyzed ontology editors. Protégé, thanks to a client-server architecture support concurrent browsing and editing of ontologies by exploiting the desktop or the browser-based interface. Changes to ontological contents are tracked and annotation of the contents of an ontology is supported so as to support users' argumentation. The NeOn toolkit enables collaboration among the different actors involved in the editing of an ontology by exploiting proper

plug-ins like CICERO. Ontoverse implements collaborative ontology editing by the possibility for users to dynamically interact through a chat in order to discuss about part of an ontology they are concurrently modifying. In Ontostudio, the collaborative server provides basic features for the collaborative management of ontologies. In general, providing proper mechanism to support concurrent and collaborative editing contents is a core feature of ontology editors to enable community of users to cooperate in the knowledge editing activities.

In order to *validate* the edited ontological statements two different approaches are adopted. An external reasoner can be exploited to perform consistency checks over ontologies, like in TopBraid Composer and in SWOOP. Alternatively, a proper set of consistency rules can be specified to check for ontology validity. OntoStudio adopts the latter approach.

Considering the *technologies adopted to implement ontology editors*, it is important to note that Java is used in almost all the cases. The main reasons for the global adoption of Java are the cross-platform support of this programming language and the availability of many Semantic Web libraries and tools developed in Java. Moreover three ontology editors, OntoStudio, the NeOn toolkit and TopBraid Composer have been developed on the top of Eclipse, a widely exploited Java-based multi-language software development environment. Basic versioning features are implemented by most of he tools. Protégé and the NeOn toolkit are extensible through a plug-in mechanism and a considerable number of plug-ins has been developed.

The exploitation of *controlled languages to edit ontologies* represents an important alternative to graphical ontology editors, especially when users without any experience in knowledge engineering need to be involved since natural language interactions are exploited. Ontology editing methodologies based on controlled languages are usually adopted to produce initial draft versions of an ontology that will be refined through a graphical editor. Users often need some training to start using a controlled language, since controlled languages restricts the syntactic construct of the natural language to a limited set of possibilities that can be formalized by means of an ontology. This problem has been partially addressed by the adoption of natural language generation procedures in the RoundTrip Ontology Authoring methodology.

2.3.3 The desirable features of a collaborative knowledge editor

The main issues to be taken into consideration when a collaborative knowledge editor is designed and developed are summarized below:

1. knowledge browsing and editing patterns: on the basis of the typology of users involved in knowledge editing tasks, proper interaction patterns need to be defined. If editing contributions are required from users without any knowledge engineering background, they should be provided with simplified knowledge visualization patterns and editing tools to provide the possibility to deal with the set of complex knowledge structures that often characterize knowledge resources.
2. visualization of knowledge structures: considering the complexity of the data structures usually involved, the possibility to effectively visualize and have a significant global view of the contents of a knowledge resource is fundamental. For this purpose it is important to choose proper widgets like tree-based and graph-based views.
3. concurrency control and consistency checks: concurrent modifications performed by different users have to be correctly managed so as to preserve the consistency of the edited knowledge resources. Proper methodologies should be adopted: they are mainly based on resource locking or conflict reconciliation protocols.
4. community awareness and argumentation: facilities to support users' interactions are relevant to enable their cooperation in knowledge editing tasks. Users should have the possibility to access statistical data concerning the way knowledge resources are socially modified. The most edited knowledge items, the most recent editing actions globally and locally performed on a knowledge resource, and the set of knowledge items that need to be edited or refined represent relevant examples. The availability of tools (discussion threads, instant messaging, etc.) to support users' interactions is also a desirable feature of a collaborative ontology editor in order to discuss how to model ontological knowledge so as to reach consensus.
5. versioning: the different modifications collaboratively performed on a knowledge resource need to be tracked so as to enable the rollback of previous versions. Considering the editing of knowledge resources, the rollback of modifications represents a problematic issue to manage, since their consistency has to be preserved across rollback actions.
6. mapping knowledge resources: in order to support the interoperability among different knowledge resources, a

collaborative knowledge editor has to properly enable users to define mappings between pairs of related items of each knowledge resource.

Aspects relevant for the design of a collaborative knowledge editor are related to user roles and permissions. The definition of restricted groups of users to work on a restricted portion of a knowledge resource can be helpful in some contexts. The possibility to coordinate and organize the knowledge editing actions of different actors on the basis of a workflow model could be also relevant.

In conclusion, during the last few years the diffusion of new Web technologies and interaction patterns has considerably reduced the gap between the interface of browser-based and desktop-based applications. Therefore browser-based interfaces have been more and more chosen as a valid alternative to desktop-based ones to implement collaborative knowledge editors. Browser-based interfaces provide geographically distributed communities of users with ready-to-use knowledge editing environments, even if browser-based knowledge editors still have a limited set of knowledge editing features if compared to the desktop-based versions.

2.4 USER MOTIVATION IN COLLABORATIVE KNOWLEDGE EDITING

One of the main reasons for the success of on-line collaborative contents creation patterns resides in the motivation and involvement huge amounts of users by pushing them to spontaneously contribute. This section tries to better analyze the main reasons enabling massive Web user collaboration.

The three key factors that affect user involvement in social content creation are:

- *increased recognition*: users need to be given credit for their contributions. This is achieved by identifying them through on-line identities and by publicly tracking the modifications they do to the edited contents. In addition, user reputation mechanisms are often exploited in order to evaluate and recognize users' involvement in a community. For instance in *Yahoo Answers*[77] all the contributions are associated to a user and each user has a reputation score related to the quality of his contributions.
- *sense of efficacy*: users need to have the feeling that their contribution to the on-line contents has some immediate feedback. For instance in

[77] http://answers.yahoo.com/

Wikipedia all the changes users perform are immediate and accessible on-line.

- *sense of comm*unity: users have to perceive that they are part of a community and that their editing efforts are useful to the whole community. In *Amazon Products Review*[78] users can review products and other users can take advantage of their ratings saying if it has been useful or not.

Users can be motivated to contribute to the editing of knowledge resource by many ways. Motivation could be achieved:

- *by paying users* through direct engagement, crowd sourcing, etc. This happens in on-line translation services like *OneHourTranslation*[79] or in *Amazon Mechanical Turk*[80].
- *by creating a community of users* highly motivated *and involved in these tasks*, like in free social translation services, in Wikiprojects as *Wikipedia* or *Wikitionary* or in complex collaborative ontology editing environment like *Protégé*[81].
- *by exploiting widespread on-line tasks* that users usually carry out when they browse the Web or *involve users in on-line games* so as to collect their contributions by taking advantage of the collective intelligence.

Below are some examples of Web tools that aim at involving and challenging users in providing useful information by collaboratively editing contents. Some Web applications related to the collection of linguistically relevant information is also considered.

ReCAPTCHA[82] is an application that exploits Web captcha to validate books digitalization errors. Everyday millions of words are manually copied by users all over the Web to validate the filling process of Web forms. ReCAPTCHA exploits this great and distributed amount of work by showing the users images of couples of words taken from the text of a book: one of them is known and it is used for validation purposes; the other word hasn't been recognized by an Optical Character Recognition program. In this way

78 http://www.amazon.com/

79 https://www.onehourtranslation.com/

80 https://www.mturk.com/mturk/welcome

81 http://protege.stanford.edu/

82 http://recaptcha.net/

Web users collaboratively and unconsciously contribute to refine the digitalization process of books.

Google Image Labeler[83] is a collaborative game useful to collect meaningful words that describe specific images. Users are associated with a partner and each pair of users is asked to provide descriptive words for the same image. When users choose the same word to describe an image they gain points. In this way, through an on-line game, Google collects relevant descriptive keywords useful to characterize images and thus also to ease their search. A set of similar online games is accessible at: http://www.gwap.com/gwap/. Here users can associate words to images and music or also guess the word that your on-line game partner is thinking by making him questions.

Google Squared[84] is a Web search engine that provides access to structured contents. Search results are described through a set of distinctive properties (for instance searching for 'frog' gives a list of frog species together with an image, a textual description, the family name, etc.). Google Squared allows users to provide feedbacks about the validity of the information retrieved, to choose the most correct property values as well as to propose new properties describing each single item.

There are also some examples of *Web applications or games to collect or edit linguistically relevant information*.

One of them is the **Verbosity Game**[85], a pair game in which a player has to guess a word by interpreting the textual clues provided by the partner. In this way common sense facts are socially collected so as to train reasoning algorithms. **PhraseDetectives**[86] is another example of online collaborative tool to gather linguistic data. Texts are shown to users who point out the relationships between words and phrases by clicking over the words. The more cross-validated relationships they point out, the more points they get. In this way a rich linguistic corpus is collaboratively created for anaphora resolution training. Other examples of attempts to increase massive user involvement in content production are **Playful Tagging**, an online game to generate folksonomies, and **OntoPair**, a Web game to collaboratively build OWL-based ontologies.

83 http://images.google.com/imagelabeler/

84 http://www.google.com/squared

85 http://www.peekaboom.org/cgi-bin/verbosity/play_game

86 http://anawiki.essex.ac.uk/phrasedetectives/

All these tools underline the importance to consider the need for interactive and friendly interfaces, but also *the necessity to motivate and involve users when an environment for collaborative content editing is designed and realized*. Lexical resources and data collections are often characterized by complex structures that make them difficult to collaboratively edit especially if we consider common Web users.

Massive user motivation and involvement represent a key challenge to address if we want to really open linguistic resources to the world of massive on-line content creation. In this way it would be possible to enable the same communities of users to enrich knowledge resources and keep the resources up to date in order to reflect the changes in the knowledge they describe. In this context, the definition of the best user interaction patterns to exploit represents a fundamental issue to face so as not to demotivate non-expert users. Non-expert users need to be engaged in the editing of knowledge resources by hiding complex knowledge editing tasks behind simpler and easily understandable interaction patterns.

3. WIKYOTO KNOWLEDGE EDITOR: THE COLLABORATIVE WEB ENVIRONMENT TO MANAGE KYOTO KNOWLEDGE RESOURCES

During the last decade the Web has been characterized by an exponential growth of the information accessible on-line together with a great diversification of contents across languages, creation patterns, and typologies. As a consequence, Web users have experienced great difficulties in browsing and dealing with huge quantities of on-line data that are often disconnected and distributed across multiple sources. Therefore the adoption of more structured approaches to organize and access on-line information has become fundamental. Many initiatives have tried to exploit the automated understanding of the meaning of on-line contents so as to make explicit and processable their semantics, thus providing Web users with enhanced possibilities to aggregate data and search for the needed information. Among these initiatives there is KYOTO.

KYOTO is an information and knowledge sharing system that relies on cross-lingual text mining procedures to interpret the meaning of documents in multiple languages. By processing textual contents, KYOTO enables the extraction of deep semantic relations and facts as well as their exploitation across languages and cultures to support users' informative needs. KYOTO has been developed since 2008 in the context of the homonym European FP7 Project. The interested reader is referred to the Official Web Site of the KYOTO Project, http://www.kyoto-project.eu/.

All the knowledge mining tasks of KYOTO are carried out by exploiting the Multilingual Knowledge Base, a collection of lexical and ontological knowledge resources. The knowledge structures formalized in the Multilingual Knowledge Base can be extended and customized with respect to the domain of interest to the community of KYOTO users (i.e. environment, medicine, biology, etc.). The more the knowledge resources of KYOTO are tailored to a particular domain, the more effective the cross-lingual mining of textual contents concerning the considered domain is.

Gathering widespread contributions from the community of KYOTO users, including also domain experts, is essential to extend and refine the knowledge resources of KYOTO by identifying and describing knowledge items relevant to describe the considered domain. The *Wikyoto Knowledge Editor*, called also Wikyoto, has been developed in order enable KYOTO users who usually have little or no experience in knowledge engineering, to easily edit the knowledge structures of the Multilingual Knowledge Base.

Wikyoto represents one of the core components of the KYOTO system. By accessing Wikyoto, users can exploit a visual intuitive interface and simplified language-driven interactions to collaboratively enrich and refine the knowledge resources of KYOTO by hiding the complexity of the edited knowledge structures. The design and the implementation of Wikyoto have taken into consideration the analysis and comparison of the main features of several collaborative knowledge editing environments (see Chapter 2).

This chapter provides an overview of KYOTO and a detailed description of Wikyoto. The first section of the chapter is devoted to analyze some important aspects of KYOTO, such as the knowledge-based approach adopted to perform cross-lingual text-mining, the exploited knowledge resources and the architecture of the system. The second section is focused on Wikyoto. The motivations, the design, the implementation, and the evaluation of the Wikyoto are discussed in detail. Examples of knowledge editing task that can be performed in Wikyoto are provided. Finally, TMEKO, a more experimental component of Wikyoto, useful to support users to easily formalize cross-lingual information by natural language interviews is presented.

3.1 KYOTO: A CROSS-LINGUAL TEXT MINING ENVIRONMENT

This section is divided in three parts. The first part introduces the cross-lingual text mining approach adopted in KYOTO, an approach that will influence the whole architecture of the system. The second part describes in detail the knowledge resources exploited by KYOTO as they are at the basis of all the cross-language knowledge mining tasks of the system. Finally, taking into consideration the issues discussed previously, the third part of the chapter describes the architecture of the KYOTO by describing the main building blocks of the system together with their interactions. Considering the complexity of the KYOTO system, the provided introduction doesn't claim to be exhaustive. To get further information about KYOTO, as well as more details about each one of the different modules composing the system, the interested reader is referred to the KYOTO Project Official Web Site, http://www.kyoto-project.eu/.

3.1.1 Knowledge based cross-lingual text mining in KYOTO

Cross-lingual text mining refers to the *process of automated extraction of high-quality information from textual sources in several languages together with the possibility to homogeneously access and exploits all the mined data*. Cross-lingual support in text mining applications can be achieved by tailoring their structure and mining patterns to a specific language and domain, thus by strongly specializing the application features with respect to every considered language. This approach shows the following relevant drawback: it is usually difficult to deal with cross-language mining patterns as well as to exploit mining and search procedures that can be applied across different languages.

In KYOTO a different approach to cross-lingual text mining is adopted. *A considerable part of the text processing tasks is executed by exploiting language-independent applications that get the linguistic information needed to mine texts in distinct languages by relying on language-specific knowledge resources both generic and domain-specialized*. To support this approach to cross-lingual text mining it is needed to:

- *develop an appropriate domain model* by defining a language-independent set of concepts and relations that describe the domain of specialization of the cross-lingual text mining system. The terms mined from texts in different languages have to refer to these concepts and relations in order to make explicit their meaning. As a consequence, it is possible to apply language-independent (or generic) information extraction patterns to mine useful data from texts in different

languages since the terms of these texts refer the shared language-independent domain model.

- *exploit a Word Sense Disambiguation methodology* in order to find out the meaning of terms mined form texts in different languages and thus to anchor the same terms to the language-independent knowledge representation constituted by the domain model.

There are mainly three different approaches to Word Sense Disambiguation (WSD) [64, 65]:

- Unsupervised WSD: performed by a machine-learning application that processes raw un-annotated collections of texts, called corpora. Unsupervised WSD methods try to discriminate among the meanings of a word on the basis of similarities and differences among the different contexts that characterize the occurrences of the considered word inside the collection of texts. Another approach to unsupervised WSD can be adopted if a collection of texts translated in two or more languages, called parallel corpora, is available. In this case the different meanings of a word in a language are identified on the basis of all its translations available in the other language.
- Supervised WSD: performed by a machine-learning application that chooses the right meaning to be assigned to each term of a text by considering a set of features characterizing the same text. This kind of applications is trained by exploiting a corpus of annotated documents that is a collection of syntactically annotated texts where terms usually have been manually disambiguated by humans. On the basis of this set of correct examples of terms associated to a meaning, the supervised WSD application will generalize the association rules to apply so as to disambiguate terms belonging to new documents.
- Knowledge-based WSD: this group of applications exploits properly structured knowledge resources in order to choose the right meaning of terms inside a document. These resources encode language-specific linguistic features useful to support the disambiguation task. WordNet [20] constitutes one of the most exploited knowledge resources to perform knowledge-based WSD.

Unsupervised WSD does not require an annotated corpus or a specific knowledge resource so as to be performed. Only a raw un-annotated collection of texts is needed in order to cluster the meanings of words.

Supervised WSD usually requires great efforts in terms of working-hours to create a corpus of annotated documents so as to train the disambiguation

system. This is often a time-demanding task done by Natural Language specialists. Moreover the expertise of linguists is frequently needed to manually annotate a corpus of documents since it is not easy to choose the right meaning of a term in a particular context from a set of alternative and often similar ones.

Knowledge-based WSD systems need specific kinds of knowledge resources to disambiguate the meaning of a term. In order to adapt this kind of systems to a specific domain, it could be necessary to *create or enrich the considered knowledge resource so as to include domain specific linguistic information*. This task is often less time-demanding than the creation of manually annotated corpora and a smaller involvement of linguists is required. The resulting knowledge resources can be easily updated and further specialized and refined by exploiting also the contributions of users of the system without any knowledge engineering background.

Usually supervised systems outperform Knowledge-based and unsupervised ones on a specific domain, but as soon as the domain of the considered texts changes, they rapidly decrease in their disambiguation effectiveness. By the expression change of domain it is possible to refer to both a change in the topic of the mined documents and a change in the style of the same documents. The latter can take place when, considering for instance the environmental domain, there is a switch from texts belonging to environmental news to encyclopaedic texts describing environmental issues.

If we compare supervised and knowledge-based approaches to WSD, knowledge-based WSD systems are often outperformed by supervised WSD systems considering their exploitation in a particular domain. But, if we apply a WSD system tailored to a specific domain to a set of texts related to a different domain, usually a knowledge-based system manages to scale and in some cases to improve its effectiveness, while a supervised system tends to collapse. In general, *knowledge-based WSD systems are more stable in performances across different domains since they have the advantage of a larger coverage* [66, 67, 68]. Knowledge-based WSD is usually applicable to all the words in unrestricted text, while supervised techniques are applicable only to those words for which annotated corpora are available [64].

In KYOTO a knowledge based approach to cross-lingual text mining is adopted, relying on *knowledge-based WSD procedures* that exploit proper linguistic information so as to interpret the meaning of textual contents. Figure 18 shows a general example of how KYOTO deals with knowledge based cross-lingual text mining. The terms of texts in several languages ('cat' in English, 'gatto' in Italian, and 'gato' in Spanish) are connected to their language independent representation (the concept of cat) included in the domain model. This connection is created by disambiguating the meaning of these terms thanks to the availability of language-specific linguistic information, called also language-specific knowledge resources, each one encoding the features of a particular language (English, Italian, and Spanish). Knowledge-based WSD is applied to the texts in different languages by exploiting the related language-specific knowledge resource. Therefore the terms 'cat', 'gatto' and 'gato' are linked to the language-independent representation of their shared meaning constituted by the concept of cat in the domain model.

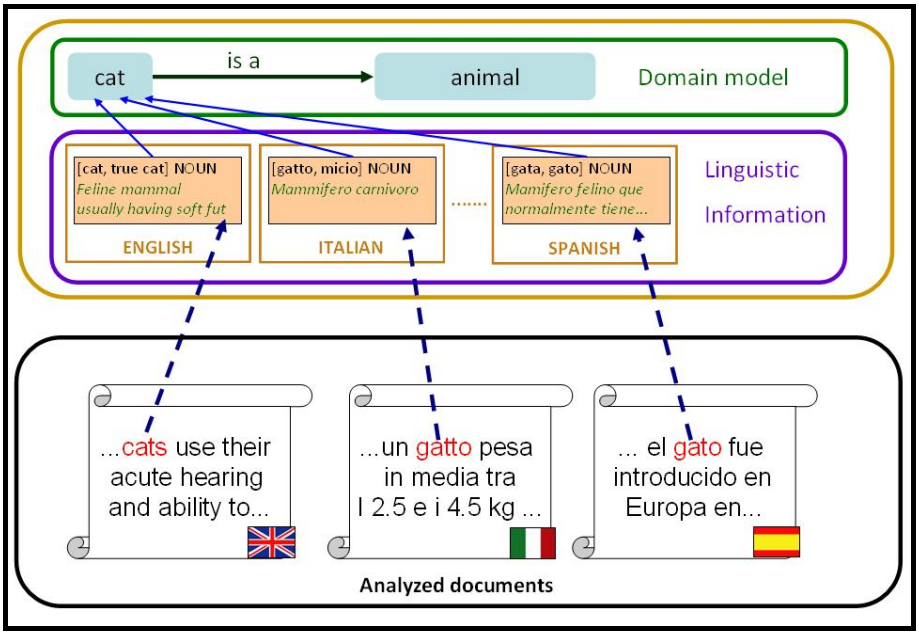


Figure 18: The knowledge-based approach to cross-lingual text mining adopted in KYOTO

3.1.2 The knowledge resources of KYOTO: the Multilingual Knowledge Base

Since KYOTO adopts a knowledge-based approach to cross-lingual text mining, the KYOTO system strongly relies on a the exploitation of a set of knowledge resources in order to interpret the meaning of the terms of texts

in different languages (linguistic resources) and link these terms to their shared language-independent representations included in the domain model.

In KYOTO, the *language-independent domain model is constituted by the KYOTO Central Ontology* that is an OWL DL ontology composed by three layers with a growing level of specialization. The most generic one is constituted by the Dolce-light-Plus Ontology [69] together with the OntoWordNet [70]. The middle layer is made of the set of Base Concepts derived from the Princeton English WordNet 3.0, including about 500 nominal entities. The most specific layer is a collection of concepts and relations that are useful to describe the specific domain of interest the KYOTO system is specialized in [71]. Since in KYOTO the environmental domain has been the test domain, the most specific layer of the KYOTO Central Ontology is currently constituted by a set of entities useful to describe concepts, events, processes and qualities proper to the environmental domain.

The linguistic information characterizing each language involved in KYOTO is encoded in a set of WordNets, one for each language. They are computational lexicons adhering to the WordNet model. As shown in Figure 19, seven European and Asian languages are addressed by the KYOTO Project (English, Spanish, Italian, Dutch, Basque, Chinese and Japanese). For each language KYOTO includes a Generic WordNet (WN in Figure 19) and one or more domain WordNets (DW in Figure 19). The Generic WordNet encodes all the generic linguistic information characteristic of that language. The Domain WordNets can be figured out as domain specializations of the Generic one. It is possible to develop each Domain WordNet independently from the Generic one. Proper mappings can be created so as to link synsets (concepts) from a Domain WordNet to other synsets belonging to the Generic one.

In order to support cross-lingual text mining, for each language the synsets of both the Generic and the Domain WordNets can be mapped to the classes and properties of the KYOTO Central Ontology. Therefore the mapped synsets are described in a language-independent way.

The KYOTO Central Ontology together with the Generic and Domain WordNets of different languages constitute the set of knowledge resources exploited by KYOTO: they are globally referred to as the *Multilingual Knowledge Base*.

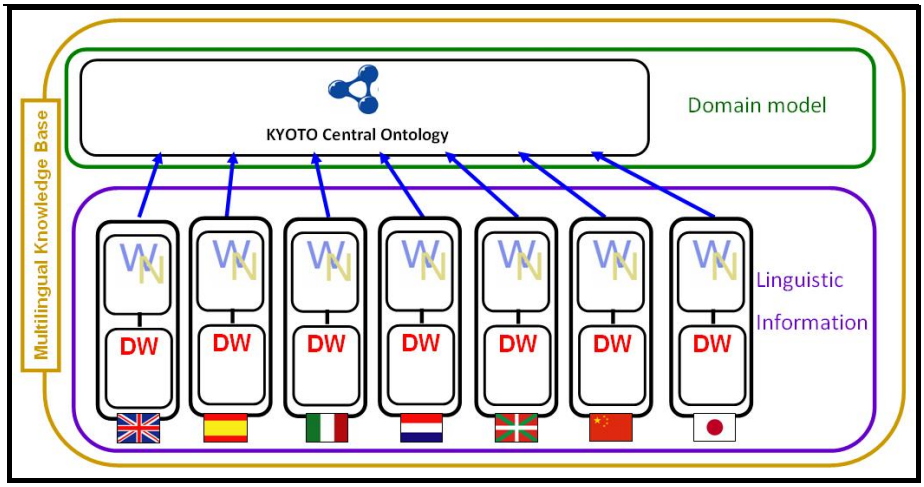


Figure 19: Structure of the Multilingual Knowledge Base

The Knowledge-based WSD methodologies exploited by KYOTO in order to automatically define the meaning of terms are based on the language specific knowledge formalized in both the Generic and the Domain WordNets. Therefore, *the better the structure of a specific domain is described by the related Domain WordNets, the more effective the Word Sense Disambiguation of KYOTO is.*

WSD is performed by applying the UKB algorithm (Graph Based Word Sense Disambiguation and Similarity) [72]. WSD links the terms of the parsed texts to the concepts and relations of the domain model constituted by the Central Ontology. In particular, WSD associates each term to the WordNet synset that better represents its meaning. Thus terms are also indirectly associated to the knowledge structures of the Central Ontology, since WordNet synsets are mapped to language-independent ontological concepts and relations. In this way, language-independent knowledge extraction patterns can be exploited in order to mine useful information from texts in different languages.

3.1.2.1 The data formats of KYOTO knowledge resources: OWL and WordNet-LMF

The adoption of a standardized format to represent both the lexical and ontological resources exploited by KYOTO, stored in the Multilingual Knowledge Base, constitutes a fundamental requirement for the whole KYOTO system. For instance, it allows the integration and exploitation of lexicons that share the same WordNet-like structure. In addition, knowledge resources structured by means of different theoretical and

implementation approaches can be adapted and represented in a standard-compliant way so as to take advantage of them.

The Central Ontology is represented in the Web Ontology Language (OWL DL), the description language globally exploited over the Web to represent ontologies.

So as to comply with the adoption of a standard format, to represent WordNets, in the context of the KYOTO Project, WordNet-LMF (WN-LMF) has been defined. WN-LMF is a dialect of the Lexical Markup Framework (LMF) [73], an ISO standard for the representation of lexical resources (LR). The goals of LMF are to provide a common model for the creation and use of LRs, to manage the exchange of data between and among them, and to enable the merging of a large number of individual resources to form extensive global electronic resources.

In WN-LMF, LMF has been tailored so as to encode lexical resources adhering to the WordNet model of lexical knowledge representation, thus WN-LMF is an example of the practical use of LMF in a real-world application. WN-LMF fully complies with the standard LMF. It builds on the representational devices made available by LMF and tailors them to the specific content requirements of the WordNet model of lexical knowledge representation. An XML Schema for the WN-LMF data model has been specified in KYOTO so as to support the XML-representation representation of LR. To get further information about WN-LMF, the interested reader is referred to [74].

3.1.2.2 Mapping relations among KYOTO knowledge resources

The linguistic and ontological knowledge resources of the KYOTO Multilingual Knowledge Base are interlinked by means of two kinds of mappings:

- from the Domain to the Generic WordNet synsets of each language;
- from Domain and Generic WordNet synsets to the Central Ontology.

In KYOTO, both the Generic and Domain WordNets share the same structure, the WordNet model made of synsets linked by means of a defined set of semantic relations. Domain WordNets represent the specialization with respect to a specific domain of interest of the knowledge formalized in the Generic WordNet of the same language. Even if each Domain WordNet can be developed independently from the Generic one, in KYOTO it is possible to *map Domain WordNet synsets to the corresponding Generic WordNet ones*. A structured and consistent set of mapping from

Domain to Generic WordNets is essential because it constitutes a check for the consistence of the information formalized in Domain WordNets and it supports a more effective execution of the knowledge-based WSD procedures. There are three kinds of mapping relations to connect a Domain WordNet synset to a Generic WordNet one:

- DGM_equivalence: the two synsets are equivalent;
- DGM_hypernym: the Domain WordNet synset is a hyponym (more specific concept) of the Generic WordNet one.
- DGM_hyponym: the Domain WordNet synset is a hypernym (more general concept) of the Generic WordNet one;

Figure 20 schematizes the structure of the mappings between synsets of the Domain and Generic WordNets in the Multilingual Knowledge Base.

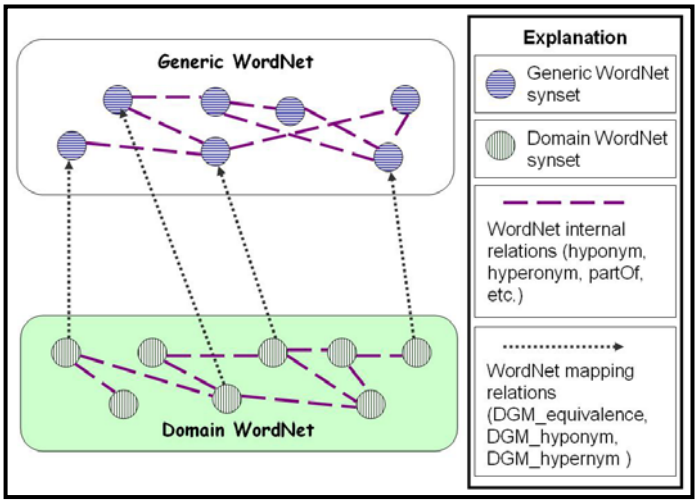


Figure 20: Mapping relations between Domain and Generic WordNet synsets

In order to support cross-lingual text mining in KYOTO the synsets of both the Generic and Domain WordNets of each language can be mapped to the language-independent concepts and relations of the domain model constituted by the Central Ontology. A whole set of mapping relations can be exploited in order to semantically characterize each synset by means of ontological entities [75].

Another mechanism adopted to manage cross-lingual mappings among the synsets of Domain WordNets of different languages exploits the English Domain WordNet synsets as pivot elements. Domain WordNet synsets of languages other than English can be mapped to the corresponding English Domain WordNet synset. Four kinds of mapping relations can be used:

- Equal_synonym: if the synset of the non-English Domain WordNet corresponds exactly to one English Domain WordNet synset;
- Equal_near_synonym: if the synset of the non-English Domain WordNet is mapped to more than one English Domain WordNet synset that represents its meaning;
- Has_hypernym: if the synset of the non-English Domain WordNet is mapped to a more generic English Domain WordNet synset;
- Has_hyponym: if the synset of the non-English Domain WordNet is mapped to a more specific English Domain WordNet synset;

Figure 21 schematizes the structure of the mappings between English and non-English Domain WordNet synsets in the Multilingual Knowledge Base.

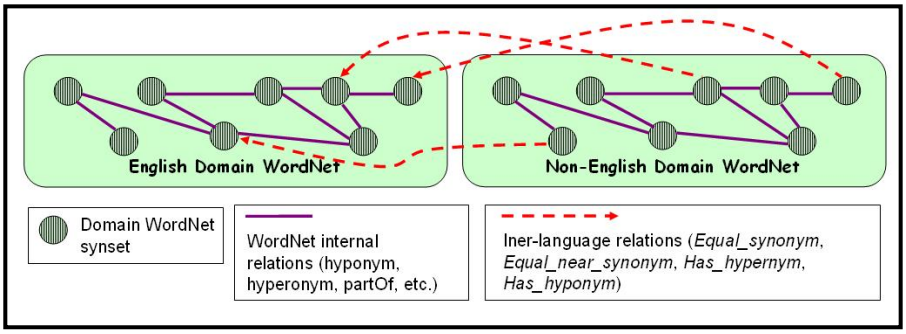


Figure 21: Mapping relations between Domain and Generic WordNet synsets

3.1.3 The architecture of KYOTO

KYOTO is constituted by three components (see Figure 22):

- the *KYOTOCore*, a pipeline made of a set of modules (linguistic tools) for processing textual documents through cross-lingual text mining procedures, extracting facts and terminologies;
- the *KYOTO Multilingual Knowledge Base*, a database optimized for storing the ontological and lexical knowledge resources exploited by the KYOTOCore;
- the *Wikyoto Knowledge Editor*, called also Wikyoto, a wiki environment to collaboratively enrich and maintain the knowledge resources stored in the Multilingual Knowledge Base.

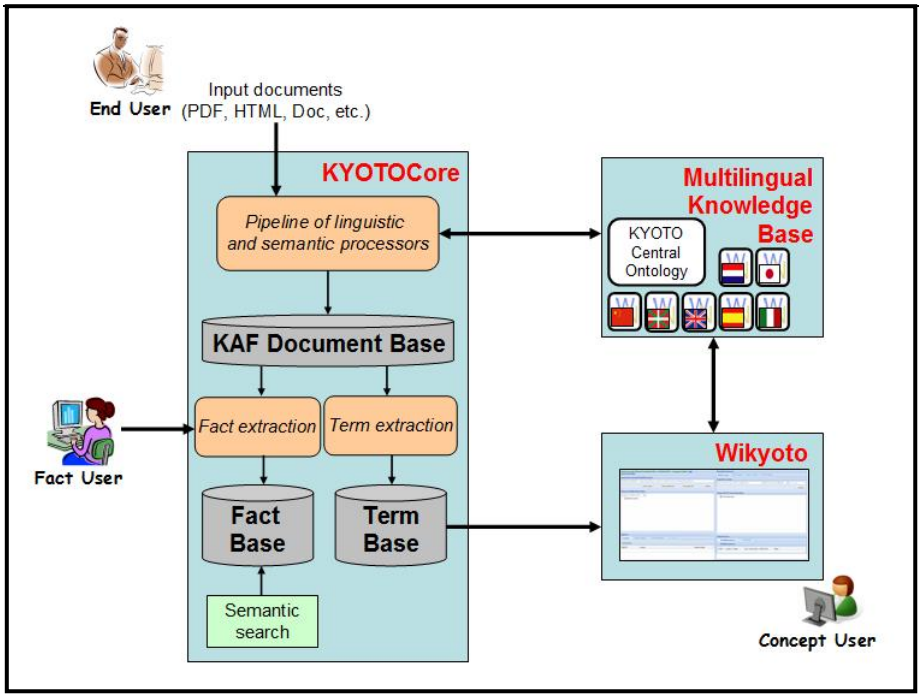


Figure 22: The architecture of the KYOTO system

In KYOTO there are three different kinds of users interacting with the system:

- *End users* propose relevant documents to be processed by the system and perform semantic searches by relying on the information mined by KYOTO;
- *Fact users* are responsible for the definition of new fact extraction patterns, thus improving the fact extraction effectiveness of the whole system;
- *Concept users* access Wikyoto in order to maintain and extend the Multilingual Knowledge Base.

The *Multilingual Knowledge Base* has been introduced at the beginning of the section and the *Wikyoto Knowledge Editor*, the collaborative Web environment to edit KYOTO knowledge resources, will be described in detail in the next section. The rest of this subsection gives an overview of *KYOTOcore*, the set of linguistic tools that extract knowledge from textual documents in different languages is presented starting from the selection of input documents.

Input documents: the textual documents in different languages to be processed by KYOTO can be selected by End users or automatically retrieved from the Web by exploiting crawling procedures. KYOTO supports the analysis of textual documents in different formats (HTML, DOC, PDF, etc.). Once acquired by the system, the documents are converted in raw text in order to be processed.

Pipeline of linguistic and semantic processors: a chain of linguistic tools mines the raw texts by adding textual annotations that specify their syntactic and semantic features. The results of the linguistic and semantic analysis of texts are encoded in KAF [76], the deep semantic annotation format developed in the context of the KYOTO Project.

KAF is a language neutral annotation format that represents both morpho-syntactic and semantic annotation of documents through a layered structure. Starting from the lower of all its annotation layers, where tokens, sentences and paragraph are identified, in KAF each additional layer is built on top of the lower one, referring to its constituent elements. In this way, several levels of text annotation can be added by different linguistic processors. In addition, specialized linguistic processors can be developed to generate incremental annotations for each specific layer.

An XML Schema has been defined to serialize KAF annotated textual document in XML.

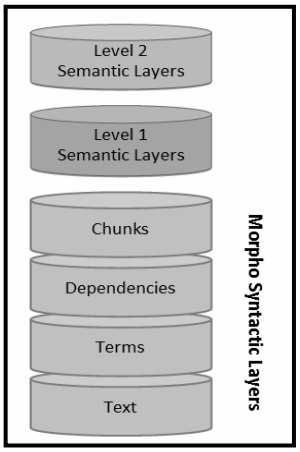


Figure 23: The macro layer of KAF document annotation

In KAF there are three macro-layers of document annotation (see also Figure 23):

- *morpho-syntactic layer* groups all the language-specific text annotations. Tokens, sentences and paragraphs are identified in a

specific document. Terms made of words or multi-words are pointed out, along with their Part Of Speech. Chunks and functional dependencies are also represented in this layer.

- *level-1 semantic layer* includes linear annotation of expressions of time, events, quantities and locations.
- *level-2 semantic layer* is mainly devoted to represent facts, in a non linear annotation context, thus possibly aggregating evidences from the lower layers of multiple textual sources.

In order to get more information about the KYOTO linguistic tools responsible for mining of the different document annotation features represented in KAF, it is possible to refer to the KYOTO Project Official Web Site, http://www.kyoto-project.eu/. It is important to notice that in KYOTO the disambiguation of the terms identified inside a text is carried out by knowledge-based WSD procedures relying on the Generic and Domain WordNets of each language (UKB algorithm, [72]). Each disambiguated term is thus described by means of a KAF annotation that points to the WordNet synset of the related language that better identifies its meaning and therefore to the corresponding language-independent entities of the Central Ontology.

KAF Document Base: stores all the KAF documents annotated by the pipeline of linguistic and semantic processors.

Fact extraction: by parsing the KAF annotated documents, facts are extracted relying upon fact extraction patterns that are specified and refined by Fact Users. Each fact is an event that can be characterized by a set of distinguishing traits including a place and time of occurrence. Fact extraction patterns can exploit the linguistic and semantic features of each KAF annotated document in order to detect a fact. In particular, cross-lingual fact extraction patterns can be specified by exploiting language-independent linguistic and semantic document annotations, for instance the connections of terms to the entities of the Central Ontology. These fact extraction patterns can be applied to KAF annotated documents of multiple languages. The linguistic tool that extracts facts from KAF annotated documents by applying one or more fact extraction patterns is called kybot, Knowledge Yielding roBOT.

Fact Base: the facts extracted from each KAF documents can be included inside the same documents by exploiting the Semantic Level 2 of KAF annotations. But in KYOTO all facts are stored in the Fact Base so as to be easily exploited to perform semantic searches.

Semantic search: is the interface that supports KYOTO End users in the execution of semantic searches over the facts mined by the system.

Term extraction: starting from the KAF annotated texts, term collections, called KYOTO terminologies, are also mined. Each collection is made of a set of terms hierarchically organized. The linguistic tool that extracts terms from KAF annotated documents is called tybot, Term Yielding roBOT.

Term base: constitutes the database where all the KYOTO terminologies are stored. These terminological resources represent a valuable input for Concept Users to refine and extend the knowledge formalized in the Multilingual Knowledge Base by exploiting Wikyoto.

3.2 WIKYOTO KNOWLEDGE EDITOR

This section presents the Wikyoto Knowledge Editor, otherwise known as Wikyoto. Wikyoto is the collaborative Web environment where the multilingual community of KYOTO users interacts to maintain and extend, with respect to their particular domain of interest, the background knowledge of KYOTO, constituting the Multilingual Knowledge Base. Thanks to the adoption of an intuitive visual interface and the exploitation of language-driven interactions, users with little or no experience in knowledge engineering can also contribute to the editing of complex knowledge structures. The possibility to browse knowledge resources different from the Multilingual Knowledge Base is a relevant source of suggestions for users so as to model KYOTO knowledge resources.

This section discusses in detail the core set of issues that have motivated and led the creation of Wikyoto within the KYOTO project (subsection 3.2.1), the requirements and architectural decisions that have influenced the design of Wikyoto (subsection 3.2.2), the implementation of Wikyoto as a browser-based Web application (subsection 3.3.3). The section concludes with an example of exploitation of the use of Wikyoto (subsection 3.3.4), the definition of TMEKO, a methodology useful to support users to easily formalize cross-lingual information by natural language interviews (subsection 3.3.5) and some evaluation data of Wikyoto (subsection 3.3.6).

3.2.1 Collaborative editing of the Multilingual Knowledge Base: motivations

This subsection presents the main motivations that have led to Wikyoto, and thus to the adoption of the collaborative editing paradigm to manage the knowledge resources of KYOTO. The fundamental considerations that stand at the basis of the definition of Wikyoto are:

- KYOTO is exploited in order to parse multilingual documents and mine relevant information concerning a specific domain of interest to the community of KYOTO users (i.e. environment, medicine, biology, etc.).
- KYOTO is a knowledge-based cross-lingual text mining system, thus it strongly relies on the availability of well-structured knowledge resources characterized by a wide coverage of the domain addressed. Therefore, it is essential to tailor the knowledge resources of KYOTO with respect to the considered domain in order to effectively exploit the system.
- Knowledge engineering skills are required to consistently edit the complex knowledge structures that characterize the knowledge resources of KYOTO so as to enrich and refine them. However, knowledge engineers usually do not have any knowledge of the considered domain.
- Users interested in a particular domain and domain experts should be involved in the identification and description of knowledge items that characterize their domain of interest. However, KYOTO users usually do not have any knowledge engineering background, thus they cannot directly formalize the domain knowledge.

On the basis of the considerations exposed, knowledge engineers can be involved to gather contributions from KYOTO users in order to extend and customize KYOTO knowledge resources with respect to a particular domain. They have to interact with KYOTO users, especially domain experts, so as to collect informal descriptions of the considered domain. Then knowledge engineers need to formally represent the gathered information so as to include it in the knowledge resources of KYOTO, thus tailoring these resources to the considered domain. This approach requires a great involvement of knowledge engineers and is usually time-demanding. Moreover, since the domain knowledge is often characterized by continuous evolution, knowledge engineers need to keep a constant interaction with KYOTO users in order to track changes and collect new information describing the domain so as to periodically update the knowledge formalized in the knowledge resources of KYOTO.

KYOTO adopts an alternative approach. *The collaborative editing paradigm is exploited to enable users to directly edit and maintain KYOTO knowledge resources so as to extend and customize them with respect to the considered domain of interest*. This goal is achieved by means of the Wikyoto Knowledge Editor, also called also Wikyoto. As stated above,

Wikyoto is a collaborative Web tool that allows users with no expertise in knowledge engineering to edit the knowledge resources used by KYOTO.

In Wikyoto, *users point out knowledge objects of their interest by collaboratively tailoring the knowledge resources of KYOTO to a specific domain*. By doing so, they implicitly define their informative needs. Since users tend to formalize knowledge structures they consider relevant, the text mining procedures of KYOTO will be automatically targeted to parse with greater accuracy that kind of information from texts, making it available to semantic searches. Therefore the resulting domain-customized knowledge resources are *biased towards the point of view and the needs of a particular community of users*.

It is important to consider that different communities of users may view the same knowledge objects from different points of view. For example, a group of environmentalists and a group of wood selling companies would probably refer to the concept “maple wood” in very different ways. If both communities were developing their own Domain WordNets in Wikyoto, each community would probably create a “maple wood” synset, but would define it very differently. The environmentalists might define the new synset as a sub-concept of the “wood” concept intended as “the hard fibrous lignified substance under the bark of trees”. On the contrary, the community of wood selling companies might define the “maple wood” as a kind of “sale product”, because they would be interested in “maple wood” as a sale good. Wikyoto enables the KYOTO system to take into consideration these differences in use.

Taking into account the needs and the characteristics of the users’ community, and the requirements of collaborative knowledge editing environments presented in Chapter 2, Wikyoto should fulfil two fundamental high-level requirements:

- the adoption of simplified visualization and user interaction patterns to enable users without experience in knowledge formalization to easily browse and edit the information included in the knowledge resources of KYOTO;
- the possibility to perform browsing and editing actions over knowledge resources without any time and space constraint, since the KYOTO users are usually spread across different locations.

Wikyoto visualization and user interaction patterns have been derived from the detailed analysis of the editing actions that should be performed by KYOTO users on the basis of the structure of the knowledge resources of the Multilingual Knowledge Base. The choice to implement Wikyoto as Web

application allows to access to Wikyoto over the Web by means of a browser, without any time and space constraint.

3.2.2 Shaping Wikyoto: system design issues

This subsection discusses in detail the main issues that have influenced and motivated the design of Wikyoto.

3.2.2.1 Editing actions targeted to gather linguistic information useful to support cross-lingual text mining

The aim of all the knowledge editing actions of Wikyoto is to gather linguistic knowledge useful to support the cross-lingual mining of textual document in the KYOTO system.

There are different linguistic knowledge types or features that can be exploited to support cross-lingual text mining. They can be specified during the annotation of the texts of a corpus or formalized and included in knowledge bases. The latter approach is the adopted in KYOTO.

A general list of linguistic knowledge features that can be exploited to improve text mining is presented [77]:

- *frequency of senses*: on the basis of a specific corpus, it represents the number of times a particular sense occurs;
- *part of speech*: it describes the way a word is used inside a particular context usually defined by the sentence it occurs in. Examples of parts of speech are: verb, noun, pronoun, adjective, adverb, preposition, conjunction and interjection. Depending on its context of usage, a word can have different parts of speech. For instance the word 'books' is a noun in the sentence "Books are made of paper" and a verb in the sentence "She books two tickets".
- *morphology*: morphological information mainly refers to the association to a lemma of all its forms. For instance 'runs' and 'running' are possible forms of the lemma 'run'.
- *collocational information*: collocations are groups of words that belong to a given semantic domain and occur with mutual expectancy greater than chance. Examples of possible collocations made of pairs of words are 'nuclear family' and 'cosmetic surgery'.
- *semantic properties of words*: this expression refers to the whole set of properties that are useful to semantically characterize a word. It is possible to identify the three following properties:

  - *taxonomical organization*: it is the organization of words into hierarchies where the semantic relationship between a specific word and all its related ones is well defined. The most exploited kind of semantic relationship is the IS-A that identifies subsumption hierarchies of entities. For instance, a 'dog' can be represented as a child word of 'animal' to express that a 'dog' is a kind of 'animal'.
  - *situation or topic*: it represents the association of a meaning to a given word on the basis of its context of usage. For instance the word 'mouse' in an animal-related context can represent a kind of animals, but the same word in a computer-related context can refer to the widespread computer pointing device.
  - *argument-head relation*: it represents the semantic relation holding between the head of a phrase and all the related arguments. The head of a phrase is the word that determines the syntactic type of the phrase. For instance, in the noun phrase "the dog in the street", the head "dog" is specified by the prepositional phrase "in the street".

- *syntactic cues*: they are usually related to the valency of verbs, in other words, to the number of predicates that can characterize a verb. For instance specifying that a verb is transitive or not can determine if the verb can hold or not object predicates.
- *semantic roles*: they refer to the classification of the arguments of natural language predicates into a closed set of participant types which were thought to have a special status in grammar. Examples of participant types are: agent, patient, instrument, locative, temporal, manner, coarse, etc. For instance in the sentence "John cleans the room" the predicate is represented by the verb "clean": "John" represents the Agent and "the room" represents the Patient.
- *selectional preferences*: by this term we denote a word's tendency to co-occur with words that belong to certain lexical sets. For example, the adjective "delicious" prefers to modify nouns that denote food and the verb "marry" prefers subjects and objects that denote humans. The definition of selectional preferences is usually a language-dependent task.
- *domain*: it represents the general topic that characterizes the use of a word. For instance if we are reading a text about programming languages, we are dealing with the domain of 'computer science'.

In order to create and refine knowledge resources useful to support cross-lingual text mining, collaborative editing environments need to enable users in the definition and formalization of a relevant and consistent subset of the linguistic features just described. On the basis of the analysis carried out in Chapter 2, *current editing tools for knowledge resources are mainly targeted to manage ontologies and usually do not explicitly support the possibility to gather this kind of linguistic information or they need to be extensively customized for these tasks*. The approach most adopted to collect cross-lingual information consists in the simple association of labels in different languages to a concept.

In order to enrich the KYOTO Multilingual Knowledge Base with information useful to support cross-lingual text mining, Wikyoto users need:

- to edit the Domain WordNets of their language so as to customize them with respect to the considered domain of interest. In this way the effectiveness and coverage of Word Sense Disambiguation is enhanced. In particular, users should be supported in:
  - the creation of new Domain WordNet synsets or the editing of existing Domain WordNet synsets by modifying their set of lemmas, part of speech, and gloss as well as by evaluating their rigidity;
  - the creation of new semantic relations between Domain WordNet synsets or the modification of the existing semantic relations;
  - the creation of mappings between Domain WordNet synsets and Generic WordNet synsets.
- to map the Domain WordNet synsets of their language to the language-independent entities of the Central Ontology. Therefore the cross-lingual text mining of KYOTO is improved.

Considering the general list of linguistic knowledge features that can be exploited to support text mining, thanks to Wikyoto, KYOTO users can provide several kinds of linguistic information. Wikyoto enables the definition of the *part of speech* of synsets chosen among noun, verb, adjective and adverb. In Wikyoto synsets can be linked by semantic relations of different kinds. By mapping Domain WordNet synsets to Generic WordNet ones, each Domain WordNet synset can be characterized by a set of Base Concepts together with the their semantic type thus defining its *situation of use*. The mappings that can be instantiated from Domain WordNet synsets to the Central Ontology are also useful to specify the *semantic roles* of the same synsets with respect to ontological classes and properties.

Besides the linguistic information already considered, in Wikyoto it is also essential to support users to evaluating the rigidity of Domain WordNet synsets. Determining if a synset is rigid or non-rigid allows to check the consistency of the hyponymy/hypernymy hierarchies of synsets and represents the necessary prerequisite to ontologize the same synset by creating proper mappings to the Central Ontology.

3.2.2.2 Exploitation of external knowledge resources to enrich the Multilingual Knowledge Base: the KYOTO Terminology, SKOS Thesauri, and DBpedia

All over the Web it is possible to access several information items and knowledge structures that can constitute useful suggestions to enrich or refine the knowledge formalized in the Multilingual Knowledge Base. Therefore, in Wikyoto, users have been enabled to browse a set of external knowledge resources and import data into the Multilingual Knowledge Base. These knowledge resources are:

- *KYOTO Terminology*: the set of terminological collections mined from the documents parsed by KYOTO. Each collection is made of a set of terms hierarchically organized. Term hierarchies represent valid information to enrich Domain WordNets with new synsets. An example of terminology, related to frog species is shown in Figure 24.
- *SKOS Thesauri*: the Simple Knowledge Organization Format (SKOS) is a data model conceived to define, share on-line and link knowledge organization systems such as thesauri, taxonomies, classification schemes and subject heading systems. SKOS is a World Wide Web Consortium (W3C) Recommendation[87], developed in the context of the W3C Semantic Web Activity Working Group. By exploiting SKOS, concepts can be identified by using URIs. Lexical strings, called also labels, and textual descriptions can be assigned to each concept to identify it in one or more natural languages. Different concepts can be interlinked so as to define informal hierarchies and association networks. Each concept can be associated to more general (broader), ore specific (narrower) and related (relatedTo) concepts. In order to express SKOS data through RDF triples has been defined a SKOS RDF Schema, so as to semantically specify a proper SKOS modelling vocabulary. SKOS Thesauri can be published over the as RDF datasets that can be queried by exploiting SPAQRQL. SKOS concept networks represent valid resources to enrich Domain WordNets with new

[87] W3C SKOS Recommendation - http://www.w3.org/TR/2009/REC-skos-reference-20090818/

synsets. Figure 25 shows a portion of a SKOS thesaurus related to the description of frogs.

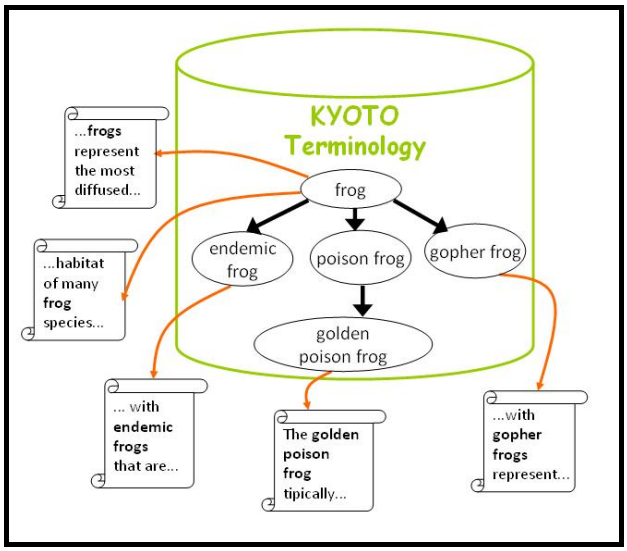


Figure 24: External knowledge resources: KYOTO Terminology

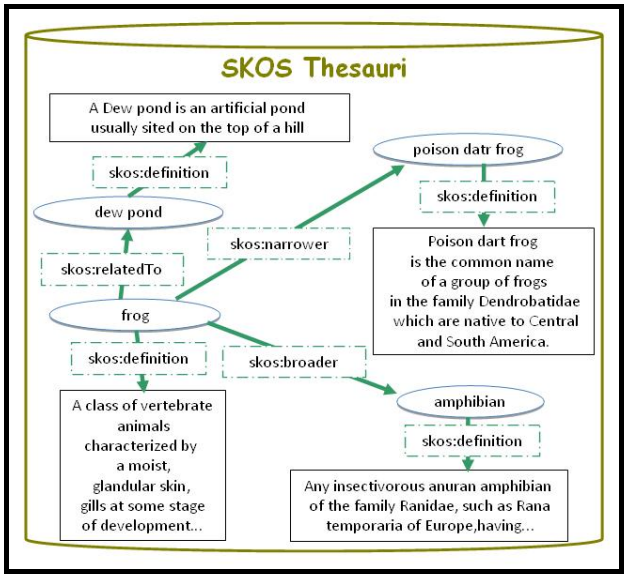


Figure 25: External knowledge resources: SKOS Thesauri

- *DBpedia*[88]: includes a lot of structured contents mined from Wikipedia and represented as sets of RDF triples. All these RDF datasets can be queried on-line by exploiting SPARQL. In Wikyoto, DBpedia is queried

[88] Dbpedia - http://wiki.dbpedia.org/

in order to retrieve and import the definition (gloss) of new Domain WordNet synsets. Table 2 includes some statistical data about DBpedia.

Total number of entities described by label and abstract (in one of more languages)	3,5 million
Total number of RDF triples	672 million
Total number of links to external Web resources	5,9 million
Total number of links to other RDF datasets	6,5 million

Table 2: Statistical data about DBpedia

3.2.2.3 Intuitive visualization and simplified language-driven user interaction patterns to browse and edit knowledge resources

A fundamental requirement that stands at the basis of the development of Wikyoto is the need to enable users without any experience in knowledge representation to easily provide their editing contributions to the knowledge resources of KYOTO. In order to involve users in knowledge editing tasks it is needed to keep as shallow as possible the learning curve of Wikyoto. KYOTO users should easily learn how to browse the structure of the Multilingual Knowledge Base and edit its contents. Three approaches have been adopted in order to achieve this goal in Wikyoto:

***A) Intuitive visualization widgets* so as to browse the knowledge structures of the Multilingual Knowledge Base.**

In several typologies of knowledge resources, *hierarchical as well as graph-based patterns constitute the most adopted structures to relate knowledge items* (concepts, properties, etc.). In the Multilingual Knowledge Base, WordNet synsets can be linked by several types of semantic relations. Among them, the hyponymy/hypernymy relations are fundamental to support WSD algorithms. Moreover, this kind of information is usually easy to understand for users: synsets are related each other with respect to their level of specificity (the concept of 'cat' is related to 'animal' since it is a most specific concept).

Considering the analysis and comparison of several knowledge editing tools (refer to Chapter 2), *tree-based views* constitute the visualization pattern globally adopted to represent the hierarchical structure often characterizing the items of knowledge resources. Tree-based views provide an effective mean to get an idea and browse the structure of knowledge resources. Tree-based views have been adopted also in Wikyoto in order to browse the semantic relations between synsets (see Figure 26). In particular, in

Wikyoto the focus has been put on effectively dealing with hyponymy/hypernymy relations, even though other WordNet semantic relations can also be browsed and edited. In Wikyoto, tree-based views are also exploited to navigate the contents of knowledge resources external to the Multilingual Knowledge Base like the KYOTO Terminologies, SKOS Thesauri and the Central Ontology.

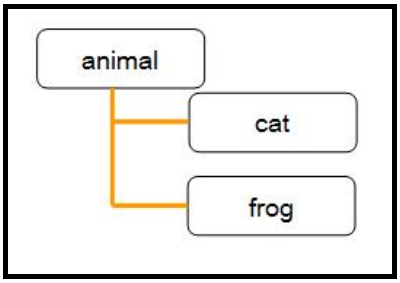


Figure 26: Example of tree-based view of WordNet synsets (hyponymy/hypernymy relations)

In addition to the adoption of tree-based views, the interface of Wikyoto has been properly structured to provide users with a visual overview of all the necessary information describing each synset while browsing WordNets.

B) Intuitive interaction patterns* to deal with external knowledge resources: *drag&drop

In Wikyoto it is possible to browse knowledge resources external to the Multilingual Knowledge Base like the KYOTO Terminologies and SKOS Thesauri. Tree-base views are exploited to navigate their hierarchies of items, respectively terms and concepts. Domain WordNets synsets can be mapped on these resources. Single items or sub-hierarchies of items of these resources can be imported in Domain WordNets thus creating a corresponding hierarchy of synsets.

So as to simplify as much as possible these operations, Wikyoto interface exploits drag&drop interactions. Terms from the KYOTO Terminology or concepts from SKOS thesauri can be dragged over the tree-based views of Domain WordNet synsets in order to map these concepts/terms over the selected synset or to create new synsets. Figure 27 shows an example of a drag&drop action to create the 'tree frog' WordNet synset from an item of an external resource is shown. The drag&drop of parts of hierarchies of terms and concepts is also possible so as to build the same structure in WordNet.

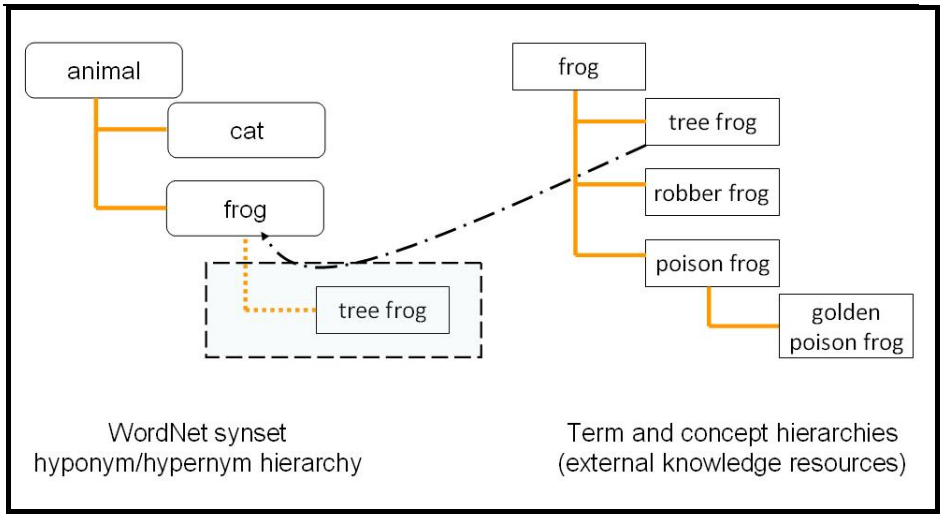


Figure 27: Drag&drop of items (tree frog) from hierarchies of external resources to hierarchies of WordNet synsets

C) Language-driven interactions to help users in the most difficult knowledge editing tasks: the Rigidity interview and the TMEKO procedure

Some knowledge editing task cannot be simplified by exploiting knowledge visualization facilities so as to enable users with no expertise in knowledge engineering to perform it by means of Wikyoto. This happens when a user needs:

- to define the rigidity value of a Domain WordNet synset;
- to map Domain WordNet synsets to the KYOTO Central Ontology.

In these knowledge editing contexts, *natural language based user interactions are exploited*. In this way, users feel more comfortable since they have to deal with the language as they use it in everyday life without handling complex knowledge structures. In particular:

- the *Rigidity interview* has been defined to easily define the rigidity value of a Domain WordNet synset by answering two yes/no questions.
- the *TMEKO procedure* has been defined to drive users in properly defining mappings from Domain WordNet synsets to the KYOTO Central Ontology. TMEKO has been developed as a more experimental component of Wikyoto. TMEKO exploits natural-language user interactions including also user interviews.

3.2.2.4 Need for a collaborative tool balancing between semantic wikis and ontology editors

During the last few years, several collaborative knowledge editing environments have been developed as desktop or Web based applications. Considering the analysis carried out in Chapter 2, they can be divided into two groups:

- semantic wikis (lightweight environments): they are usually Web based tools that can be easily accessed and exploited by users with little or no expertise in knowledge formalization. These environments mainly support users in making explicit computer-processable structured knowledge starting from unstructured texts. As a consequence, the formalized knowledge can be exploited to improve information navigation and search.
- ontology editors (formal environments): they are usually adopted by knowledge engineers in order to fully model information according to a specific, well known knowledge representational schema. Due to their complex knowledge editing patterns, these tools are mainly realized as standalone desktop applications even if in some case they can be also accessed by a Web interface.

Semantic wikis often expose visual intuitive Web interfaces so as to simplify knowledge editing tasks for users, without any experience in knowledge formalization. Moreover these environments frequently exploit natural language to represent information by asking users to point out structured entities from textual documents. As a consequence, users are required to deal with the language as they use it in everyday life. By exploiting this kind of tools it can be difficult to edit complex knowledge structures or it can be elaborate and sometimes impossible to customize these tools to support these tasks. As a consequence, *semantic wiki can hardly be exploited to manage the complex knowledge structures needed by cross-lingual text mining*.

By contrast, *ontology editors* are exploitable mainly by knowledge engineers since the considered knowledge structures are usually too complex to be edited by users without any experience in knowledge formalization. Therefore *ontology editors are not the best choice to gather social contributions from common Web users*. Moreover *these tools usually aim at editing generic knowledge structures that need to be specialized and often have not been conceived in order to model the linguistic features of natural language exploited by cross-lingual text mining systems*.

In these two groups of tools collaborative knowledge editing features are supported with different approaches and methodologies, as discussed in Chapter 2.

Wikyoto represents *a balance between semantic wikis and ontology editors*. Wikyoto aims at building a Web environment with a shallow learning curve for users without experience in knowledge formalization, like in semantic wikis. By exploiting an interactive Web interface, users are invited to enrich the linguistic knowledge of Domain WordNet starting from natural language texts and interviews, but also by browsing domain terminologies. However, like in ontology editors, the resultant knowledge has a level of formalization useful to support the cross-lingual knowledge mining tasks of KYOTO.

The diagram of Figure 28 places over the 'complexity' and 'level of knowledge structuring' axes ontology editors, semantic wiki and Wikyoto. Wikyoto tries to balance between the features of both of them.

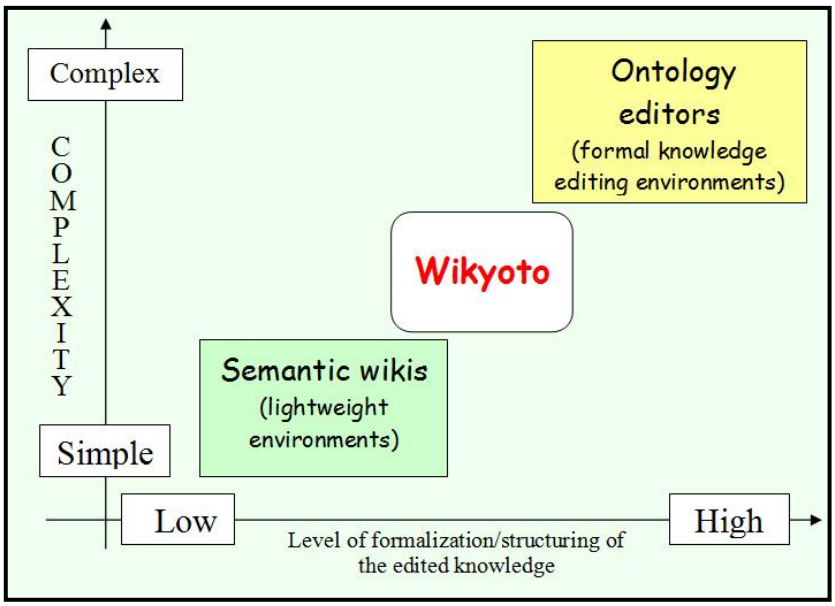


Figure 28: Diagram of collaborative knowledge editing environments on 'complexity' and 'level of knowledge structuring' axes

3.2.3 Wikyoto architecture and implementation

This subsection presents the architecture and implementation of Wikyoto that has been defined on the basis of the set of design issues and

requirements concerning the collaborative editing of KYOTO knowledge resources previously analyzed.

A *browser-based Web application architecture* has been adopted so as to allow geographically distributed communities of users to access Wikyoto and collaborate to the editing of the Multilingual Knowledge Base without any time or space constraint. By means of their Web browser, Wikyoto users should be enabled to perform concurrent editing actions over KYOTO knowledge resources as well as to access and browse external knowledge resources so as to get relevant knowledge editing suggestions. Therefore the typical software design concerns of browser-based real time collaborative editing system have to be faced in the implementation of Wikyoto: are presented in subsection 3.2.3.1.

In subsection 3.2.3.2 the global architecture of Wikyoto is introduced by identifying the main modules, their interactions as well as by describing their implementation choices.

Some fundamental aspects of the system architecture are then discussed in more detail. In particular, subsection 3.2.3.3 describes the Data Repositories of Wikyoto. Subsection 3.2.3.4 discusses the browser based part of Wikyoto constituted by the Web based interface and the client-side Javascript logic. The layout and interaction patterns characterizing the interface represent a core component of the whole system to support users to effectively navigate and edit KYOTO knowledge resources. AJAX based Javascript interactions are exploited to access and modify the Multilingual Knowledge Base as well as a set of knowledge resources external to KYOTO.

Finally, Subsection 3.2.3.5 introduces how data consistency across concurrent modifications is managed in Wikyoto.

A working prototype of Wikyoto is available at: http://www.wikyoto.net/. In addition the Web site provides a detailed description of the system, the user manual and a set of video-tutorials.

3.2.3.1 Software design concerns of browser-based real-time collaborative editing systems

Wikyoto is a browser-based real-time collaborative editing system. A collaborative editing system is a computer system enabling a geographically distributed group of users to edit some sort of shared content. It is a specific typology of computer supported cooperative work. Collaborative knowledge editing environments are a particular kind of collaborative editing systems in which the edited contents are constituted by knowledge resources or by deep or shallow structured contents useful to represent

some aspect of formalized knowledge. Collaborative-editing systems are referred to as real-time or synchronous systems when multiple users can concurrently perform changes to the shared contents and these changes are immediately propagated to all the other actors. They are called Web based systems when they can be accessed through a Web interface, usually through a Web browser (in this case these systems can also be called browser-based).

The software design of Web based collaborative editing systems involves several concerns. The most important ones are:

- *client-server partitioning*: in Web based systems, it refers to the division between the server and the clients of the different processing tasks to be performed. In recent Web applications, data manipulation tasks are more and more performed client-side using Javascript.
- *data aggregation*: is the collection and integration of data needed by the collaborative system by accessing to distinct sources. Current Web applications often perform this task also client-side by Javascript.
- *client-side logic*: strictly related to client-server partitioning and data aggregation, refers to the definition of the data elaborations that should be performed client-side by Javascript, the choice of the best design patterns to adopt and the selection of the most adequate Javascript tools and libraries to exploit.
- *data consistency and management of concurrent modifications*: includes the set of strategies to adopt to manage the consistency of the shared documents across concurrent modifications. When several users edit a document in parallel, their concurrent modifications may cause conflicts. A good strategy need to be defined to avoid or deal with conflicting concurrent modifications.

In Wikyoto the above software design concerns have been faced. Specific solutions have been proposed and implemented.

3.2.3.2 The architecture of Wikyoto

Wikyoto has been structured as a Rich Web Application that strongly relies on client-side Web technologies, languages, and standards that are widely supported by the majority of Web browser. HTML and CSS are exploited to define the layout and graphical features of the interface and Javascript to enable client-side user interactions and data elaborations. In particular, Javascript is extensively exploited in order to:

- provide users with an interface characterized by visual widgets and intuitive interaction patterns;
- retrieve, aggregate, display, and modify data by accessing multiple repositories.

Consequently the high level architecture of Wikyoto is constituted by two main building blocks:

- *browser module*, which is constituted by the browser-based Web interface and a set of Javascript libraries that support the interactivity of the interface as well as the retrieval of information from the data repositories;
- *data repositories* that store the set of knowledge resources accessible by Wikyoto. Each data repository exposes a proper Web based Applications Programming Interface to enable the Javascript libraries running on the Web browser to access to the data stored by the same repositories.

The main data repository of Wikyoto is the *Knowledge Base DR* that stores the Generic and Domain WordNets of the Multilingual Knowledge Base by supporting browsing and editing actions. All the other data repositories allow browsing knowledge resources and knowledge structures, without any possibility to modify them. They are:

- the *KYOTO Terminology DR*, which provides access to the terminologies automatically mined from the documents processed by KYOTO;
- the *KYOTO Central Ontology DR*, which exposes the contents of the KYOTO Central Ontology;
- *SKOS Thesauri DR*, which enable the access to SKOS Thesauri;
- *DBpedia*, a semantic Web dataset that exposes on-line the structured information mined from Wikipedia.

The Knowledge Base DR, the KYOTO Terminology DR and the KYOTO Central Ontology DR constitute parts of the KYOTO system, thus they are called *KYOTO Data Repositories*. By contrast, the SKOS Thesauri DR and DBpedia are external to the KYOTO system, thus they are called *External Data Repositories*. In Figure 29 schematizes the global architecture of Wikyoto.

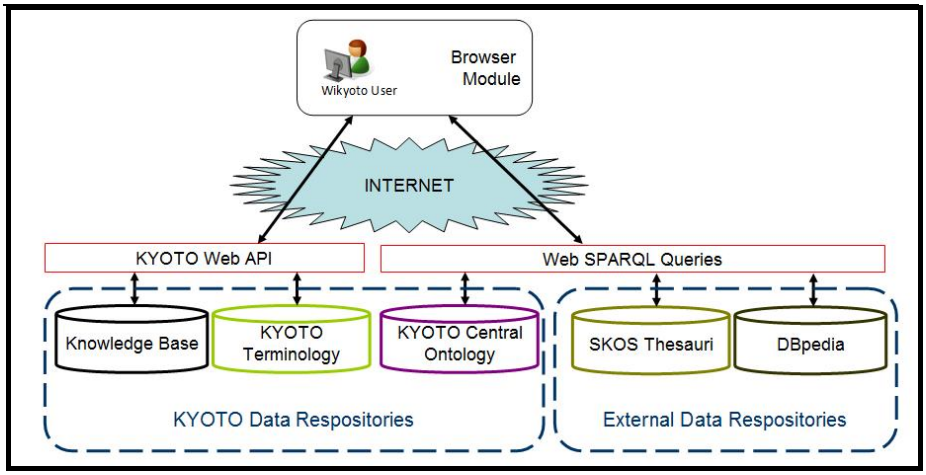


Figure 29: Global architecture of Wikyoto

The Browser module can access each Data Repository by means of Web AJAX[89] calls (Asynchronous Javascript And XML) performed through the Javascript code. Each Web AJAX call invokes a specific methods (called Web API call) exposed on-line by a Data Repository in order to retrieve or modify its data.

In particular, each Data Repository exposes on-line a Web API (Application Programming Interface). A Web API is composed by a set of methods, the Web API calls. Each call can be invoked over the Web by external agents as the Browser module. Each Web API call is useful to perform specific data retrieval or data storage operations over the datasets managed by the considered Data Repository. Web API calls are characterized by:

- a *Web URL* where the request of execution of the call has to be sent.
- the *request parameters*, usually specified by means of a set of name/value pairs. The request parameters are used to define the operations that the involved Data Repository is required to perform.
- the *response*, which is the set of data sent back by the considered Data Repository when a Web API call is invoked by the Browser Module. The response data can represent a portion of the knowledge stored in the same Data Repository and/or can specify the outcome (success/failure) of the operation requested by invoking the related API call. The response data formats most adopted are XML and JSON[90].

[89] Introduction to AJAX in Wikipedia - http://en.wikipedia.org/wiki/Ajax_%28programming%29
[90] JSON - http://www.json.org/

In KYOTO, the methods of the Web API of the Knowledge Base DR and the KYOTO Terminology DR have been defined and customized with respect to the needs of Wikyoto. Instead the interactions with the KYOTO Central Ontology DR, the SKOS Thesauri DR and DBpedia are carried out in a standard way. All the knowledge resources they manage are RDF datasets and can be accessed to perform custom searches. Therefore SPARQL is exploited in order to issues queries. SPARQL[91] is the query language for RDF graphs standardized by the W3C.

3.2.3.3 Data Repositories

The Data Repositories can be divided into two groups: *KYOTO Data Repositories* and *External Data Repositories*. For each data repository is described the set of knowledge resources it manages, how it is exploited by the Browser Module, and some implementation details.

KYOTO Data Repositories

These data repositories are internal to the KYOTO system and accessible, with the exception of the KYOTO Ontology Server, through a custom set of Web API calls. There are three KYOTO Data Repositories: the Knowledge Base DR, the Terminology DR and the KYOTO Ontology DR.

Knowledge Base DR

The Knowledge Base DR manages the WordNets of different languages constituting the linguistic resources of KYOTO. Together with the KYOTO Central Ontology, the set of WordNets is referred to as the Multilingual Knowledge Base. The Knowledge Base DR exposes a Web API customized to supports both Generic and Domain WordNet browsing as well as the possibility to perform any kind of modifications of the Domain WordNets (creation of new synsets or refinement of existing ones, refinement of internal relations, mapping of Domain synsets to Generic WordNet ones, to the KYOTO Central Ontology or to other external resources). Currently the Knowledge Base DR stores the Generic and Domain WordNets of the seven languages involved in the KYOTO Project: English, Dutch, Italian, Spanish, Basque, Simplified Mandarin Chinese and Japanese. In addition, the Knowledge Base DR manages the registration and authentication of the users of Wikyoto. The Knowledge Base DR has been built by customizing

[91] SPARQL W3C Recommendation - http://www.w3.org/TR/rdf-sparql-query/

DEBVisDic[92] [78], a server platform to store and provide online access to the linguistic knowledge resources.

KYOTO Terminology DR

The KYOTO Terminology DR stores all the term hierarchies mined by parsing textual contents in KYOTO. All these data constitute an important source of suggestions for KYOTO users so as to extend or refine Domain WordNets. KYOTO users can browse KYOTO terminologies by accessing the Knowledge Editor. In a similar way to that of the Knowledge Base DR, the KYOTO Terminology DR exposes a Web API customized to support the browsing of term hierarchies. Thanks to the KYOTO Terminology DR Web API calls it is possible to search for terms, to retrieve all the children terms of a specific term and all the relevant features that describe each term.

KYOTO Ontology DR

The KYOTO Ontology DR stores and provides access to the KYOTO Central Ontology. This data repository is based on the VIRTUOSO Server Open Source Edition[93]. Since the Central Ontology is formalized in OWL-DL and serialized as a set of RDF triples, the same ontology has been stored in VIRTUOSO as a RDF dataset and can be queried exploiting a SPARQL Endpoint. The Browser Module of Wikyoto issues proper SPARQL queries directly to the SPARQL Endpoint of the VIRTUOSO Server in order to support users in browsing the KYOTO Central Ontology.

External Data Repositories

The navigation of External Data Repositories allows KYOTO users to retrieve useful suggestions to enrich and refine the Multilingual Knowledge Base. Both the External Data Repositories considered are constituted by RDF datasets (SKOS Thesauri and DBpedia). Therefore they can be accessed by the Browser Module of Wikyoto through SPARQL queries.

SKOS Thesauri DR

Currently in Wikyoto, users can access and browse four SKOS Thesauri representing multilingual environmental data collections since the environmental domain represents the application domain chosen to test the KYOTO system. These SKOS Thesauri have been respectively derived or refined starting from the following data collections:

[92] DebVisDic Web Site - http://deb.fi.muni.cz/index.php

[93] VIRTUOSO Server Open-source Edition: http://virtuoso.openlinksw.com/dataspace/dav/wiki/Main/

- *General Multilingual Environmental Thesaurus* (GEMET)[94], developed as an indexing, retrieval and control tool for the European Topic Centre on Catalogue of Data Sources (ETC/CDS) and the European Environment Agency (EEA);
- *Species 2000*[95], a project supported by the European Union jointly with other academic and environmental organizations, aiming at creating and keeping updated a validated checklist and classification of all the world's species (plants, animals, fungi and microbes);
- *Habitat types from EUNIS Biodiversity Database*[96], developed by the European Environment Agency; it contains collections of habitats, species and sites across Europe;
- *WWF Ecoregions Database*[97], developed by the World Wildlife Found (WWF); it is a collection of the Earth's most biologically diverse and representative terrestrial, freshwater, and marine habitats-areas where the Earth's natural wealth is most distinctive and rich, where its loss will be most severely felt, and where we must fight the hardest for conservation.

The aforementioned SKOS thesauri have been represented as RDF datasets and loaded into the VIRTUOSO Server Open Source Edition. Therefore they have been exposed over the Web through a proper VIRTUOSO SPARQL Endpoint so as to be queried on-line by the Browser Module of Wikyoto.

DBpedia

All the RDF datasets of DBpedia, maintained and update by the Institute of Informatics of the University of Leipzig (Germany), are exposed on-line through a public VIRTUOSO SPARQL Endpoint[98]. Therefore the Browser Module of Wikyoto issues SPARQL queries to the DBpedia in order to retrieve specific content snippets; in particular, in Wikyoto, it is possible to search in DBpedia for a definition of a synset and eventually import or refine it.

94 GEMET Web Site: http://www.eionet.europa.eu/gemet

95 Species2000 Web Site: http://www.sp2000.org/

96 Eunis Biodiversity Database: http://eunis.eea.europa.eu/

97 WWF Ecoregions Database: http://www.worldwildlife.org/wildplaces/about.cfm

98 DBpedia VIRTUOSO Public SPARQL Endpoint: http://dbpedia.org/sparql

3.2.3.4 Browser Module: Wikyoto interface and Javascript libraries

The Browser Module of Wikyoto is devoted:

- to realize the set of intuitive data visualization patterns and simplified language-driven user interactions previously identified (3.2.2.3) in order to enable users without any knowledge engineering expertise to provide linguistic knowledge useful to support cross-lingual text mining in KYOTO;
- to interact with the Data Repositories by issuing Web AJAX calls so as to retrieve, integrate, and edit the considered knowledge resources.

In the first part of this subsection the layout of the Web interface of Wikyoto is presented. In the second part the Javascript logic of the Browser Module of Wikyoto, based on the Ext.js library, is introduced.

Wikyoto interface layout

Once authenticated and after having chosen a language to browse and enrich the corresponding Domain WordNet, users access the Web interface of Wikyoto (see Figure 30).

The interface is divided into two parts: on the right side there is the *Domain WordNet Browser and Editor* and on the left side the *Semantic Resources Browser*.

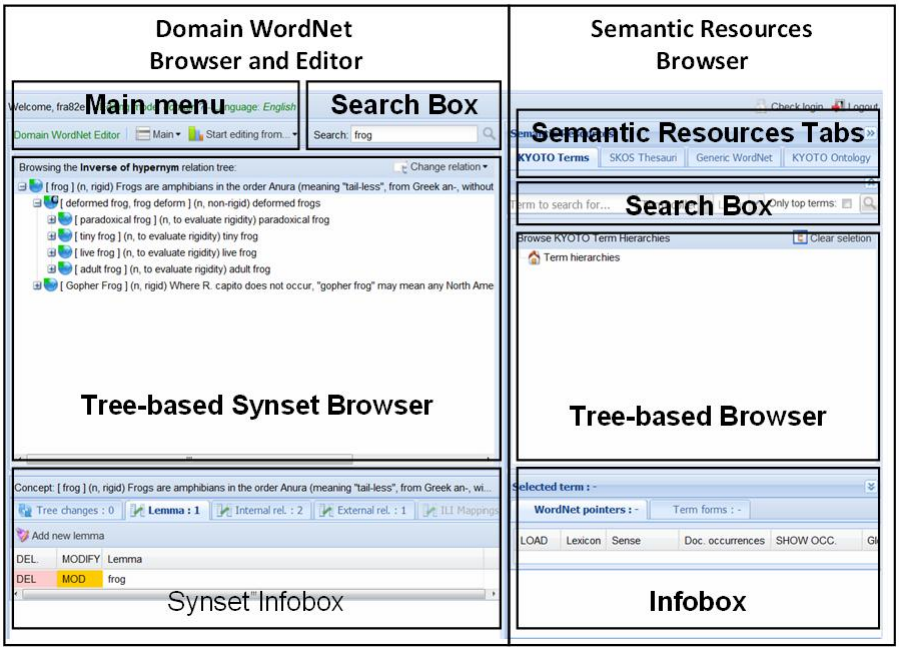


Figure 30: Layout of the interface of Wikyoto

Users can browse and edit the Domain WordNet of their language of choice through the *Domain WordNet Browser and Editor* by accessing:

- the *Main Menu* that provides access to global functionalities like the download of the Domain WordNet in the XML WN-LMF format defined in KYOTO or also the possibility to visualize statistical information about the collaborative editing of the same Domain WordNet (recently added synsets, most edited synsets, etc.).
- the *Search Box*, to search for all the Domain WordNet synsets identified by a particular lemma
- the *Tree-based Synset Browser*, to visualize hierarchies of synsets with respect to a specific semantic relation (i.e. hyponymy/hypernymy, meronymy/holonymy, etc.).
- the *Synset Infobox* to visualize and edit all the information characterizing the synset node selected in the *Tree-based Synset Browser*.

The Semantic Resource Browser enables users to browse four kinds of Semantic Resources with different purposes. The Semantic Resources Tab allows switching among the four different browsing interfaces:

- *KYOTO Terms*: browse the KYOTO terminologies so as to search for relevant suggestions to structure Domain WordNet by adding new synsets.
- *SKOS Thesauri*: navigate the four SKOS Thesauri accessible in KYOTO so as to get relevant suggestions to enrich Domain WordNets.
- *Generic WordNet*: browse the Generic WordNet of the language chosen by the user during his authentication to Wikyoto so as to instantiate mappings from Domain to Generic WordNet synsets.
- *KYOTO Ontology*: browse the classes and properties of the KYOTO Central Ontology so as to instantiate mappings from Domain WordNet synsets to the KYOTO Central Ontology.

The general structure of the four browsing interfaces of Semantic Resources is similar and composed of:

- *Search Box*: to search for a term inside the KYOTO terminologies, a concept inside a SKOS Thesaurus, a synset inside the Generic WordNet or a class/property inside the KYOTO Central Ontology.
- *Tree-based Browser*: to navigate hierarchies of terms, concepts, Generic WordNet synsets, and ontological classes and properties.

- *Infobox*: to visualize the information describing the term, concept, Generic WordNet synset, ontological class or ontological property corresponding to the node selected in the *Tree-based browser*.

  The tree nodes corresponding to a term, a concept, a Generic WordNet synset, an ontological class or an ontological property can be dragged and dropped over a Domain WordNet synset tree node in the *Tree-based Synset Browser*.

  If a tree node of a term or a concept is dragged over a Domain WordNet synset tree node, it is possible:

  - to create a mapping relation from the considered node to the selected Domain WordNet synset by adding the name of the node as a new lemma to the same synset.
  - to create a new synset disconnected from other Domain WordNet synsets or as an hyponym of the selected Domain WordNet synset. The name of the node becomes the lemma of the new synset. If present, the definition of the node becomes the gloss of the new synset (the considered SKOS thesauri often include the definition of concepts).

- Generic WordNet synset node: create a mapping from the selected Domain WordNet synset to the Generic WordNet synset node. The user is asked to choose the appropriate mapping relation.
- Ontological class or property node: creating a mapping from the selected Domain WordNet synset to the ontological class/property node.

In Figure 31, an example of a term from a KYOTO terminology, 'poison frog' dragged over the Domain WordNet synset 'frog' is shown. In this situation probably the user wants to enrich the classification of frogs in the Domain WordNet by adding a new 'poison frog' synset.

A more detailed description of the features of Wikyoto is available at: http://www.wikyoto.net/. The tool can be accessed through the same Web site. Subsection 3.2.4 describes a practical example of usage of Wikyoto providing the description of a set of actions usually performed by exploiting the interface of Wikyoto.

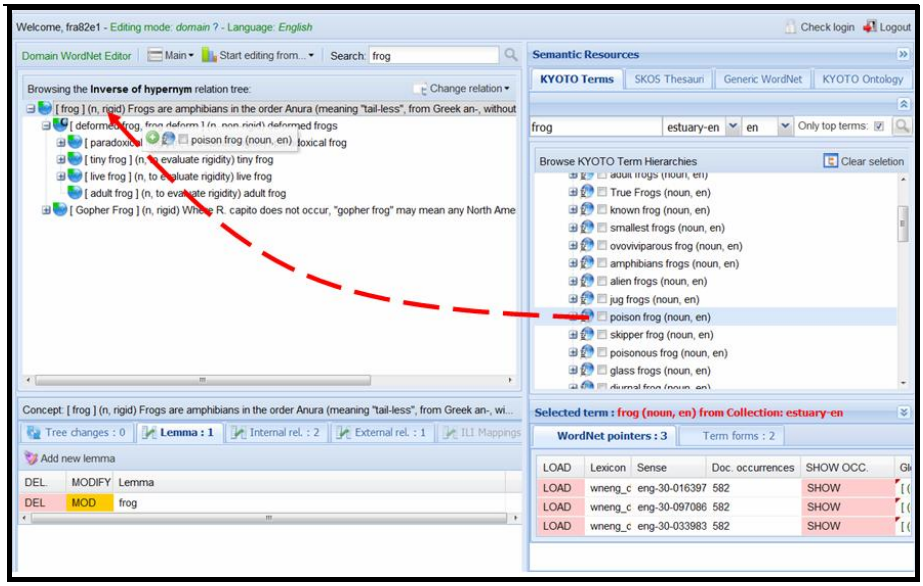


Figure 31: Drag&drop of nodes in the interface of Wikyoto

Javascript client-side elaborations

Javascript, the most widespread browser scripting language, has been strongly exploited to implement the Browser Module of Wikyoto. In particular, Javascript has been used to support the creation of the Web interface and to manage the interactions of the Browser Module with the Data Repositories.

Many Javascript libraries are currently available to develop highly interactive Web interfaces by providing users with programming facilities to define complex layouts and graphical widgets to create easy and intuitive data visualization patterns. Javascript libraries often include also functionalities to ease client-server data exchanges based on Web AJAX calls as well to create and manage a small browser data cache to temporary store the data retrieved from the server.

The Javascript library used for the implementation of the Browser Module of Wikyoto had to meet some requirements. In particular, it had to provide:

- the possibility to define a strongly structured layout for the interface;
- the support for a wide range of customizable data-visualization widget and intuitive user interaction patterns and in particular the management of tree-based views and the drag&drop of nodes across multiple trees;

- the possibility to freely use the library in open-source applications like Wikyoto.

Considering these requirements, the Ext.js[99] Javascript library was chosen. Ext.js is a commercial Javascript library extensively adopted all over the Web in a many contexts. Ext.js can be exploited for free under the GNU GPLv3 Licence[100] in open source applications like Wikyoto.

The Javascript code of the Browser Module of Wikyoto is composed of (see Figure 32):

- the *Ext.js Javascript library*;
- the *Wikyoto-Ext.js Extensions*, including all the components of the Ext.js library that have been extended or customized in order to be exploitable in the Browser Module;
- the *Web API Wrappers*, that are Javascript libraries that provide a set of Javascript functions useful to access and interact with the different Data Repositories;
- the *User Interface Logic*, the collection of all the Javascript code useful to manage the layout and the interaction patterns of the Web interface of the Browser Module.

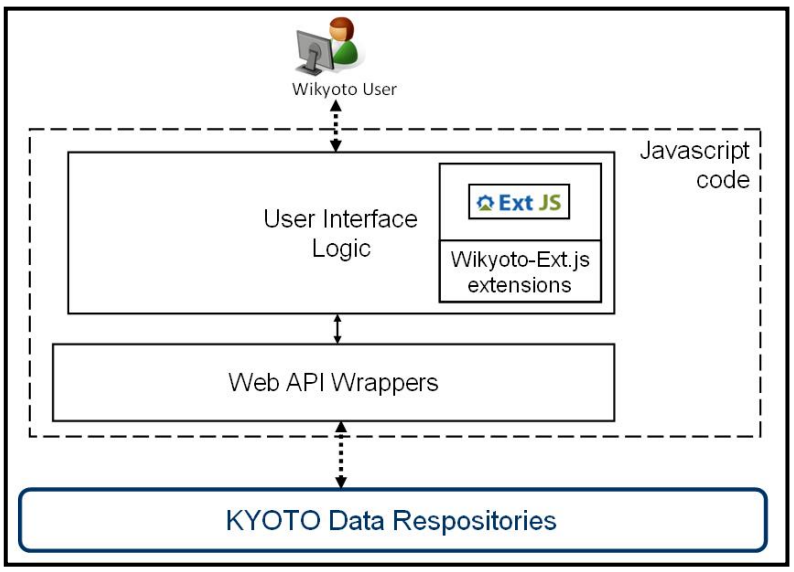


Figure 32: Global structure of the Javascript code of Wikyoto

[99] Ext.js Web Site - http://www.sencha.com/products/extjs/
[100] The GNU GPLv3 Licence - http://www.gnu.org/copyleft/gpl.html

The *Web API Wrappers* are useful to provide the Javascript code with a uniform interface (set of Javascript functions) to access to the following Data Repositories of Wikyoto: the Knowledge Base DR, the SKOS Thesauri DR and the KYOTO Central Ontology DR. Each Web Wrapper directly manages by means of Web AJAX calls the interactions with the Web API exposed by the Data Repository.

The structure of the Javascript code of the *User Interface Logic* is organized on the basis of the layout and the features of the same interface. Thus the code has been partitioned into the following functional units:

- Global interface management;
- Domain WordNet browsing and editing;
- KYOTO Terminology browsing;
- SKOS Thesauri Browsing;
- Generic WordNet browsing;
- KYOTO Central Ontology browsing;
- Interface update;
- Concurrency management.

The Browser Module of Wikyoto represents a complex object-oriented Javascript application. Apart from external libraries like Ext.js, the Browser Module includes about 30000 lines of Javascript code spread over 30 files. In order to ease the management and increase code structuring and modularity, Javascript objects have been logically grouped into a structured hierarchy of Javascript namespaces that are all placed under the global 'Wikyoto' namespace. Also the division of the Javascript code into files has been based on the functional role of the different portions of code.

3.2.3.5 Managing concurrent editing actions in Wikyoto

A fundamental software design concern of Wikyoto is represented by the management consistency across concurrent modifications done by different users to the shared knowledge resources stored in the Multilingual Knowledge Base. A strategy to deal with these modifications is needed in order to solve concurrent and potentially conflicting changes and maintain consistency among the different replicas of portions of the knowledge resources visualized by users on their browser.

In general the management of concurrency and consistency is a core issue in real-time collaborative editing systems [79]. Three possible strategies commonly adopted are briefly introduced here.

- *Pessimistic locking*

In order to preserve the consistency of a collaboratively edited resource it is possible for every actor, before starting editing, to require a lock over it. In this way the editor prevents concurrent editing actions from other actors thus avoiding any form of inconsistency. Once finished the editing session, the same actor has to release its lock so as to make the shared document editable by others. This kind of locking is called pessimistic because it assumes that it is highly likely that many users will cause inconsistencies by editing the same resources at the same time.

- *Optimistic locking*

Optimistic locking assumes that multiple editing actions of a shared resource can complete without affecting each other. In this case every actor of a collaborative editing system can start editing a copy of the shared resource without any delay. When a user has ended its editing actions he tries to commit the changes. The system verifies if the editing actions performed by the user are not conflicting with the others concurrently performed by other actors. If any conflict rises, different approaches can be adopted, but usually some conflict resolution strategy is defined in order to go over inconsistencies. Optimistic locking ensures a high responsiveness of the collaborative editing system, it can cause the loss of important document modifications, and it can require additional user involvement if some conflict rises. MediaWiki[101], the free wiki software exploited to develop several wiki projects like Wikipedia, exploits the optimistic locking approach to manage concurrent modifications over its pages / documents.

- *Operational transformation*

The operational transformation is a technique to support a wide range of functionalities, including concurrency and consistency management in collaborative editing software [80, 81]. Operational transformation has often been exploited to collaborative edit textual documents since it has proven to be particularly adequate for this kind of shared resources. Operational transformation does not require locking and supports any number of users.

[101] MediaWiki Web Site: http://www.mediawiki.org/wiki/MediaWiki

By operational transformation, all the users of a collaborative editing system can edit a local copy of the document and experiment great system responsiveness regardless of the network lag of the system. Indeed, all the modifications done are immediately included in the local copy of the document without having to wait any kind of remote acknowledgment. The local editing actions of each actor are then propagated to the remote replicas of the document hold by others. If needed, local editing actions are transformed so as to be adapted to the state of every remote replica of the shared document. One of the most diffused Web-based applications that exploit the operational transformation technique is Google Docs & SpreadSheet[102]. Also Google Wave[103] concurrency control system is based on operational transformations.

Considering the structure of the knowledge resources edited by means of Wikyoto, the *pessimistic locking approach has been adopted*. The use of optimistic locking would have avoided any problem of lock request/release, but it would have strongly complicated Wikyoto with the need to define conflict resolution strategies. The operational transformation paradigm, often exploited to collaboratively edit textual documents, would have required a strong adaptation so as to be applied to complex knowledge structures like the ones of the Multilingual Knowledge Base.

In Wikyoto we have decided to support the possibility to manage locks at synset level. Before performing editing actions over a synset, the lock on the same synset has to be obtained and release once the same editing actions are performed. The request and release of the lock over a synset are transparent to the user. These operations are managed in background by Wikyoto. If a user wants to edit a synset locked by another user, he is notified and his editing action is aborted. From the Web interface of Wikyoto, a proper icon of the Tree-based synset browser shows users if a synset is locked or not (see Figure 33). The choice to manage fine-grained locks at synset level consistently reduces the risk that the editing actions performed by different users collide.

3.2.3.6 The implementation technologies of Wikyoto

Considering the global architecture of Wikyoto just described, the main implementation technologies adopted in the Data Repositories are summarized below (see Figure 33):

- Browser Module: HTML, CSS, Javascript (Ext.js Javascript library)

[102] Google Docs & Spreadsheet - https://docs.google.com/
[103] Google Wave - https://wave.google.com/wave/

- *KYOTO Knowledge DR*: servlet Ruby and BERKELEY XML DBMS;
- *KYOTO Terminology DR*: PHP scripts and MYSQL DBMS;
- *KYOTO Central Ontology DR, SKOS Thesauri DR,* and *DBpedia*: VIRTUOSO Server Open Source Edition accessed by means of SPARQL Endpoints.

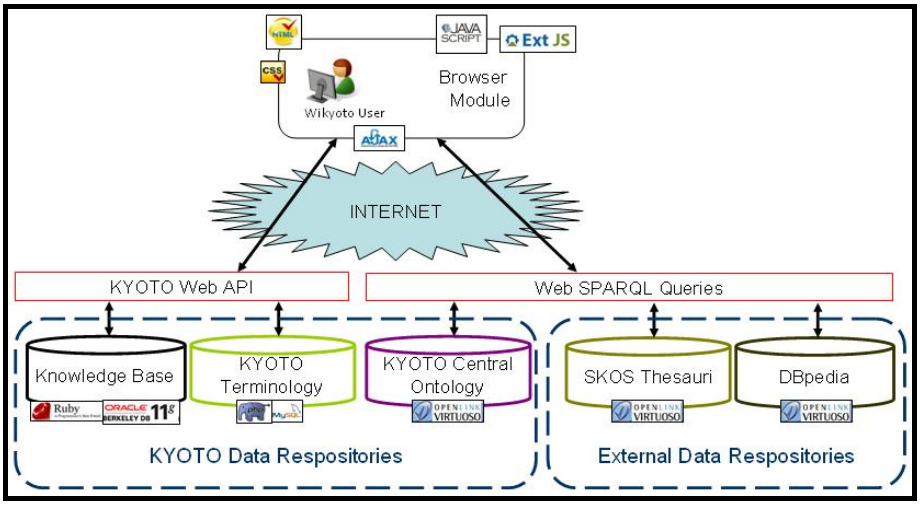


Figure 33: Implementation technologies and platforms in Wikyoto

The KYOTO Knowledge DR has been implemented as an extension of the DEBVisDic [78] server platform by the Department of Computer Science of the Masaryk University of Brno, Czech Republic. DBpedia can be queried on-line by its Public SPARQL Endpoint. All the other components of Wikyoto have been implemented or proper platforms have been set-up and customized in the context of the work described in this thesis.

3.2.4 Exploiting Wikyoto

A meaningful example of knowledge editing tasks that can be performed through Wikyoto in order to enrich an English Domain WordNet related to the environment is described. A set of video-tutorials that more extensively describe examples of usage of Wikyoto and the Wikyoto user guide are accessible at http://www.wikyoto.net/.

Let's suppose that an expert of environmental issues is refining the English Domain WordNet hierarchy of synsets related to 'pollution' by specifying relevant kinds of pollution (see Figure 34). On the left side of the interface of Wikyoto, the Domain WordNet hierarchy of synsets that are hyponyms of pollution is shown. The expert searches for the term 'pollution' in the KYOTO Terminology in order to find relevant suggestion to enrich the

Domain WordNet. He finds a list of many kinds of pollution that have been automatically mined from KYOTO parsed documents (left side of Figure 34 – i.e. 'oil pollution', 'marine pollution', etc.). He decides that the term 'nutrient pollution' belonging to the list of KYOTO terms is an important type of pollution not present in the Domain WordNet. Thus he drags the KYOTO term over the synset 'pollution' in order to create a new child synset, the synset 'nutrient pollution'.

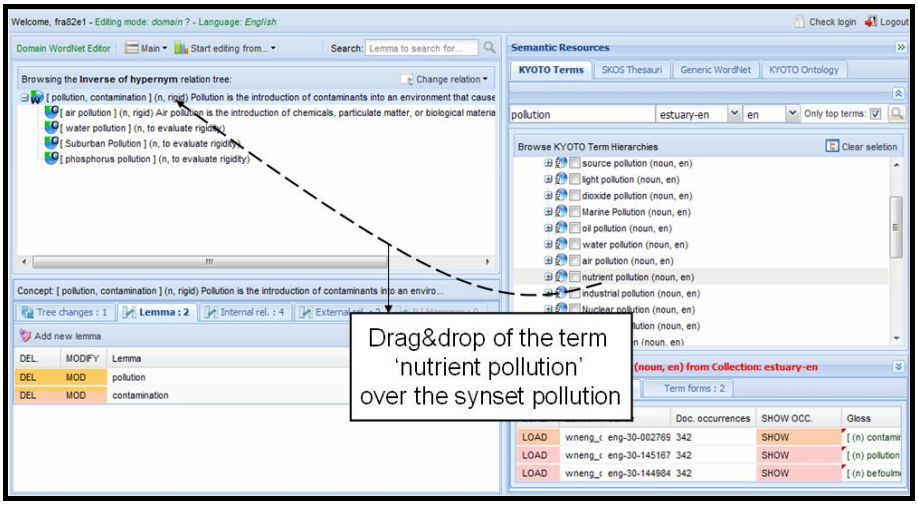


Figure 34: Creating a new synset from a KYOTO term

The features (lemma, part of speech, and gloss) of the synset 'nutrient pollution' need to be specified to finalize its creation. In order to do that, a pop-up windows opens (Figure 35 - A). The main lemma of the new synset, 'nutrient pollution' has been imported from the label of the KYOTO term from which the same synset has been created. The part of speech is set to noun. Concerning the gloss, it is possible to search in DBpedia for 'nutrient pollution' descriptions and import the most appropriate one, if present. Only one result is retrieved in DBpedia (Figure 35 - B). Since the description of 'nutrient pollution' from DBpedia correctly defines the meaning of the new synset, it is possible to import it in Wikyoto as the gloss of the 'nutrient pollution' synset (Figure 35 - C).

Once determined the lemma, part of speech, and gloss of the 'nutrient pollution' synset, its creation is finalized. The new 'nutrient pollution' synset is shown among the hyponyms of the 'pollution' synset in the Domain WordNet tree-view (see Figure 36). But the 'nutrient pollution' synset represents a particular kind of 'water pollution', thus the expert decides to drag this synset and drop it over the 'water pollution' synset (see Figure 36). In this way, the hyponym hierarchy of 'pollution' is better

structured because it specifies that 'nutrient pollution' is a hyponym of 'water pollution'.

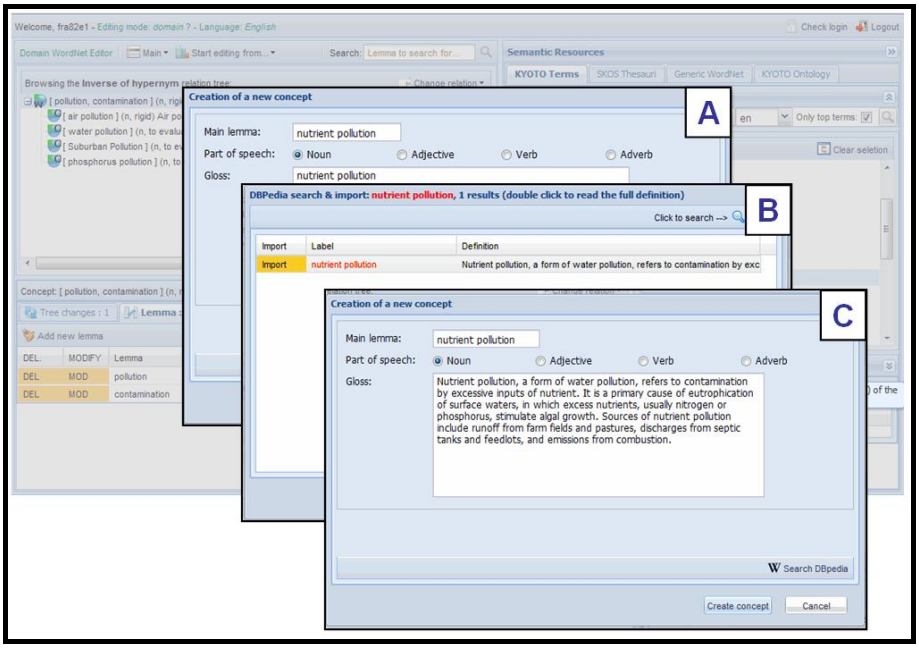


Figure 35: Importing a synset definition from DBpedia

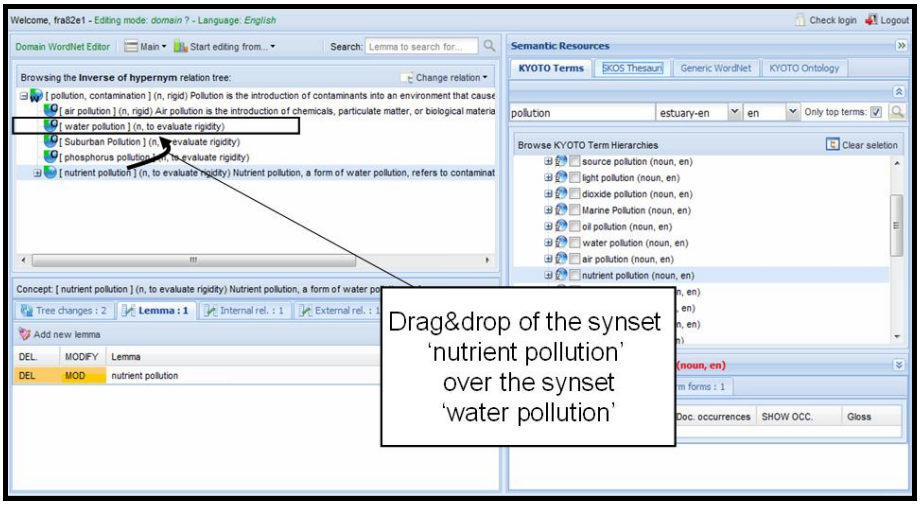


Figure 36: Visualization of a new synset in the tree-view box

3.2.5 TMEKO: supporting users to formalize cross-lingual information

This subsection introduces the Tutoring Methodology for Enriching the KYOTO Ontology (TMEKO) [82], an experimental component of Wikyoto designed to help users in the formalization of cross-lingual information by means of simplified interactions and natural language interviews.

The ontologization of WordNet synsets is the instantiation of proper mapping relations from the synsets to language-independent ontological entities constituted by the properties and classes of the KYOTO Central Ontology. The ontologization is the basic feature that enables cross-lingual text mining in KYOTO. To this purpose the creation of a rich and complete set of mappings from WordNet synsets of different languages to the Central Ontology is fundamental. The definition of these mappings is not an easy task. People without any knowledge engineering background can determine neither the right set of relations that characterize a synset nor the most appropriate ontological class or property representing the target of each one of the chosen relations.

In Wikyoto, users can exploit two editing patterns to map Domain WordNet synsets to the KYOTO Central Ontology. In particular, they can:

- browse the KYOTO Central Ontology in order to identify the right classes or properties to map a Domain WordNet synset to. Once identified the target of the mapping, users are asked to select the kind of mapping relation to instantiate from a list. This approach requires from users knowledge of both the structure of the KYOTO Central Ontology and the meaning and usage of the available mapping relations.
- execute the TMEKO procedure. TMEKO is a language-driven approach to ontologize synsets useful to gather contributions from users with no experience in knowledge formalization. To map a synset to the ontology in TMEKO users have to deal only with textual contents and to answer a set of questions. In this way, users feel more comfortable because they perform mainly natural language interactions, and exploit the language as they use it in everyday life.

In particular, in TMEKO, when a user wants to ontologize a synset, he is asked to select textual excerpts by defining the referred concept among a set of definitions retrieved by accessing Web search engines or by querying on-line encyclopaedic resources. Once selected, the user is invited to choose the most relevant words and expression that characterize the same

concept. On the basis of the results of the automated disambiguation of these words an interview is generated. The user is invited to answer a set of yes/no natural language questions. Each answer or each set of answers is useful to define if a mapping from the considered synset to a class or a property of the Central Ontology can be instantiated.

This subsection is divided in 3 parts. The first part describes the kinds of relations that can be exploited to map WordNet synsets to the KYOTO Central ontology and explains the TMEKO procedure. The second part describes the steps of the TMEKO procedure. Finally, the third part compares TMEKO with TMEO, an alternative approach to map linguistic knowledge over ontological entities by means of user interviews. At the time writing the TMEKO procedure is being refined and a Web application to execute a simplified version of TMEKO is under development and will be exploited to support its evaluation.

3.2.5.1 Mapping WordNet synsets to the KYOTO Central Ontology

In KYOTO a wide range of mapping relations has been defined in order to ontologize WordNet synsets by linking the same synsets to the KYOTO Central Ontology. On the basis of the structure of the KYOTO Central Ontology and on the features of the specific synset to ontologize, the set of possible mapping relations are described.

The classes of the KYOTO Central Ontology can be divided into three broad groups:

- *Endurants*, are entities that can be observed or perceived as a complete concept in any considered snapshot of time. Examples are material objects, such as a car or a human, but also an organisation or the border of a country.
- *Perdurants* (called also events) are entities for which only a part exists if we look at them at any given snapshot in time. Perdurants are often what we know as processes, for example 'walking'. If we freeze time, then we only see a part of the considered action.
- *Quality-regions* (called also states) do not exist on their own. They need another entity in which they resume. Examples of qualities and the values they assume are colours, or temperatures.

For a more detailed introduction to this distinction into three categories the interested reader is referred to the Wikipedia article describing Formal Ontologies[104].

The set of ontology mappings that can characterize a WordNet synset can be determined on the basis of the *rigidity* of the synset and the *kind of ontological class* (Endurant, Perdurant or Quality-region) the nearest hypernym of the same synset is mapped to. In Figure 37, the Domain WordNet synset 'golden frog' is considered. It is a rigid synset. In order to find the kind of ontological class the nearest hypernym of the synset 'golden frog' is mapped to, it is possible to consider the knowledge formalized in the Multilingual Knowledge Base. The Domain WordNet synset 'frog' is a hypernym of 'golden frog'. The 'frog' Domain WordNet synset is mapped to the corresponding 'frog' Generic WordNet synset as an equivalent concept (DGM_equivalence mapping relation). In the Generic WordNet, the 'frog' synset includes the 'animal' synset among its hypernyms. From the set of mappings from Generic WordNet synsets to the KYOTO Central Ontology, it is possible to notice that the 'animal' synset is mapped to the 'animal' ontological concept of the KYOTO Central Ontology (sc_equivalence mapping relation). The 'animal' ontological concept is an Endurant (since the 'animal' class is directly or indirectly subsumed by the 'endurant' class). As a consequence, the kind of ontological class of the nearest hypernym of the 'golden frog' synset is Endurant.

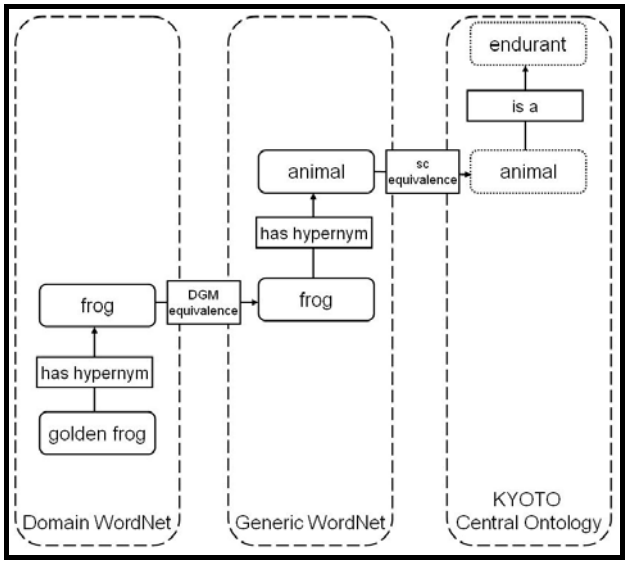


Figure 37: Definition of the nearest ontological class of a synset (TMEKO)

[104] Formal ontology, from Wikpedia - http://en.wikipedia.org/wiki/Formal_ontology

Starting from the evaluation of the rigidity and the kind of ontological class of the nearest hypernym, three groups of mapping relations from a WordNet synset to the KYOTO Central Ontology have been considered in the initial version of TMEKO so as to drive users in their instantiation: mappings from a rigid synsets with the nearest hypernym mapped to an Endurant class, mappings from a non-rigid synsets with the nearest hypernym mapped to an Endurant class, and mappings from a synset with the nearest hypernym mapped to a Perdurant class.

Mappings from a rigid synsets with the nearest hypernym mapped to an Endurant class

In this case, the synset to ontologize can be mapped to the Endurant class by means of a "sc_subclassOf" or a "sc_equivalenceOf" relation. Examples of this kind of mappings are:

- rigid synset: [species] → sc_equivalenceOf → ontological class: species (Endurant)
- rigid synset: [tree] → sc_equivalenceOf → ontological class: tree (Endurant)
- rigid synset: [local fish species] → sc_subClassOf → ontological class: fish (Endurant)
- rigid synset: [ecosystem condition] → sc_subClassOf → ontological class: environment_condition (Endurant)

Mappings from a non-rigid synsets with the nearest hypernym mapped to an Endurant class

The synset to ontologize can be mapped to the KYOTO Central Ontology by means of the following relations:

- "sc_domainOf" towards an ontological class representing the domain of the synset. The synset is not a proper subclass of the considered ontological class and is not disjoint from the other rigid synsets mapped to the same class;
- "sc_participantOf" towards an ontological class representing the process the concept represented by the synset takes part to;
- "sc_playRole" towards an ontological property describing the role of the synset in the process identified by means of the relation "sc_participantOf";
- "sc_hasState" towards an ontological class describing a quality of the synset.

Two examples of these mappings are:

- non-rigid synset: [contaminant]
  - sc_domainOf → ontological class: amount_of_matter (Endurant)
  - sc_participantOf → ontological class: pollution (Perdurant)
  - sc_playRole → ontological property: done-by
- non-rigid synset: [valuable species]
  - sc_domainOf → ontological class: species (Endurant)
  - sc_hasState → ontological class: valuable (Quality-region)

Mappings from a synset with the nearest hypernym mapped to a Perdurant class

The synset to ontologize can be mapped to the KYOTO Central Ontology by means of the relations:

- "sc_hasParticipant" towards an ontological class describing an entity that participates in the process identified by the synset;
- "sc_hasRole" towards an ontological class describing the role of the ontological class identified by means of the relation "sc_hasParticipant" in the context of the process described by the synset. The role is chosen in a proper set of ontological classes in DOLCE (agent, patient, change of situation, etc.).

Three examples of these mappings are:

- synset: [habitat restoration]
  - sc_hasParticipant → ontological class: habitat (Endurant)
  - sc_hasRole → FunctionalParticipation.owl#patient
- synset: [erosion protection]
  - sc_hasParticipant → ontological class: erosion (Perdurant)
  - sc_hasRole → Causality.owl#has_change_situation
- synset: [flood protection]
  - sc_hasParticipant → ontological class: flood (Perdurant)
  - sc_hasRole →Causality.owl#has_change_situation

3.2.5.2 The steps of the TMEKO procedure

The sequence of steps to drive users in the ontologization of a synset by means of the TMEKO procedure is described together with a real example.

The final aim of the TMEKO procedure is to drive non-expert users in instantiating proper mappings from a WordNet synset to the entities of the KYOTO Central Ontology. Although the TMEKO procedure still represents a knowledge-intensive task, it can be performed by users without any knowledge of the linguistic and ontological resources of KYOTO.

The TMEKO procedure consists of the following steps:

1) Synset selection:

The Domain WordNet synset to map to the KYOTO Central Ontology is selected.

EXAMPLE: the 'animal migration' Domain WordNet synset is chosen so as to be ontologized.

2) Check of the connection to the Generic WordNet:

The user is asked to verify if the mapping to the Generic WordNet of the chosen Domain WordNet synset or one of its hypernyms is correct. If no mappings are present, the user is redirected to Wikyoto so as to create a new mapping. The presence of a consistent mapping to a Generic WordNet synset is essential to identify the kind of ontological class of the nearest hypernym of the considered Domain WordNet synset so as to proceed with TMEKO.

EXAMPLE: the synset 'animal migration' is linked to the English Generic WordNet synset 'migration'. A proper mapping of the synset ‘animal migration’ to the synset ‘migration’ of the Generic WordNet is defined.

3) Definition of the kind of ontologization interview:

By accessing to the Multilingual Knowledge Base, it is possible to automatically determine if the nearest hypernym of the Domain WordNet synset to ontologize is linked to an Endurant, Perdurant or Quality-region ontological class. The initial version of the TMEKO procedure concerns the insanitation of mappings only for Endurant or a Perdurant classes.

If the nearest hypernym of the synset is an Endurant class, the rigidity of the synset to ontologize is considered. If the rigidity of the synset in not specified, the users is asked to evaluate it by means of a rigidity interview.

Once defined if the ontological class of the nearest hypernym of the synset to ontologize is an Endurant or a Perdurant and the rigidity value of the same synset, the kind of interview is determined. There three kinds of interviews:

- Rigid endurant interview
- Non-rigid endurant interview
- Perdurant interview

EXAMPLE: the ontological class of the nearest hypernym of the synset 'animal migration' is mapped to a Perdurant class, thus the interview to ontologize this synset is a Perdurant interview.

4) Collection of definitions of the synset:

The user is asked to select a set of definitions of the Domain WordNet synset to ontologize. A list of definitions retrieved by querying a Google custom search engine[105] is proposed in the TMEKO procedure. The search engine has been customized so as to search in encyclopaedic Web resources and to use specific query patterns ("X is a", etc.).

The user can select textual excerpts describing the WordNet synset to ontologize browsing the set of search results and editing or modifying the set of definition.

EXAMPLE: considering the Domain WordNet synset 'animal migration', many definitions are retrieved from Wikipedia, the Encyclopaedia Britannica, the Kids Encyclopaedia and so on by exploiting the Google custom search engine. The gloss of the synset 'animal migration' is automatically included in the list of definitions. The users can create a new definition with the first sentence of the ‘Animal migration’ Wikipedia article (constituting the first search result): "Animal migration is the travelling of long distances in search of a new habitat. The trigger for the migration may be local climate, local availability of food, or the season of the year". Once the list of 'animal migration' definitions has been completed the user can proceed to the next step.

5) Selection of relevant words from the definitions:

From the set of definitions of the Domain WordNet synset to ontologize, the user is asked to select the words that better describe the same synset by specifying the part of speech of each word selected.

[105] Google Custom Search Engine - http://www.google.com/cse/

EXAMPLE: considering the Domain WordNet synset 'animal migration', the user can select the following words from the definitions: 'migration' (noun), 'movement' (noun), ‘animal’ (noun).

6) Disambiguation of relevant words:

The KYOTO disambiguation service (called EHU-Disambiguation service) is invoked in order to retrieve from the set of relevant words describing the Domain WordNet synset to ontologize, the set of WordNet synsets that better describe the meanings of these words and thus the set of classes of the KYOTO Central Ontology associated to these synsets.

EXAMPLE: considering the relevant words 'migration' (noun) and 'movement' (noun) related to the Domain WordNet synset 'animal migration', the invocation of the KYOTO disambiguation service retrieves the following set of related classes from the KYOTO Central Ontology:

- Natural event

  (Kyoto#happening__occurrence__occurrent__natural_event-eng-3.0-07283608-n)

- Spend (Kyoto#spend__pass-eng-3.0-02708420-v)
- Motion (Kyoto#motion__movement__move__motility-eng-3.0-00331950-n)
- Animal (Kyoto#animal)
-

7) Ontologization user interviews:

To each class of the KYOTO ontology a set of user interview templates has been associated by experts of knowledge representation, on the basis of the kind of interview considered (Rigid endurant, non-rigid endurant, and perdurant). Each interview template can be exploited to create a real user interview considering a particular Domain WordNet synset to ontologize. Each user interview is made of a yes/no question. Each answer to an user interview determine if to instantiate or not a specific kind of mapping from the considered Domain WordNet synset to the class of the KYOTO Central Ontology the interview is associated to.

A set of user interviews is automatically generated on the basis of the interview templates associated to all the classes of the KYOTO Central Ontology that have been retrieved by the KYOTO disambiguation service. The user is asked to answer these interviews in order to instantiate or not

mapping from the Domain WordNet synset to ontologize to the related classes of the KYOTO Central Ontology.

EXAMPLE: considering the synset 'animal migration' and the Perdurant interview templates associated with the ontological classes Natural event, Spend, and Motion, 8 user interviews are generated. Each answer could generate a mapping from the synset 'animal migration' to the related ontological class.

An example of user interview is:

- Does 'animal' takes part to 'Animal migration'? (YES/NO).

If the answer is YES the following mapping relation is created:

synset: [animal migration] → sc_subClassOf → ontological class: animal (Endurant)

By accessing Wikyoto, knowledge engineers can collaboratively define the set of interview templates associated to each class of the KYOTO Central Ontology.

3.2.5.3 TMEKO and TMEO: language-driven vs. logic-driven approaches to enrich ontologies

A similar approach to ontologize linguistic knowledge by natural language interviews has been proposed in the context of the Senso Comune Project[106], aiming at developing a collaborative platform to build and maintain an open (hybrid) knowledge base of Italian language. In Senso Comune users are invited to map the lexicalizations of concepts over classes of the DOLCE top-level ontology by means of a series of natural language questions. This procedure is referred to as TMEO, the Tutoring Methodology for Enriching Ontology [83]. Questions are dynamically generated on the basis of the structure of the DOLCE ontology with the aim of defining the most appropriate ontological classes that describe a concept. A set of templates to generate questions has to be defined on the basis of the structure of the considered ontology, in this case, DOLCE. Conditional chains of questions have to be determined so that the sequence of question is dynamically chosen according to the answers provided by the user.

Both the TMEKO and the TMEO procedures require a preliminary involvement of knowledge engineers to define the set of question templates useful to generate user interviews. The TMEKO procedure,

[106] Senso Comune Web Site - http://www.sensocomune.org/

adopted in KYOTO, is language-driven since it starts from the user selection of natural language definitions to ontologize a concept. On the contrary, the TMEO procedure is logic-driven since the set of questions exploited to ontologize concepts are derived from the static structure of the DOLCE reference ontology.

In TMEKO, users are asked to choose among descriptive textual excerpts, a set of words and expressions that characterizes a concept. The interview to ontologize the concept is originated from the automated disambiguation of these words. Questions are useful to verify if it is possible to instantiate specific mappings to defined ontological classes. As a consequence, TMEKO interviews do not start from generic ontological distinctions like TMEO interviews, but are targeted to a specific set of questions exploited to define particular mappings. In TMEO, since a top-level reference ontology has been adopted, it is difficult to define precise ontological descriptions of concepts; interviews aim at instantiating basic ontological claims. Instead, thanks to the specialization of TMEKO interviews, rich sets of ontological mappings can be instantiated.

3.2.6 Evaluation

Wikyoto has been evaluated by involving a group of four experts of the environmental domain from the World Wide Fund for Nature[107] (WWF Netherlands) and the European Centre for Nature Conservation[108] (ECNC Netherlands). In particular, they contributed to the building of an English Domain WordNet about issues affecting estuaries. After a presentation of the main features of Wikyoto, domain experts were required to freely edit and refine the English Domain WordNet for a period of one month. A collection of documents describing environmental issues concerning estuaries was processed by KYOTO. Therefore a KYOTO terminology was automatically mined from these documents. This terminology was made accessible in Wikyoto so as to support domain experts by providing relevant examples of terms describing the considered domain.

This subsection presents the results of the analysis the resulting English Domain WordNet. Some considerations are exposed. The Domain WordNet is made of 1216 synsets and 1294 lemmas with an average of 1.064 lemmas per synset. All the created synsets are nouns. These synsets have been linked by 908 semantic relations including 907 hyponymy/hypernymy relations and 1 holonymy relation. From these data it is possible to draw that only nouns have been included in the Domain WordNet and thus

[107] WWF - http://www.wwf.org/
[108] ECNC - http://www.ecnc.org/

considered as relevant entities to describe the domain of interest. No synsets related to adjectives, verbs, and adverbs have been created. Moreover, users have provided only one lemma describing each synset in most of the cases, mainly because it is more difficult to specify many synonyms associated to a concept when a specific domain is considered. Almost all the semantic relations defined among synsets are hyponymy/hypernymy relations because the classification of concepts (synsets) on the basis of their specialization usually represents the most direct and easy understandable way to organize knowledge for common users. Other semantic relations have not been exploited since it is more difficult to reason about them.

Considering the hyponymy/hypernymy hierarchies of synsets, in the Domain WordNet 309 small hierarchies have been defined. Considering the synsets representing the roots of these hierarchies, 122 have been mapped to a Generic WordNet synset. In total 210 Domain WordNet synset have been mapped at least to a Generic WordNet synset. The organization of Domain WordNet synset in small hyponymy/hypernymy hierarchies can be explained considering that Domain WordNets are constituted mainly by collections of concepts describing a specific domain. It is normal that these concepts cannot be grouped into big specialization hierarchies since they often refer to entities that are relevant to the considered domain but differently related. More than one over three root synsets has been mapped to the Generic WordNet.

The mappings from English Domain WordNet synsets to the KYOTO Central Ontology have been created by knowledge engineers with the support of domain experts. 412 Domain WordNet synsets over a total number of 1216 (about 34%) have been mapped to at least one class or property of the KYOTO Central Ontology. 106 Domain WordNet synsets constituting roots of hyponymy/hypernymy hierarchies have been ontologized. This set of mappings from Domain WordNet synsets to the KYOTO Central Ontology will be adopted as a gold standard to evaluate the mappings of the same WordNet synsets that will be automatically created by domain experts by exploiting TMEKO. Further evaluation sessions are planned.

It is also relevant to consider that 420 Domain WordNet synsets over 1216 (about 35%) have been created starting from a KYOTO term. This fact highlights that the possibility to browse external knowledge resources so as to search for relevant suggestions to enrich Domain WordNets is fundamental for the users of Wikyoto.

The same group of users that built the English Domain WordNet also provided many relevant advices that have supported the improvement of

the interface and the user interaction patterns of Wikyoto. In particular, they suggested:

- the simplification of the layout of the interface;
- the enrichment of the set of items included in the contextual menus so as to provide users with more intuitive shortcuts to perform editing actions over single synsets visualized by tree-based views;
- the possibility to consult statistical data concerning the editing action performed globally or by a single user;
- the simplification of the interaction patterns to manage hyponymy/hypernymy hierarchies of synsets.

Throughout the development of the project, a continuous interaction has been kept with the group of domain experts and several cycles of refinement of the features of the interface and addition of new features have been done on the basis of their suggestions.

4. CONCLUSIONS AND PERSPECTIVES

In this thesis I have investigated methodologies, patterns, technologies, and tools to enable users with little or no experience in knowledge engineering to edit knowledge resources by easily modelling the distinguishing semantic features of natural language that are needed to support cross-lingual text mining. By adopting the wiki paradigm for collaborative content editing, new methodologies and simplified user interaction patterns useful to browse and modify knowledge resources have been designed and implemented throughout the development the Wikyoto Knowledge Editor, within the European Project KYOTO.

This research has compared the features of several knowledge editing tools, both semantic wikis and ontology editors, so as to identify extensively adopted knowledge editing patterns and approaches. Much emphasis has been put on the analysis of widespread user interactions as well as on the aspects related to the collaborative editing of knowledge structures.

The collaborative knowledge editing paradigm has been proven to be useful to manage and enrich the multilingual linguistic and ontological knowledge resources of KYOTO, a cross-lingual text mining system. Considering the linguistic features needed to support the mining of textual contents in KYOTO, a set of intuitive knowledge visualization patterns and language-driven user interactions has been identified in order to help non-expert users to easily contribute to the knowledge editing activity. Among them, we have decided to exploit visual representations of knowledge structures - like tree-based views - and simplified user interactions - like drag&drop actions to rearrange hierarchies of knowledge objects. Methodologies based on natural language interactions have been defined in order to simplify the most difficult knowledge editing actions.

The Wikyoto Knowledge Editor, the collaborative Web environment useful to edit the multilingual knowledge resources of KYOTO, has been designed and implemented in order to put in practice the identified knowledge editing patterns. In Wikyoto users, by means of their browser, are supported to easily navigate, enrich, and refine the knowledge resources of KYOTO by accessing a wide range of data sources, both internal and external to the KYOTO system. Wikyoto faces the main software design concerns of real-time collaborative editing environments and exploits natural-language interactions to easily involve users in complex knowledge editing tasks like the definition of the rigidity of WordNet synsets. A more experimental component of Wikyoto, TMEKO has also been designed and implemented. TMEKO is a methodology useful to support users to easily

enrich KYOTO knowledge resources with cross-lingual information by natural language interviews. Wikyoto has been evaluated by involving a group of domain experts so as to personalize the knowledge resources of KYOTO with respect to the environmental domain, by modelling an English Domain lexicon. Wikyoto has proven to be effective in the simplification of several knowledge editing tasks otherwise extremely difficult for users with no experience in knowledge representation. Further evaluation of the support of Wikyoto to formalize cross-lingual information has been planned.

Summarizing, by means of Wikyoto several patterns and methods to collaboratively edit knowledge structures useful to mine textual contents have been explored and implemented. Even if there is still room for many improvements to extensively exploit widespread users contributions in the editing of knowledge resources, Wikyoto represents a relevant attempt in this direction because it has proposed and experimented a wide range of user interaction possibilities, and it has identified some critical issues to be faced in future projects.

BIBLIOGRAPHY

1. Bergamaschi S., Guerra F., Leiba B., “Guest Editors' Introduction: Information Overload”, *IEEE Internet Computing*, vol. 14, no. 6, 2010, pp. 10-13
2. Bawden D., Robinson L., 2009. “The dark side of information: overload, anxiety and other paradoxes and pathologies”, *Journal of Information Science*, vol. 35, no. 2, 2009, pp. 180-191
3. Aidi X., “Cognitive Overload and Its Countermeasures from the Angle of Information Processing”, Proceedings of *Intelligent Information Technology Application (IITA 2009). Third International Symposium*, Nanchang (China) , 21-22 Nov. 2009, vol. 1, pp. 391 - 394
4. Baddeley A., “Working memory”, *Science* , vol. 255, no. 5044, 1992, pp. 556-559
5. Berners-Lee T., Hendler J., Lassila O., “The Semantic Web: A new form of Web content that is meaningful to computers will unleash a revolution of new possibilities”, *The Scientific American*, May 2001
6. Jackson P., Moulinier I., **Natural Language Processing for online applications (second revised edition)**, John Benjamins Publishing Company, 2007
7. Mitkov R., **The Oxford Handbook of Computational Linguistics**, Oxford University Press, 2003
8. Lassila O., “Semantic web, quo vadis?” Keynote speech of the *Ninth Scandinavian Conference on Artificial Intelligence (SCAI 2006)/ Finnish Artificial Intelligence Conference (STeP 2006)*, Espoo (Finland), 25-27 Oct. 2006
9. Booth D., “The Four uses of a URL: Name, Concept, Web location and Document instance”, *World Wide Web Consortium document*, Jan. 2003, http://www.w3.org/2002/11/dbooth-names/dbooth-names_clean.htm
10. Auer S., Bizer C., Kobilarov G., Lehmann J., Cyganiak R., Ives Z., “DBpedia: A Nucleus for a Web of Open Data”, Proceedings of the *6th International Semantic Web Conference, 2nd Asian Semantic Web Conference (ISWC 2007, ASWC 2007)*, Busan (Korea), 11-15 Nov. 2007, *Lecture Notes in Computer Science*, 2007, vol. 4825, pp. 722-735

11. Bizer C., Heath R., Berners-Lee T., "Linked Data - The Story So Far" *International Journal on Semantic Web and Information Systems (IJSWIS),* vol. 5, no. 3, 2009, pp. 1-22

12. Bizer C., Lehmann L., Kobilarov G., Auer S., Becker C., Cyganiak R., Hellmann S., "DBpedia – A Crystallization Point for the Web of Data", *Journal of Web Semantics: Science, Services and Agents on the World Wide Web*, vol. 7, 2009, pp. 154–165

13. Wilks Y., Brewster C., **Natural Language Processing as a foundation of the Semantic Web**, Foundation and Trends in Web Science, 2009

14. Lortal G., Chaignaud N., Kotowicz JP., Pécuchet JP., "NLP Contribution to the Semantic Web: Linking the Term to the Concept", Proceedings of the *13th International Conference on Knowledge-Based and Intelligent Information and Engineering Systems: Part I (KES 2009)*, Santiago (Chile), 28-30 Sep., *Lecture Notes in Computer Science,* vol. 5711, pp. 309-317

15. Butuc MG., "Semantically Enriching Content Using OpenCalais", Technical report, 2009

16. Mihalcea R., Csomai A., "Wikify! Linking Documents to Encyclopedic Knowledge", Proceedings of the *16th ACM conference on Information and Knowledge management (CIKM07)*, Lisbon (Portugal), 6-9 Nov. 2007, pp. 233-242

17. Hodge G., **Systems of Knowledge Organization for Digital Libraries: Beyond Traditional Authority Files**, Council on Library and Information Resources, 2000

18. Palavitsinis N., Manouselis N., "A Survey of Knowledge Organization Systems in Environmental Sciences", *Information Technologies in Environmental Engineering Environmental Science and Engineering*, Part 2, 2009, pp. 505-517

19. Ruppenhofer J., Ellsworth M., Petruck MRL., Johnson CR., Scheffczyk J., "FrameNet II: Extended Theory and Practice", *ICSI Tecnical report*, 2010

20. Fellbaum C., (Ed.) **Wordnet An Electronic Lexical Database**, MIT press, 1993.

21. Navigli R., Ponzetto SP., "BabelNet: Building a Very Large Multilingual Semantic Network" Proceedings of the *48th Annual*

Meeting of the Association for Computational Linguistics (ACL 2010), Uppsala (Sweden), 11-16 Jul. 2010

22. Llorens J., Morato J., Marzal MA., Moreiro J., "Wordnet applications", *Technical report*, Dept. Computer Science, Universidad Carlos III, Madrid, Spain - Dept. Library Science, Universidad Carlos III, Madrid, Spain, 2004.
23. Pease A., Fellbaum C., Vossen P., "Building the Global WordNet Grid", Proceedings of the *CIL-18 Workshop on Linguistic Studies of Ontology*, Seoul (South Korea), 21-26 Jul. 2008
24. Vossen P., "WordNet, EuroWordNet and Global WordNet", *Revue française de linguistique appliquée,* vol VII, 2002, pp. 27-38
25. Pease A., Niles I., Li J., "The Suggested Upper Merged Ontology: A Large Ontology for the Semantic Web and its Applications**"**, *Working Notes of the AAAI-2002 Workshop on Ontologies and the Semantic Web*, Edmonton (Canada), 28 Jul. - 1 Aug. 2002
26. Gruber T., "A translation approach to portable ontologies", *Knowledge Acquisition*, vol. 5, no. 2, 1993, pp. 199-220
27. Allemang D., Hendler J., **Semantic Web for the Working Ontologist**, Morgan Kaufmann, 2008
28. Masolo C., Borgo S., Gangemi A., Guarino N., Oltramari A., Schneider L. "The WonderWeb, Library of Foundational Ontologies", *WonderWeb deliverable 18*, 2003, http://wonderweb.semanticweb.org/
29. Soualmia LF., Golbreich C., Darmoni SJ., "Representing the mesh in owl: Towards a semi-automatic migration", Proceedings of the *KR 2004 Workshop on Formal Biomedical Knowledge Representation,* Whistler (Canada), 1 Jun. 2004
30. Pan JZ., **Description logics: reasoning support for the Semantic Web**, PhD thesis, University of Manchester - Faculty of Science and Engineering, 2004.
31. Cuenca Grau B., Kalyanpur A., Katz Y., Sirin E., Parsia B., "Pellet: A practical owl-dl reasoner", *Web Semantics: Science, Services and Agents on the World Wide Web*, vol. 5, no. 2, 2007, pp. 51-53
32. Campanini SE., Castagna P., Tazzoli R.**, "**Platypus Wiki: a Semantic Wiki Wiki Web", Proceedings of the *3rd International Semantic Web Conference (ISWC 2004),* Hiroshima (Japan), 7-11 Nov. 2004

33. Krötzsch M., Vrandecic D., Völkel M., "Semantic MediaWiki", *The Semantic Web - ISWC*, Proceedings of the *5th International Semantic Web Conference (ISWC06),* Athens (USA), 5-10 Nov. 2006, *Lecture Notes in Computer Science*, vol. 4273, 2006, pp. 935-942

34. Herzig DM., Ell B., "Semantic MediaWiki in Operation: Experiences with Building a Semantic Portal" *Proceedings of the 9th International Semantic Web Conference (ISWC10),* Shangai (China), 7-11 Nov. 2010, *Lecture Notes in Computer Science,* vol. Part II, 2006, pp. 114-128

35. Schaffert S., Westenthaler R., Gruber A., **"**IkeWiki: A UserFriendly Semantic Wiki", Demo Proceedings of the *3rd European Semantic Web Conference (ESWC 2006)*, Budva (Montenegro), 11-14 Jun. 2006

36. Schaffert S., "IkeWiki: A SemanticWiki for Collaborative Knowledge Management**",** Proceedings of the *15th IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE '06)*, Manchester (UK), 26-28 Jun. 2006, pp. 388-396

37. Kurz T., Schaffert S., Bürger T., Stroka S., Sint R., Radulescu M., Grunwald S., "KiWi - A Platform for building Semantic Social Media Applications", Proceedings of *the 9th International Semantic Web Conference (ISWC10),* Shangai (China), 7-11 Nov. 2010

38. Auer S., Dietzold S., Riechert T., **"**OntoWiki - A Tool for Social, Semantic Collaboration" Proceedings of the *5th International Semantic Web Conference (ISWC06),* Athens (USA), 5-10 Nov. 2006, *Lecture Notes in Computer Science,* vol. 4273, 2006, pp. 736-749

39. Tramp S., Frischmuth P., Heino N., "OntoWiki: a Semantic Data Wiki Enabling the Collaborative Creation and (Linked Data) Publication of RDF Knowledge Bases", Demo Proceedings of *Knowledge Engineering and Knowledge Management by the Masses (EKAW 2010)*, Lisbon (Portugal), 11-15 Oct. 2010

40. Tramp S., Heino N., Auer S., Frischmuth P., "RDFauthor: Employing RDFa for Collaborative Knowledge Engineering", Proceedings of *Knowledge Engineering and Knowledge Management by the Masses (EKAW 2010)*, Lisbon (Portugal), 11-15 Oct. 2010, *Lecture Notes in Artificial Intelligence*, vol. 6317, 2010, pp. 90-104

41. Tramp S., Frischmuth P., Ermilov T., Auer S., **"**Weaving a Social Data Web with Semantic Pingback", Proceedings of *Knowledge*

Engineering and Knowledge Management by the Masses (EKAW 2010), Lisbon (Portugal), 11-15 Oct. 2010, *Lecture Notes in Artificial Intelligence*, vol. 6317, 2010, pp. 135-149

42. Ermilov T., Heino N., Auer S., **"**OntoWiki Mobile – Knowledge Management in your Pocket" Proceedings of *the 7th Extended Semantic Web Conference (ESWC 2010),* Heraklion (Greece), 30 May - 03 Jun. 2010

43. Souzis A., **"**Bringing the "Wiki-Way" to the Semantic Web with Rhizome", Proceedings of the *1st Workshop on Semantic Wikis: From Wiki to semantics (SemWiki 2006)* at the *3rd European Semantic Web Conference (ESWC 2006)*, Budva (Montenegro), 11-14 Jun. 2006, pp. 222-229

44. Souzis A., "Building a Semantic Wiki", *IEEE Intelligent Systems*, vol. 20, 2005, no. 5, pp. 87-91

45. Buffa M., Gandon F., Ereteo G., Sander P., Faron C., "SweetWiki: A semantic wiki", *Web Semantics: Science, Services and Agents on the World Wide Web*, vol. 6, no.1, 2008, pp. 84-97

46. Landefeld R., Sack H., Schiller F., "Collaborative Web-Publishing with a Semantic Wiki", Proceedings of the *1st Conference on Social Semantic Web (CSSW)*, Leipzig (Germany), 26-28 Sep. 2007

47. Nakanishi T., Zettsu K., Kidawara Y., Kiyoki Y., **"**SAVVY Wiki: A Context-oriented Collaborative Knowledge Management System", Proceedings of the *5th International Symposium on Wikis and Open Collaboration* (*WikiSym 2009)*, Orlando (USA), 25-27 Oct. 2009

48. Tudorache T., Noy N., "Collaborative Protégé", Proceedings of the *Social and Collaborative Construction of Structured Knowledge Workshop* at the *16th International World Wide Web Conference 2007*, Alberta (Canada), 8-12 May 2007

49. Sebastian A., Tudorache T., Noy NF., Musen MA., "Customizable Workflow Support for Collaborative Ontology Development", Proceedings of the *4th International Workshop on Semantic Web Enabled Software Engineering* in the *7th International Semantic Web Conference 2008 (ISWC 2008)*, Karlsruhe (Germany), 26-27 Oct. 2008

50. Tudorache T., Vendetti J., Noy NF., "Web-Protégé: A Lightweight OWL Ontology Editor for the Web", Proceedings of the *OWLED 2008: OWL experiences and directions, first international workshop*

in the 7th International Semantic Web Conference (ISWC 2008), Karlsruhe (Germany), 26-27 Oct. 2008

51. Tudorache T., Falconer SM., Noy NF., Nyulas C., Üstün TB., Storey MAD., Musen MA., **"**Ontology Development for the Masses: Creating ICD-11 in WebProtégé", Proceedings of *Knowledge Engineering and Knowledge Management by the Masses (EKAW 2010)*, Lisbon (Portugal), 11-15 Oct. 2010, *Lecture Notes in Artificial Intelligence*, vol. 6317, 2010, pp.74-89
52. Weiten M., "OntoSTUDIO® as a Ontology Engineering Environment", *Semantic Knowledge Management***,** Springer, 2009, pp. 51-60
53. Bai F., El Jerroudi Z., "Interactive and Collaborative Ontology Development", Proceedings of *Mensch & Computing 2008 and DeLFI Cognitive Design 2008*, Berlin (Germany), 2008, pp. 174-179
54. Mainz I., Weller K., Paulsen I., Mainz D., Kohl J., von Haeseler A., "Ontoverse: Collaborative Ontology Engineering for the Life Sciences", Proceedings of the *German e-Science Conference 2007 (GES 2007),* 2-4 May 2007
55. Loskyll M., Heckmann D., Kobayashi I., "UbisEditor 3.0: Collaborative Ontology Development on the Web", Proceedings of *Web 3.0: Merging Semantic Web and Social Web,* Workshop at *Hypertext 2009*, Torino (Italy), 29 Jun.- 1 Jul. 2009
56. Loskyll M., Heckmann D., "Towards Collaborative Ontology Development in the Upcoming Web 3.0 Era with UbisEditor", Proceedings of the *11th International. Protégé Conference*, Amsterdam (Netherlands), 23-26 Jun. 2009
57. Kalyanpur A., Parsiaa B., Sirina E., Cuenca Grau B., Hendlera J., "Swoop: A Web Ontology Editing Browser", *Journal of Web Semantics*, vol. 4, no. 2, 2006, pp. 144-153
58. Kuhn T., "AceWiki: Collaborative Ontology Management in Controlled Natural Language", Proceedings of the *3rd Semantic Wiki Workshop* in the *5th European Semantic Web Conference (ESWC),* Tenerife (Spain), 01-05 Jun. 2008, *CEUR Workshop Proceedings,* 2008
59. Wyner A., van Engers T., "Towards Web-based Mass Argumentation in Natural Language", Proceedings of *Knowledge Engineering and Knowledge Management by the Masses (EKAW 2010)*, Lisbon (Portugal), 11-15 Oct. 2010

60. Kuhn T., "Attempto Controlled English as Ontology Language", Proceedings of the *3rd REWERSE annual meeting*, Munich (Germany), 21-24 March 2006

61. Funk A., Tablan V., Bontcheva K., Cunningham H., Davis B., Handschuh S., "CLOnE: Controlled Language for Ontology Editing", Proceedings of the *6th International Semantic Web Conference, 2nd Asian Semantic Web Conference (ISWC 2007, ASWC 2007)*, Busan (Korea), 11-15 Nov. 2007, *Lecture Notes in Computer Science,* vol. 4825, 2007, pp. 142-155

62. Davis B., *Iqbal* AA., Funk A., Tablan V., Bontcheva K., Cunningham H., Handschuh S., "RoundTrip Ontology Authoring", Proceedings of the *7th International Semantic Web Conference (ISWC 2008)*, Karlsruhe (Germany), 26-30 Oct. 2008, *Lecture Notes in Computer Science,* vol. 5318, 2008, pp. 50-56

63. Bernstein A., Kaufmann E., "GINO - A Guided Input Natural Language Ontology Editor", Proceedings of the *5th International Semantic Web Conference (ISWC 2006),* Athens (USA), 5-10 Nov. 2006, *Lecture Notes in Computer Science*, vol. 4273, 2006, pp. 144-157

64. Agirre E., Edmonds P., (Eds.) **Word Sense Disambiguation - Algorithms and Applications**, Springer, 2006

65. Navigli R., "Word sense disambiguation: A survey" *Journal ACM Computing Surveys (CSUR),* vol. 41 no. 2, 2009, pp. 1-61

66. Martinez Iraolak D., **Supervised Word Sense Disambiguation: Facing Current Challenges**, PhD thesis, Universidad del País Vasco - Departamento de Lenguajes y Sistemas Informáticos, 2004

67. Agirre E., Lopez de Lacalle O., Soroa A., "Knowledge-Based WSD on Specific Domains: Performing Better than Generic Supervised WSD", Proceedings of the *21st International Joint Conference on Artificial Intelligence (IJCAI 09)*, Pasadena (USA), 11-17 Jul. 2009

68. Navigli R., Ponzetto SP., "Knowledge-richWord Sense Disambiguation Rivaling Supervised Systems", *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics (ACL 2010),* Uppsala (Sweden), 11-16 Jul. 2010

69. Gangemi A., Guarino N., Masolo C., Oltramari A., "Sweetening wordnet with dolce", *AI Mag.,* vol. 24, no.3, 2003, pp. 13-24

70. Gangemi A., Navigli R., Velardi P., "The OntoWordNet Project: Extension and Axiomatization of Conceptual Relations in WordNet", Proceedings of *On the Move to Meaningful Internet Systems (OTM 2003)*, Catania (Italy), 3-7 Nov. 2003, pp.820-838

71. Hicks A., Herold A., "Evaluating ontologies with rudify", Proceedings of the *2nd International Conference on Knowledge Engineering and Ontology Development (KEOD 09),* Madeira (Portugal), 6-8 Oct. 2009, pp. 5-12

72. Agirre E., Soroa A., "Personalizing PageRank for Word Sense Disambiguation", Proceedings of the *12th Conference of the European Chapter of the Association for Computational Linguistics (EACL 09)*, Athens (Greece), 30 Mar. - 03 Apr. 2009

73. Francopoulo G., Monachini M., Declerck T., Romary L., "The relevance of standards for research infrastructure", Proceedings of the *6th Workshop Towards Research Infrastructures for Language Resources (LREC 2006)*, European Language Resources Association (ELRA), Genoa (Italy), 22-28 May 2006

74. Soria C., Monachini M., Vossen P., "Wordnet-lmf: fleshing out a standardized format for wordnet interoperability" Proceedings of the *International Workshop on Intercultural Collaboration (IWIC 09),* Palo Alto (USA), 20-21 Feb. 2009, pp. 139 - 146

75. Vossen P., Segers R., Hicks A., Herold A., Rigau G., Agirre E., Estarrona A., Cuadros M., Laparra E., Kanzaki K., "Wordnets mapped to central ontology – revised", Deliverable 6.6 – KYOTO Project

76. Bosma W., Vossen P., Soroa A., Rigau G., Tesconi M., Marchetti A., Monachini M., Aliprandi C., 2009 "KAF: a generic semantic annotation format", Proceedings of the *5th International Conference on Generative Lexicon (GL 2009)*, Pisa (Italy), 17-18 Sep. 2009

77. Agirre E. and Martinez, D. (2001) "Knowledge sources for word sense disambiguation", Proceedings of the *4th International Conference Text, Speech and Dialogue (TSD 2001)*, Zelezna Ruda (Czech Republich), 11-13 Sep. 2001, *Lecture Notes in Computer Science*, vol. 2166, 2001, pp. 1-10

78. Horak A., Karel P., Rambousek A., "The Global WordNet Grid Software Design", Proceedings of the *4th Global WordNet Conference*, Szeged (Hungary), 22-25 Jan. 2008, pp. 194-199

79. Liu J., Shao Y., Teng G., Yao W., Dong S., “Concurrency Control Strategy in Real-time Collaborative Editing Systems”, Proceedings of the 2nd International Conference on Education Technology and Computer (ICETC 2010), Shangai (China), 22-24 Jun. 2010

80. Sun C., Ellis C., “Operational Transformation in Real-time Group Editors: Issues, Algorithms, and Achievements”, Proceedings of the *1998 ACM conference on Computer supported cooperative work*, Seattle (USA), 14-18 Nov. 1998

81. Wang D., Mah A., Lassen S., “Google Wave Operational Transformation”, Whitepaper, July 2010, http://www.waveprotocol.org/whitepapers/operational-transform

82. Segers R., Vossen P., “Facilitating on-expert Users of the KYOTO Platform: the TMEKO Editing Protocol for Synset to Ontology Mappings”, Proceedings of the *7th Conference on International Language Resources and Evaluation (LREC 2010)*, Valletta (Malta), 17-23 May 2010

83. Oltramari A., Vetere G., “Lexicon and Ontology Interplay in Senso Comune”, Proceedings of *OntoLex 2008*, Workshop at the *6th. International Conference on Language Resources and Evaluation (LREC 2008)*, Marrakech (Morocco), 28-30 May 2008

APPENDIX: LIST OF PUBLICATIONS

Hereby I provide a list of publications related to this thesis:

Book chapters and journal articles

Ronzano, F., Marchetti A., *"From words to concepts: enabling shared semantic descriptions of Web contents by Tagpedia".* In the Book "**Introduction to Web Semantics: Concepts, Technologies and Applications**". ISBN: 9780980733013. Editor: Gabriel Fung. iConcept Press, Australia, October 2010

Vossen P., Agirre E., Bond F., Bosma W., Fellbaum C., Hicks A., Hsieh S., Isahara H., Huang Ch., Kanzaki K., Marchetti A., Rigau G., Ronzano F., Segers R., Tesconi M., *"KYOTO: a wiki for establishing semantic interoperability for knowledge sharing across languages and cultures".* In the "**Handbook of Research on Culturally-Aware Information Technology: Perspectives and Models**" ISBN: 9781615208838. Editors: Dr. E. Blanchard (Mc Gill University, Canada) and Dr. D. Allard (Dalhousie University), IGI Global USA - August, 2010

Ronzano F., Monachini M., Marchetti A., Tesconi M., Calzolari N., *"Bootstrapping and collaboratively enriching the Italian Domain WordNet through the Wikyoto Knowledge Editor"*. In a forthcoming "**Volume of the Romanian Academy Publishing House**". Editors: Dan Tufis (Romanian Academym, Research Institute for Artificial Intelligence) and Corina Forascu (University of Iasi, Faculty of Computer Science)

Conferences and Workshops

Aliprandi C., Ronzano F., Marchetti A., Tesconi M., Minutoli S., "*Extracting events from Wikipedia as RDF triples linked to widespread Semantic Web Datasets*" Accepted at the 14th International Conference on Human-Computer Interaction, July 2011

Ronzano F., Tesconi M., Minutoli S., Marchetti A., *"Collaborative management of KYOTO Multilingual Knowledge Base: the Wikyoto Knowledge Editor".* In the Proceedings of the 5th International Conference of the Global WordNet Association (GWC-2010), February 2010

Tesconi M., Ronzano F., Minutoli S., Aliprandi C., Marchetti A. *KAFnotator: a multilingual semantic text annotation tool"* In the Proceedings of the Fifth Joint ISO-ACL/SIGSEM Workshop on Interoperable Semantic Annotation, January 2010

Marchetti A., Minutoli S., Ronzano F., Tesconi M., "*A Wiki framework for cooperatively building multilingual resources".* In the Proceedings of the Second International Conference on Global Interoperability for Language Resources, January 2010

Marchetti A., Minutoli S., Ronzano F., Tesconi M., "*Wikyoto Knowledge Editor: the collaborative Web environment to manage KYOTO multilingual Knowledge Base".* In the Proceedings of the International Conference on Knowledge Management 2009: The 6th International Conference on Knowledge Management, December 2009

Bertagna F., Monachini M., Soria C., Calzolari N., Ronzano F., Tesconi M., Marchetti A., "*Cooperative Building of Semantic Resources".* In the Proceedings of the 10th Congress of Italian Association for Artificial Intelligence - Cooperative construction of linguistic knowledge bases Workshop, June 2007

Marchetti A., Tesconi M., Ronzano F., Rosella M., Bertagna F., Monachini M., Soria C., Calzolari N., Huang CR., Hsieh SK., "*Toward an Architecture for the Global Wordnet Initiative".* In the Proceedings of the SWAP 2006 Semantic Web Applications and Perspectives - 3rd Italian Semantic Web Workshop, December 2006

European project deliverables

Marchetti A., Ronzano F., Tesconi M., Hicks A., "*Wiki Environment for Ontology Editing"* - Deliverable 7.5a, Workpackage: 'Wiki for WordNets and Ontology', European Project KYOTO, February 2009 (Version 1) and February 2010 (revised version)

Marchetti A., Ronzano F., Tesconi M., Horak A., Rambousek A., "*Wiki Environment for WordNet"* - Deliverable 7.4a, Workpackage: 'Wiki for WordNets and Ontology', European Project KYOTO, February 2009 (Version 1) and February 2010 (revised version)

.

Marchetti A., Horak A., Ronzano F., Tesconi M., Rambousek A., "*Multilingual WordNet Services"* - Deliverable 7.3a, Workpackage: 'Wiki for WordNets and Ontology', European Project KYOTO, January 2009 (Version 1) and January 2010 (revised version)

Marchetti A., Ronzano F., Tesconi M., "*Knowledge Base Server API"* - Deliverable 7.2a, Workpackage: 'Wiki for WordNets and Ontology', European Project KYOTO, November 2008 (Version 1) and November 2009 (revised version)

Marchetti A., Ronzano F., Tesconi M., Soria C., Monachini M., Vossen P., Bosma W., "XML Schema for Wordnet and Ontology" - Deliverable 7.1, Workpackage: 'Wiki for WordNets and Ontology', European Project KYOTO, June 2008

Other works

Marchetti A., Ronzano F., Tesconi M., Minutoli S., "*Formalizing Knowledge by Ontologies: OWL and KIF".* CNR-IIT Technical Report IIT 2008-TR-007, July 2008

Tesconi M., Ronzano F., Minutoli S., Marchetti A., Rosella M., "*Semantic Web gets into collaborative tagging*". CNR-IIT Technical Report 2007-TR-006, May 2008