# An open and scalable framework for enriching ontologies with natural language content

Maria Teresa Pazienza and Armando Stellato

AI Research Group, Dept. of Computer Science, Systems and Production
University of Rome, Tor Vergata
Via del Politecnico 1, 00133 Rome, Italy
{pazienza,stellato}@info.uniroma2.it

**Abstract.** Knowledge Sharing is a crucial issue in the Semantic Web: SW services expose and share knowledge content which arise from distinct languages, locales, and personal perspectives; a great effort has been spent in these years, in the form of Knowledge Representation standards and communication protocols, with the objective of acquiring semantic consensus across distributed applications. However, neither ontology mapping algorithm nor knowledge mediator agent can easily find a way through ontologies as they are organized nowadays: concepts expressed by hardly recognizable labels, lexical ambiguity represented by phenomena like synonymy and polysemy and use of different natural languages which derive from different cultures, all together push for expressing ontological content in a linguistically motivated fashion. This paper presents our approach in establishing a framework for semi-automatic linguistic enrichment of ontologies, which led to the development of Ontoling, a plug-in for the popular ontology development tool Protégé. We describe here its features and design aspects which characterize its current release.

## 1 Introduction

The scenario offered by the SW (and by the Web in general) is characterized by huge quantities of documents and by users willing to access them. Though machine readability is a primary aim for allowing automatic exchange of data, several SW services like Intelligent Q&A, Semantic Search Engines etc.. still need to recognize and expose knowledge expressed in the sole way humans can easily understand it: natural language. Moreover, the role of different cultures and languages is fundamental in a real World *aWare* Web and, though English is recognized de facto as a "lingua franca" all over the world, much effort must be spent to preserve other idioms expressing different cultures. As a consequence, multilinguality has been cited as one of the six challenges for the Semantic Web [1]. These premises suggest that ontologies as we know them now, should be enriched to cover formally expressed conceptual knowledge as well as to expose its content in a linguistically motivated fashion.

In this paper we introduce our work in establishing a framework for semi-automatic linguistic enrichment of ontologies, which has run through the identification of different categories of linguistic resources and planning their exploitation to augment the linguistic expressivity of ontologies. This effort has lead to the develop-

ment of Ontoling[1], a plugin for the popular ontology editing tool Protégé [6]. We describe here the features characterizing its current release and discuss some of the innovations we are planning for the near future.

## 2 Linguistic Enrichment of Ontologies: motivation and desiderata

Whether considering the billions of documents which are currently available on the web, or the millions of users which access to their content, enriching conceptual knowledge with natural language expressivity seems to us a necessary step for realizing true knowledge integration and shareability.

To achieve such an objective, we should reconsider the process of Ontology Development to include the enrichment of semantic content with proper lexical expressions in natural language. Ontology Development tools should reflect this need, supporting users with dedicated interfaces for browsing linguistic resources: these are to be integrated with classic views over knowledge data such as class trees, slot and instance lists, offering a set of functionalities for linguistically enriching concepts and, possibly, for building new ontological knowledge starting from linguistic one.

By considering some of our past experiences [8, 9] with knowledge based applications dealing with concepts and their lexicalizations, a few basic functionalities for browsing linguistic resources (from now on, LRs) emerged to be mandatory:

- *Search term definitions (glosses)*
- *Ask for synonyms*
- *Separate different sense of the same term*
- *Explore genus and differentia*
- *Explore resource-specific semantic relations*

as well as some others for ontology editing:

- *Add synonyms (or translations, for bilingual resources) as additional labels for identifying concepts*
- *Add glosses to concepts description (documentation)*
- *Use notions from linguistic resources to create new concepts*

While ontologies have undergone a process of standardization which culminated, in 2004, with the promotion of OWL [4] as the official ontology language for the semantic web, linguistic resources still maintain heterogeneous formats and follow different models, which make tricky the development of such an interface. In the next two sections we address this problem and propose our solution for integrating available LRs.

## 3 Lexical resources, an overview

"The term linguistic resources refers to (usually large) sets of language data and descriptions in machine readable form, to be used in building, improving, or evaluating

[1] OntoLing is freely available for download at:http://ai-nlp.info.uniroma2.it/software/OntoLing

natural language (NL) and speech algorithms or systems" [3]. In particular, this definition includes lexical databases, bilingual dictionaries and terminologies, all resources which may reveal to be necessary in the context of a more linguistic-aware approach to KR. In past years several linguistic resources were developed and made accessible (a few for free), then a wide range of resources is now available, ranging from simple word lists to complex MRDs and thesauruses. These resources largely differentiate upon the explicit linguistic information they expose, which may vary in format, content granularity and motivation (linguistic theories, task or system-oriented scope etc…). Multiple efforts have been spent in the past towards the achievement of a consensus among different theoretical perspectives and systems design approaches.

The Text Encoding Initiative [14] and the LRE-EAGLES (Expert Advisory Group on Linguistic Engineering Standards) project [2] are just a few, bearing the objective of making possible the reuse of existing linguistic resources, promoting the development of new linguistic resources for those languages and domains where they are still not available, and creating cooperative infrastructure to collect, maintain, and disseminate linguistic resources on behalf of the research and development community.

However, at present time, with lack of a standard on existing LRs, it appears evident that desiderata for functionalities which we described in section 2, would depend upon the way these resources had been organized. Often, even a local agreement on the model adopted to describe a given (a series of) resource does not prevent from an incorrect formulation of its content. This is due to the fact that many resources have been initially conceived for humans and not for machines. In some cases [12] synonyms are clustered upon the senses which are related to the particular term being examined, in others [13] they are simply reported as flat lists of terms. In several dictionaries, synonyms are mixed with extended definitions (glosses) in a unpredictable way and it is not possible to automatically distinguish them. Terms reported as synonyms may sometimes not be truly synonyms of the selected term, but may represent more specific or general concepts (this is the case of Microsoft Word synonymy prompter). Of course, the ones mentioned above represent mere dictionaries not adhering to any particular linguistic model, though they may represent valuable resources on their own. A much stronger model is offered by Wordnet [5], which, being a structured lexical database, presents a neat distinction between words, senses and glosses, and is characterized by diverse semantic relations like hypernymy/hyponymy, antonymy etc… Though not being originally realized for computational uses, WordNet has become a valuable resource in the human language technology and artificial intelligence. Furthermore, the development of WordNets in several other languages [11] has definitively contributed to the diffusion of WordNet schema as a wide accepted model for LRs.

## 4 Accessing Linguistic Resources: The Linguistic Watermark

To cope with all of these heterogeneous LRs, we introduced the notion of Linguistic Watermark, as the series of characteristics and functionalities which distinguish a particular resource inside our framework. As we can observe from the Class Diagram in Fig. 1, we sketched a sort of classification of *linguistic resources*, with the addition of

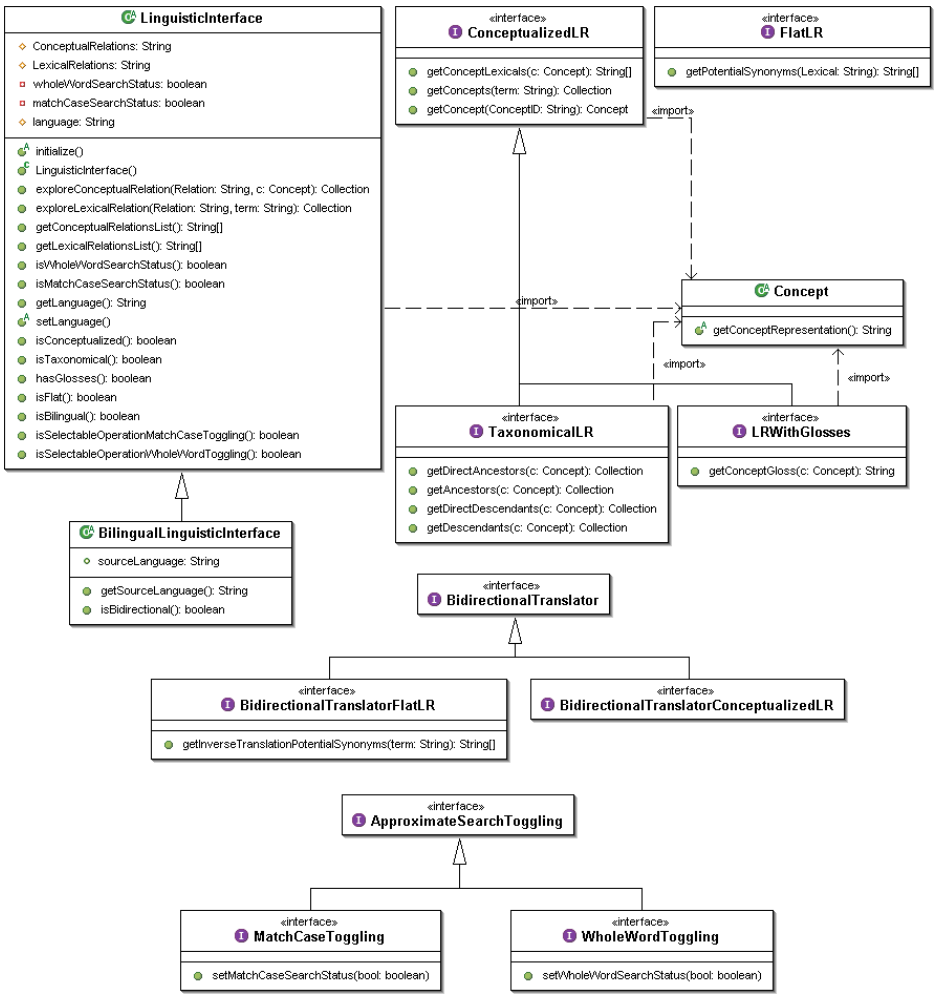


**Fig. 1.** The Linguistic Watermark

operational aspects. LRs are in fact structured and described in terms of their features and how their lexical information is organized; the diagram has then been completed with query methods for accessing resource's content.

We thus implemented this schema as a java package on its own, which can externally be imported by any application willing to exploit natural language resources like lexicons and terminologies. The core of the package is composed of an Abstract Class, named *LinguisticInterface*, which is both the locus for a formal description of a given linguistic resource and a service-provider for exposing the resource specific methods. The other abstract classes and interfaces in the package, which can be implemented or not, depending on the profile of the resource being wrapped, provide instead the signatures for known interface methods.

We have currently developed several implementations of the Linguistic Watermark. Two of them, the Wordnet Interface and the latest DICT Interface, being freely distributable, have been made publicly available on the Ontoling site.

The first one is an almost totally complete implementation of the Linguistic Watermark. The Wordnet Interface is in fact a *ConceptualizedLR,* because its linguistic expressions are clustered upon the different senses related to the each term. These senses – "synsets", in Wordnet terminology – have been implemented through the *Concept* interface, which we see bounded by the import statement in the class diagram. Wordnet is a *LRWithGlosses*, as glosses are neatly separated from synonyms and organized in a one-to-one relation with synsets. Finally, Wordnet Interface implements *TaxonomicalLR*, as its indexed word senses are organized in a taxonomy of more specific/more generic objects.

The other one, DICT Interface, is based on the Dictionary Server Protocol (DICT) [12], a TCP transaction based query/response protocol that allows a client to access dictionary definitions from a set of natural language dictionary databases. The DICT interface is *conceptualized* too, though its word senses are not indexed as in Wordnet (that is, it is not possible to correlate senses of two different terms upon the same meaning). DICT Interface is also a *BilingualLinguisticInterface*, as its available wordlists provide translations for several idioms.

Other available interface classes denote *Flat* resources (as opposed to *Conceptualized* ones), which contain flat lists of linguistic expressions for each defined term, and *BidirectionalTranslators*, which represent a further specialization of Bilingual Linguistic Interfaces providing bidirectional translation services.

We defined two classes of methods for browsing LRs: those defined in advance in the interfaces, which can thus be exploited inside automatic processes, and other very specific resource-dependent methods, which are loaded at run-time when the LR is interfaced to some browsing application (e.g. Ontoling). Two methods available in *LinguisticInterface*: *getLexicalRelationList* and *getConceptualRelationList* act thus as service publishers, the former providing different methods for exploring lexical relations among terms or relating terms to concepts, the latter reporting semantic relations among concepts.

## 5 Ontoling Architecture

The architecture of the Ontoling plugin (see Fig. 2) is based on three main components:

1. the GUI, characterized by the Linguistic Resource browser and the Ontology Enrichment panel
2. the external library *Linguistic Watermark*, which has been presented in the previous section, providing a model for describing linguistic resources
3. the core system

and an additional external component for accessing a given linguistic resource.

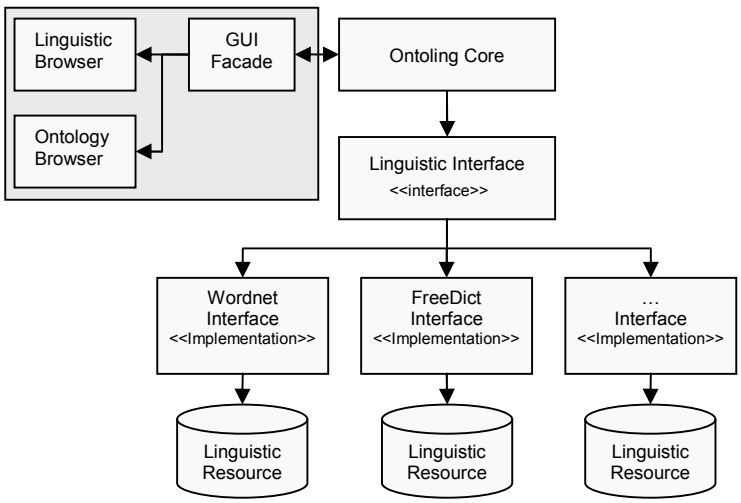


**Fig. 2.** Ontoling Architecture

This component, which can be loaded at runtime, must implement the classes and interfaces contained in the Linguistic Watermark library, according to the characteristics of the resource which is to be plugged. In the following sections we provide details on the above components.

### 5.1 Ontoling Core Application

The core component of the architecture is responsible for interpreting the Watermark of linguistic resources and for exposing those functionalities which suit to their profile. Moreover, the behavior of the whole application is dependant on the nature of the loaded resource and is thus defined at run-time. Several methods for querying LRs and for exposing results have been encapsulated into objects inside a dedicated library of behaviors: when a given LR is loaded, the core module parses its Linguistic Watermark and assigns specific method-objects to each GUI event.

With such an approach, the user is provided with a uniform view over diverse and heterogeneous linguistic resources, as they are described in the Linguistic Watermark ontology, and easily learns how to interact with them (thus familiarizing with their peculiarities) by following a policy which is managed by the system.

For example, with a *flat* resource, a search on a given term will immediately result in a list of (potential) synonyms inside a dedicated box in the GUI; instead, with a *conceptualized* resource, a list of word senses will appear in a results table at first, then it will be browsed to access synonymical expressions related to the selected sense. Analogous adaptive approaches have been followed for many other aspects of the Linguistic Watermark (mono or bidirectional Bilingual Translators, presence of glosses, Taxonomical structures and so on…) sometimes exploding with combinatorial growth.

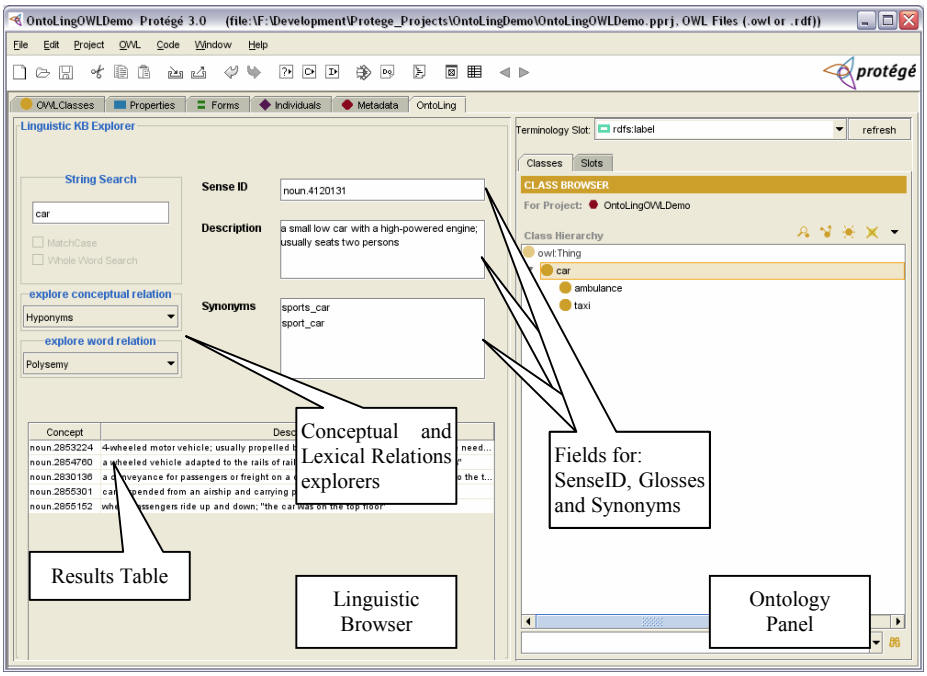


**Fig. 3** A screenshot of the Ontoling Plugin

### 5.2 Ontoling User Interface

Once activated, the plugin displays two main panels, the Linguistic Browser on the left side, and the Ontology Panel on the right side (see Fig. 3).

The Linguistic Browser is responsible for letting the user explore the loaded linguistic resource. Fields and tables for searching the LR and for viewing the results, according to the modalities decided by the core component, are made available. The menu boxes on the left of the Linguistic Browser are filled at run time with the methods for exploring LR specific Lexical and Conceptual relations.

The Ontology Panel, on the right, offers a perspective over ontological data in the classic Protégé style. By right-clicking on a frame (class, slot or instance), the typical editing menu appears, with some further options provided by Ontoling to:

1. search the LR by using the frame name as a key
2. change then name of the selected frame to a term selected from the Linguistic Browser
3. add terms selected from the Linguistic Browser as additional labels for the selected frame
4. add glosses as a description for the selected frame
5. add IDs of senses selected from the linguistic browser as additional labels for the frames
6. create a new frame with a term selected from the Linguistic Browser as frame name (identifier)

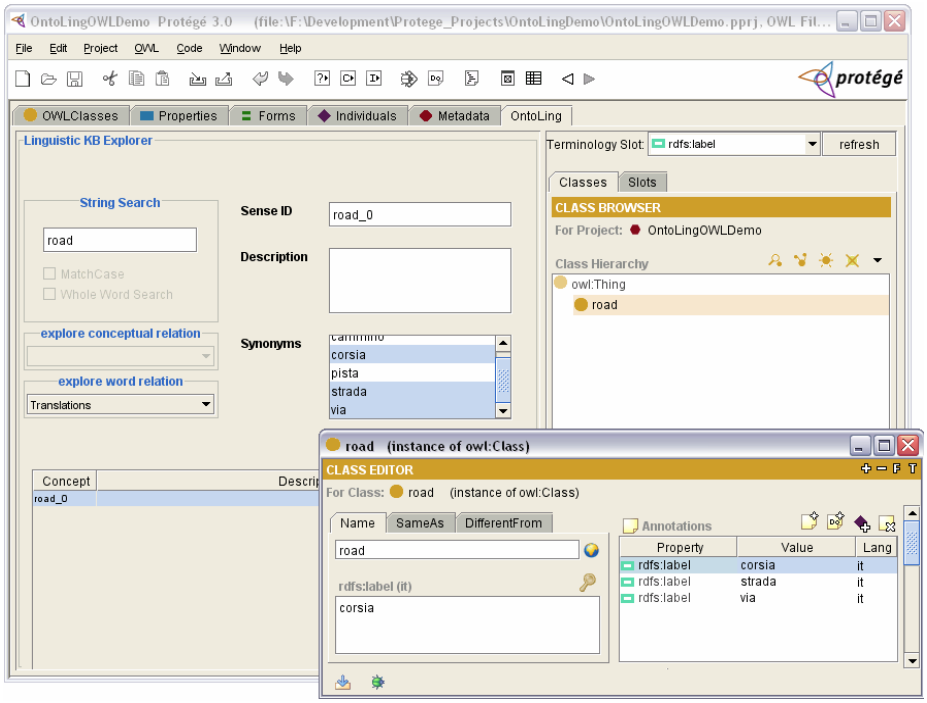


**Fig. 4** Enriching an English OWL concept with selected Italian terms

7. only in class and slot browser: if the LR is a *TaxonomicalLR*, explore hyponyms (up to a chosen level) of the concept selected on the Linguistic Browser and reproduce the tree on the frame browser, starting from the selected frame, if available

These functionalities allow not only for linguistic enrichment of ontologies, but can be helpful for Ontologists and Knowledge Engineers in creating new ontologies or in improving/modifying existing ones.

Note how functionality 5 has not a rigid linguistic motivation, but is indeed dedicated to those willing to build an artificial controlled vocabulary which contains direct references to the senses of a particular resource.

How terms and glosses are added to the description of ontologies concepts, depends on the ontology model which is being adopted and is explained in detail in the following section.

## 6 Using Ontoling with Protégé and Protégé OWL

When a frame-based approach was first adopted in Protégé as a knowledge model for representing ontologies and knowledge bases, no explicit effort was dedicated to the representation of possible alternate labels (synonyms) for concepts neither to support the idea of multilingualism in Ontologies. Frame names were almost as equivalent as IDs, and people were only encouraged, as it is common practice in computer pro-

gramming when addressing variable names, to adopt "meaningful and expressive names" to denote these IDs. The Protégé model was indeed quite strong and expressive, so that every ontology developer could deal with his linguistic needs at a meta-ontological level and find the right place for them. Rare examples exist of Protégé customized applications which deal with multilingualism and/or wider linguistic descriptions [8], but no official agreement was yet established. Later on, with the advent of OWL as a KR standard for the Semantic Web, and with the official release of the Protégé OWL plugin [7], things started to converge towards a minimal agreement for the use of language inside ontologies.

To cope with Protégé standard model, we defined the notion of *terminological slot*, as a slot which is elected by the user to contain different linguistic expressions for concepts. This way, to use Ontoling with standard Protégé, a user only needs to define a proper *metaclass* and *metaslot*, containing the elected terminological slot; naturally, the same slot can be dedicated to instances at class level. Multilingual ontologies can also be supported by creating different slots and selecting each of them as terminological slots during separate sessions of Linguistic Enrichment, with diverse LRs dedicated to the different chosen languages. Glosses can instead be added to the common "documentation" slot which is part of every frame by default.

Conversely, Linguistic Enrichment of OWL Ontologies follows a more predictable path, thanks to OWL's language dedicated Annotation Properties, such as *rdfs:label* and *owl:comment*. When Ontoling recognizes a loaded ontology as expressed in the OWL language, the terminological slot is set by default to *rdfs:label*. In this case the *xml:lang* attribute of the label property is automatically filled with the language declared by the Linguistic Interface (see Fig. 4).

As a further step, we are considering to give a greater emphasis to terms, seeing them no more as labels attached to concepts, but reifying them as concrete ontological elements. Many-to-many relationships can be established between concepts and terms, which can thus be accessed both ways. This approach guarantees greater linguistic awareness over ontological data, and is particularly useful when the conceptual content of Ontologies must be retrieved from documents, user questions and other cases where interaction with natural language content is required. This process is however again far from standardization and thus requires an agreement over the way terms, concepts and their relations are modeled.

## 7 Conclusions and future work

It appears evident that, in a process which has already been widely described and discussed in literature such as Ontology Development, the role of language must not be underestimated. If we believe that knowledge resources will really help in making the Semantic Web dream become true, we have to face the real aspects which characterize the Web as we know it, now.

Thousands of millions of documents which are available on the web are mostly written in natural language; at the same time, people like to interact with computers using even more friendly interfaces, and we do not know better solution than commonly spoken language. A more linguistic awareness could also help semantic search

engines in augmenting the retrieval of proper results, or, at least, in excluding information which is not pertinent to the intention behind the submitted query.

In this work we stressed the need of providing a general framework for dealing with heterogeneous linguistic resources and for exploiting their content in the process of ontology development. Different functionalities for augmenting the linguistic expressivity of existing ontologies or for helping users in developing new knowledge resources from scratch have been identified and implemented in the presented Ontoling plugin for Protégé.

Ontoling, with the Wordnet Interface as its first available plugin, has been adopted by a community of users coming from diverse research areas, from pure linguists approaching ontologies, to ontology developers exploiting specific parts of Wordnet's taxonomical structure as a basis for creating their own domain ontology, up to users needing its main functionalities for adding synonyms to concepts of existing ontologies. With the recent release of the DICT Interface we added a little step in assisting multilingual ontology development and we now look forward other available resources (such as [10, 11]) to be added to Ontoling plugin library.

## References


1. V. R. Benjamins, J. Contreras, O. Corcho and A. Gómez-Pérez. Six Challenges for the Semantic Web. *SIGSEMIS Bulletin*, April 2004.
2. N. Calzolari, J. McNaught, and A. Zampolli EAGLES Final Report: EAGLES Editors Introduction. EAG-EB-EI, Pisa, Italy 1996.
3. R. A. Cole, J. Mariani, H. Uszkoreit, A. Zaenen, and V. Zue, Eds. *Survey of the State of the Art in Human Language Technology*, Cambridge University Press, Cambridge, UK, 1997.
4. M. Dean, D. Connolly, F. van Harmelen, J. Hendler, I. Horrocks, D. L. McGuinness, P. F. Patel-Schneider, and L. A. Stein. OWL Web Ontology Language 1.0 Reference, W3C Working Draft 29 July 2002, http://www.w3.org/TR/owl-ref/.
5. Fellbaum, C.: WordNet - An electronic lexical database. MIT Press, (1998).
6. J. Gennari, M. Musen, R. Fergerson, W. Grosso, M. Crubézy, H. Eriksson, N. Noy, and S. Tu. The evolution of Protégé-2000: An environment for knowledge-based systems development. *International Journal of Human-Computer Studies*, 58(1):89–123, 2003.
7. H. Knublauch, R. W. Fergerson, N. F. Noy, M. A. Musen. The Protégé OWL Plugin: An Open Development Environment for Semantic Web Applications *Third International Semantic Web Conference - ISWC 2004*, Hiroshima, Japan. 2004
8. M.T. Pazienza, A. Stellato, M. Vindigni, A. Valarakos, V. Karkaletsis. Ontology integration in a multilingual e-retail system. *HCI International 2003*, Crete, Greece, 2003
9. M. T. Pazienza, A. Stellato, L. Henriksen, P. Paggio, F. M. Zanzotto. Ontology Mapping to support ontology-based question answering. *Proceedings of the second MEANING workshop.* Trento, Italy, February 2005
10. Emanuele Pianta, Luisa Bentivogli, Christian Girardi. MultiWordNet: developing an aligned multilingual database". In Proceedings of the First International Conference on Global WordNet, Mysore, India, January 21-25, 2002
11. P. Vossen. *EuroWordNet: A Multilingual Database with Lexical Semantic Networks*, Kluwer Academic Publishers, Dordrecht, 1998
12. www.dict.org/bin/Dict
13. www.freelang.com
14. www.tei-c.org/