ROMANIAN JOURNAL OF INFORMATION SCIENCE AND TECHNOLOGY
Volume 7, Numbers 1–2, 2004, 215–231

WMS: Towards a Distributed Network of Semantic Networks

I.D. KOUTSOUBOS, Vassilis ANDRIKOPOULOS,
Manolis TZAGARAKIS, Dimitris CHRISTODOULAKIS

Computers Engineering and Informatics Department, Patras University
Research Academic Computer Technology Institute, Patras, Greece

E-mail: `koutsoub@cti.gr`, `andrikop@ceid.upatras.gr`, `tzagara@cti.gr`, `dxri@cti.gr`

Abstract. The paper aims at presenting the Wordnet Management System, a system that acts as the interconnection and communication liaison between a user and a number of interlinked wordnets. Semantic information is being accessed through a distributed network of servers, forming a large-scale multilingual semantic network. The standalone and monolithic nature of most current wordnet tools limit to a great extent the usage of the developed wordnets. In particular, these tools do not support adequately the needs of collaborative and distributed working groups in heterogenous environments and cannot be easily integrated into the everyday computing environment of users. Lack of supporting collaborative working groups hinders developing and interconnecting the individual wordnets while lack of integration support makes awareness of wordnet at the application level difficult and error prone. Yet, the interconnection of wordnets as well as the utilization of wordnets in various applications are the explicit goals of the WordNet project. In this paper, the design and development of the Wordnet Management System (WMS), a large scale, distributed, service-oriented Wordnet Management System is presented. WMS attempts to overcome the limitations of standalone Wordnet tools. It enables collaborative authoring and interconnection of individual cross-lingual wordnets, and provides developers with the means to facilitate the integration of wordnet resources and services into third-party applications. WMS defines a protocol for loosely coupling individual wordnets, thus exhibiting openness. Individual wordnets may be handled by the WMS with small cost. This distributed nature of the WMS paves also the ground to witness – with respect to multilinguality – network effects in wordnets: a situation in which the utility of an individual Wordnet depends on the number of other wordnets incorporated into the WMS.

1. Introduction

WordNet [4] has been identified as an important resource in the human language technology and knowledge processing communities, due to the fact that WordNet provides structured lexico-semantic information and several monolingual wordnets can be interlinked forming a multilingual network without loosing any properties of their structure. Its usefulness has been reported in many applications such as named entity recognition [11], word-sense disambiguation [17], [1], [15], [16], corpus annotation [5], information retrieval [22], [6], [20], [3]. Almost every NLP application nowadays requires a certain level of semantic analysis and processing. The most important part of this process is semantic tagging : the annotation of each content word with a semantic category.

There is an increasing amount of wordnet resources being made available for NLP researchers. These resources constitute the basic raw materials for building applications such as the above mentioned. Semantic networks standardization is of prime importance in the case of wordnets incorporation in real life applications.

However, current wordnet tools are unable to cope adequately with the new usage of wordnet resources. This is primarily due to the fact that they are standalone, monolithic applications, focusing mainly on editing/authoring and browsing of wordnets by a single user. While editing and browsing constitute important aspects of the wordnet effort, they are not the only ones especially when considering using wordnet resources. The need to interconnect the individual wordnets that reside under different control authorities, as well as the efforts to use wordnet resources in variety of applications, that are even unaware of wordnet resources, imposes new requirements on these tools. Thus, contemporary tools exhibit inabilities with respect to supporting collaborative authoring and browsing of wordnets, large scale sharing of wordnets as well as supporting developers during their tasks of integrating wordnet resources and services in third-party applications[1].

Collaborative and distributed authoring and browsing of wordnets would enable the rapid growth of the wordnet community. There is an increasing amount of wordnet resources being made available, but the cost for interconnecting them through the Inter-Lingual-Index[2] – using the available tools – would be immense. On the contrary, using wordnet tools and systems that are designed with large scale sharing and reuse as well as collaborative work groups in mind, would significantly lessen that cost. Such tools take advantage of the network effects that such system exhibit: a particular wordnet, entering the existing shared, distributed space of wordnets would be automatically connected to the existing wordnets, raising significantly its utilization.

Deploying wordnet resources and services in applications has become an important aspect of the linguistic NLP community. The difficulty of using wordnets in various applications has its roots in the stand alone and monolithic nature of contemporary tools, that require big efforts from developers in converting, importing and transform-

[1] As "third-party application" is defined an application that has been independently developed and thus is not aware of wordnet resources. Examples of third-party applications are the MS-Office suite of tools, MS Internet Explorer, Mozilla etc.

[2] We define the notion of Inter-Lingual-Index in Section 4.1

ing such resources. In tasks related to integrating wordnet in applications, the role of developers is largely neglected. In general, research in NLP can be divided in two categories: linguistic domain research, emphasizing on how users explore NLP and where they need it and what tools would be valuable for them and linguistic systems research that emphasizes on creating the appropriate software infrastructures, in terms of algorithms, tools or entire systems, that bring NLP to the end-user. The absence of linguistic resources and services in the everyday environment of users (except word processors) is not an indication that such resources are not needed but rather that there exists a gab between linguistic domain and system research: while domain researches indicate new areas in which NLP would be useful, discover new ways on using NLP facilities, nevertheless system researchers are unable to cope with these new insights, primarily because their software infrastructure does not explicitly address their task: the development of linguistic tools.

Towards a vision of next-generation tools and services that will enable the widespread development and use of wordnet resources we present a distributed wordnet server architecture in which wordnet servers, analogous to database servers, provide facilities for storing and accessing wordnet data via a common network API. Apart from distributing wordnets over multiple servers the system is capable of distributing wordnet-related services over multiple servers.

2. Limitations of Current Tools

Although current wordnet editors such as VisDic [14] support sufficiently editing and browsing of wordnets, they exhibit shortcomings with respect to shared and interactive workspaces as well as support of linguistic systems developers.

Shared, interactive wordnet workspaces. Development of wordnets is usually carried out by specialized users (i.e. linguists) working in teams. However, most current tools are designed to be used by only one user, one author at a time and many readers. Although the notion of sharing is present in these tools, they support it in an rather awkward and complex way, particularly when considering the end-user. Sharing in these tools is realized by *copying* wordnet resources into the workspace of an user. They allow *importing* of wordnet resources that have been independently developed by different teams. These wordnet resources are usually communicated by email or the World Wide Web. Such copying/importing wordnet resources causes not only problems such as redundancy but also duplicates efforts with respect to multilingual wordnets. The task of creating mappings from one language to the other – using a special structure, named Inter-Lingual-Index – needs to be carried out by each team that imports a wordnet of another team. Moreover, although these tools have some mechanisms to discover conflicts in resources and keep track of changes using versioning facilities, they support these mechanisms not in an *interactive*, *on-line* way. Upon discovery of conflicts, teams need to communicate, to cooperate in order to solve these conflicts. This might also lead to repeating the import of the updated resources. Thus, such tasks are currently accomplished in an off-line or batch and time-consuming setting.

Infrastructure for linguistic system researchers. Development of wordnet tools is not an easy task and poses many challenges that include modeling of wordnets by designing the appropriate database, the suitable visualization of complex wordnet networks, knitting the database and the user interface by defining the necessary software interfaces. These efforts however are targeted primarily towards creation of editing and browsing tools and cannot be reused in other applications. As a result, large parts of these wordnet tools need to be redesigned or transformed in order to meet the requirements of new applications such as graphical wordnet browsers, search engines or data mining frameworks, making the process complex and thus error prone. Current tools fail to act as an *infrastructure* to support linguistic systems developer. Moreover, it is proven very difficult to integrate wordnet resources into third-party applications such as MS-Office or MS-Internet Explorer. Yet, it is the integration into such tools that will allow wordnet and NLP in general to reach their potential in the everyday computing environment of the user, evolving beyond their isolated, "insular" status. Nevertheless, users don't feel comfortable if they have to quit tools they know well even if newer ones may have many more capabilities or if these introduce cognitive overhead such as the need to switch from one tool to the other.

3. Motivation

The main goals of Wordnet Management System are:

- *Monolingual Wordnet Independency*: One of the major principles during the design of the WMS was the independency in the development and manipulation of each wordnet, regardless of its context, i.e. the environment created by the wordnets that this one is connected to. This approach complements in a way the merge approach that was adopted for the BalkaNet project but isn't limited to that, allowing the management of semantic resources an a local level, independently of whether they are inter-connected to others or not.

- *Web access*: An almost de facto requirement in a community like BalkaNet consisting of many different members, the need for access to the system via the Web is made imperative by the size of the data in case they had to be installed and the diversity of access methods that the applications that use them require.

- *Flexible access to semantic data by applications*: The design and development of WMS was mostly motivated by the need for the existence of a system that could be used not only by users but also (and mainly) by applications. But this need for machine readable information also requires a certain degree of interoperability among the system and the applications or other systems that use its services. For this purpose, the system must be able to provide information in a format that can be easily manipulated and transformed into other formats or results.

- *Unified Platform of Wordnet Structure related services*: A critical element in designing a wordnet management infrastructure is the efficient utilization of the wordnet's inherent hierarchical structures under a coherent platform. This would translate into exploiting relations that link the synsets and navigating within the relation trees (or networks in some cases). In this way, the information provided by the position of the synset in a hierarchy can be further used to provide semantic data on tree level or to allow the calculation of structural information like the semantic distance of two synsets that can be necessary in applications like Word Sense Disambiguation and Information Retrieval.

- *Distributed information sources*: The need for the distribution of the information sources stems from the nature of the sources themselves. Since the independence in development and manipulation (and therefore retrieval) was to be maintained, then the information has to be distributed among the wordnet developers. Furthermore, the current trends in system design that are mostly influenced by the Peer-to-Peer paradigm call for the location of the information to 'hidden' to the user, enabling an abstraction between the data and their actual location that can facilitate the development both of new applications and information sources.

- *Platform Independency*: WMS has been envisaged from the beginning as a platform-independent tool that could be used under the majority of the operating systems with the minimum effort possible.

3.1. Distributed Systems

We can summarize the motivations for adopting a distributed architecture for wordnet management:

- *Distributed Information Sources*: wordnet resources may be scattered across multiple physical locations. Access to multiple resources may be mediated and rendered in a uniform way.

- *Sharing*: Applications need to access several services or resources in an asynchronous manner in support of a variety of tasks. It would be wasteful to replicate problem-solving capabilities for each application. Instead it is desirable that the architecture supports shared access to agent capabilities and retrieved information.

- *Complexity Hiding*: A distributed architecture allows specifying different independent problem-solving layers in which coordination details are hidden to more abstract layers.

- *Modularity and Reusability*: A key issue in the development of robust analysis application is related to the enhancement and integration of existing stand-alone applications. Agent may encapsulate pre-existing linguistics applications, which may serve as components for the design of more complex systems. Inter-agent communication languages improve interoperability among heterogenous services providers.

- *Flexibility*: Software agents can interact in new configurations *on-demand*, depending on the information flow or on the changing requirements of a particular decision making task.

- *Robustness*: When information and control is distributed, the system is able to degrade gracefully even when some of the agents are not able to provide their services. This feature is of particular interest and has significant practical implications in natural language processing because of the inherent unpredictability of language phenomena.

- *Quality of Information*: The existence of similar analysis modules able to provide multiple interpretation of the same input offers both 1) the possibility of ensuring the correctness of data through cross-validation and 2) a mean of negotiating the best interpretation(s).

3.2. Web-Services

Definition: A Web service is a software system identified by a URI [RFC 2396], whose public interfaces and bindings are defined and described using XML. Its definition can be discovered by other software systems. These systems may then interact with the Web service in a manner prescribed by its definition, using XML based messages conveyed by Internet protocols.

Given this definition, several technologies used in recent years, such as J2EE, CORBA and CGI scripting, could have been classified as Web Services technology, but were not. The major difference between these technologies and the new breed of technology that are labeled as Web Services is their standardization.This new breed of technology is based on standardized XML [`http://www.w3.org/XML`] (as opposed to the proprietary binary standard), provides a language-neutral way of representing data, and supported globally by most major technology firms.

A Web Service has the following special behavioral characteristics:

- XML-based: By using XML as the data representation layer for all Web Services protocols and technologies that are created, these technologies can be interoperable at their core level. As a data transport, XML eliminated any networking, operating system, or platform binding that a protocol has.

- Loosely coupled: A consumer of a Web Service is not tied to that Web Service directly; the Web Service interface can change over time without compromising the client's ability to interact with the service. A tightly coupled system implies that the client and server logic are closely tied to one another, implying that if one interface changes, the other must also be updated. Adopting a loosely coupled architecture tends to make software systems more manageable and allows simpler integration between different systems.

- Coarse-grained: Web Services technology provides a natural way of defining coarse-grained services that access the right amount of operational logic, abstracting from too fine operations such as individual methods exposed by a programming language.

- Ability to be synchronous or asynchronous: Synchronicity refers to the binding of the client to the execution of the service, blocking or not its operation in the synchronous or the asynchronous mode of communication respectively. Asynchronous capability is a key factor in enabling loosely coupled systems.
- Supports document exchange: One of the key advantages of XML is its generic way of representing not only data, but also complex documents. These documents can be simple, such as when representing a current address, or they can be complex, representing an entire book or RFQ (Request For Quote) document.

But in which way can these characteristics facilitate the implementation of a system for the management of linguistic resources? The most important advantage they provide and in the same time one of the major design choices they support is the notion of neutrality in the description of data. Simons [19] describes five fundamental aspects of the nature of linguistic data:

- The data in a text unfold sequentially; the data model must therefore be able to represent text in proper sequence.
- The data are hierarchically structured; the data model must therefore be able to express hierarchical structures of arbitrary depth.
- The data elements bear information in many simultaneous dimensions; the data model must therefore be able to annotate data objects with many simultaneous properties.
- The data are highly interrelated; the data model must therefore be able to encode associative links between related pieces of data.
- The data are multilingual; the data model must therefore be able to keep track of what language each datum is in.

By using XML, those aspects can be fully realized by using the inherent hierarchical, language-independent and self-describing representation of a XML document to facilitate the description of the data.

4. WMS

4.1. Design Requirements

The main purpose of WMS [8] is to act as the interconnection and communication link between a user/application and a collection of linked monolingual semantic networks.

Multilingual wordnets [23], [21] are being developed around the notion of InterLingualIndex (ILI), an unstructured collection of concepts with the only purpose to provide efficient mapping across the different languages. While EWN's ILI is a partially structured warehouse of English concepts, in BalkaNet ILI (BILI) has been obtained from from Princeton WordNet 2.0, exhibiting thus a wordnet-like structure.

One of the basic requirements that must be fulfilled is the fact that each monolingual wordnet in the basic infrastructure shall participate in the main infrastructure without sacrificing its independency and autonomy.

For the design and implementation of a flexible system there are several design requirements that shall be addressed:

- Platform Independent
- Multiple choices of resource storage
- Easy incorporation of additional resources
- Unified/Independent access to the contained information
- Support for multiple views/representations of data.
- Flexible Incorporation of new services.
- Resource Sharing
- Out-of-the-box linking of newly incorporated data

From its definition, WMS falls into the Data Integration Framework, being able to manage a distributed, dynamic network of homogeneous data. Previous systems built for this purpose [7], [18], [24] are often characterized by a centralized system that controls and manages interactions among distributed information sources in order to serve requests for data. A a consequence, in a distributed environment where no a priori knowledge of the location of specific data is possible, the traditional mediator and federated database approaches are not appropriate. Furthermore, approaches such as [2], [9], [12] that provide a source and query independent mediator do not deal with decentralized systems with participants and information sources location unpredictability. As mentioned in [13] a P2P approach would be a more appropriate solution, since it lacks a centralized structure and promotes the equality among peers and their collaboration only for the time necessary to fulfill a request.

On the other hand a feature that was considered very important during the design of the system, was the ability of the system to provide data to the wider possible set of data consumers, ranging from simple users to industrial solution-based applications. This requirement called for a variety of rendering mechanisms and interfaces with variable complexity for the communication of the users with the system. The ideal solution to this problem would be an API for wordnet access, as described in [10] or an extension of it, covering more recent achievements in the interface technologies like the Web Services technologies.

Considering the abovementioned requirements we conclude to the design of a distributed network of servers, hosting any number of data-sources, that are interconnected to a central registry acting as a directory service. WMS follows a mixed approach regarding the architecture of the system : a client-server model for the clients (browsers, applications and Web Services users) and a distributed server model for the servers themselves. The main components of our system design are:

- *Wordnet Datasource*: Contains the semantic data that form a monolingual Wordnet and provides basic semantic operations.
- *BILI Datasource*: Contains the BILI semantic data, along with a set of ontologies that are interconnected to BILI concepts.
- *WMS Server*: It is the basic type of server in the WMS system, responsible for serving the requests to external clients. It hosts any number of monolingual wordnets and also possesses the knowledge of the existence and capability of service of the other servers and is cooperating with the CWMS in order to get this information.
- *Central WMS Server (CWMS)*: is the main key point in the WMS network. It has a dual role, serving requests for BILI data and domain information and acting as a directory service provider, registering wordnet hosts and distributing this information to the other nodes of the network.

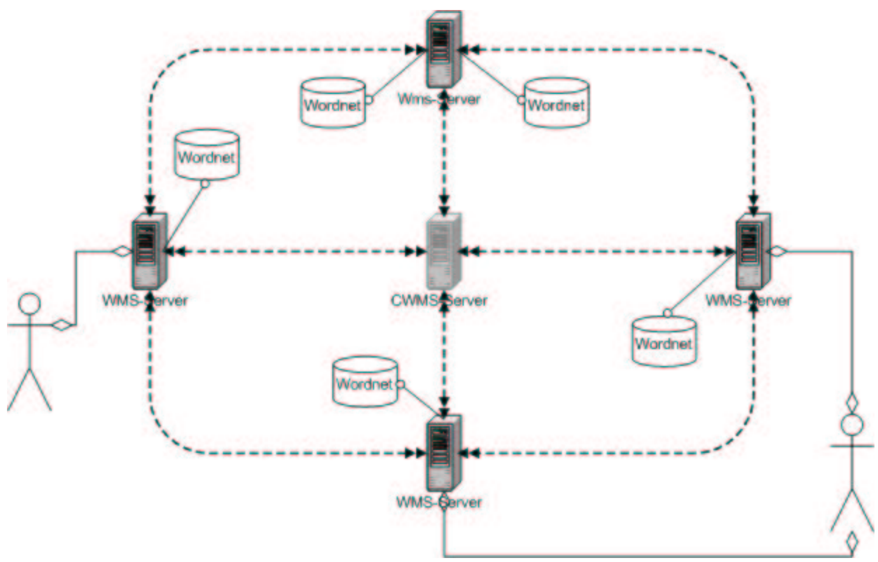

Fig. 1. WMS network.

To better understand the usage of such a distributed architecture we consider the following scenario. WMS-Server A hosts the Greek wordnet, WMS-Server B hosts Turkish wordnet and WMS-Server C hosts no wordnet at all. On startup each server contacts CWMS-Server and registers its wordnet sources. CWMS-Server doesn't store the wordnets – it stores only metadata about the different available wordnets. A client contacts WMS-Server C and requests information about hosted datasources, then WMS-Server C contacts CWMS-Server and returns to client the datasources that are contained into WMS network. The client now has two options, request semantic information about Greek wordnet either through WMS-Server A or through WMS-Server C. If WMS-Server C is chosen then it contacts WMS-Server A and returns the client the result of the requested operation. It is important to realize that through the communication with one and only one server the client has the ability to access the whole of wordnet information contained into the network or choose to directly request information through the responsible server, mostly in cases when the client has better access to the authoritative server (the server that hosts the data) than the server that it initially contacts.

4.2. WMS Implementation

The WMS System is a distributed web-application developed in Java thus making it independent of the underlying system platform (MS-Windows, Linux, etc). The interface of the system comprises of a number of services provided through SOAP. One of the main design principles is the identification of the basic semantic services provided by a semantic network and the combination of these basic services for forming more complicated ones. So we are able to define the basic API that any involved wordnet datasource shall implement, providing data storage independency and flexible incorporation of new storage types. Currently the system uses alternatively indexed XML files produced by the Visdic tool, RDBMS storage in any ANSI-SQL compliant system and specific WMS object database.

The communication between WMS servers, WMS Server and CWMS and WMS server with clients is being performed through SOAP protocol and provided services are described through the use of WSDL in a programming language independent way.

The conceptual design of a WMS-Server is shown in the following diagram:

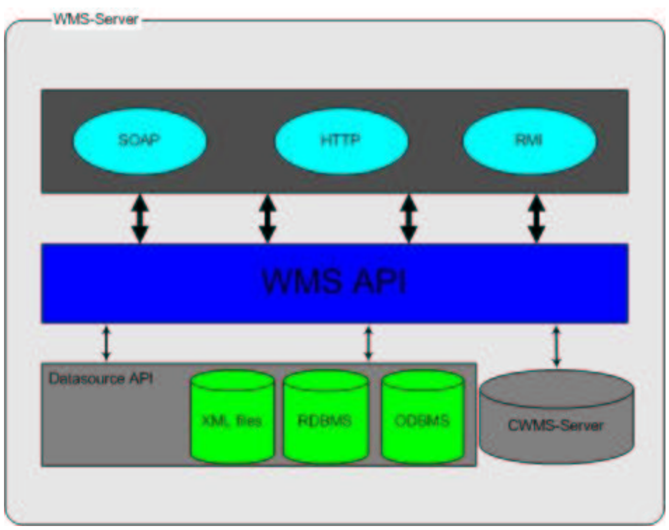

Fig. 2. Conceptual design of a WMS-server.

4.3. WMS Services

The basic services provided by the WMS are described below[3]:

4.3.1. Monolingual Services

These services handle the retrieval of either semantic (i.e. content-related) or statistic (i.e. structure-related) data from each monolingual wordnet. Every wordnet can be accessed by a unique identifier within the context of the WMS network.

getProviders()

Description: Returns information about wordnets that are hosted into WMS network.
Input: None
Output: An array of hosted wordnets, containing each one's (unique) id, server, name, version and natural language.

getSynsetIds(wordnet)

Description: This service provides all synset identifiers that exist into the requested wordnet.
Input: The unique identifier of the wordnet to be queried.
Output: An array of strings, each one containing a synset identifier.

getBaseConcepts(wordnet)

Description: Provides all synset identifiers that belong to the requested base concept set.
Input: The unique identifier of the wordnet to be queried, the Base Concept group.
Output: An array of strings, each one containing a synset identifier of a Base Concept.

getSynsetById(wordnet, synsetid)

Description: Provides information about the synset identified by the specified id in the selected wordnet. Information contains POS (Part-of-Speech), gloss, Base Concept group and senses of the synset.
Input: The unique identifier of the wordnet to be queried, the synset identifier.
Output: The synset that corresponds to the given identifier.

getSynsetByLiteral(wordnet, literal)

Description: Provides information about synsets containing the specified literal in the selected wordnet. Information contains POS, gloss, Base Concept group and senses of the synset.
Input: The unique identifier of the wordnet to be queried, the literal to find.
Output: An array of synsets that meet the query criteria.

[3]The data types mentioned in the description of the services refer to the respective Java data structures and/or custom Java classes that realize these data types. The usage of Web Services allows the decoupling of these data types into either standard basic types like strings, or into well defined structures like synsets.

getSynsetRelations(wordnet, synsetid)

Description: Provides information about the semantic relations of the synset identified by the specified id in the selected wordnet.
Input: The unique identifier of the wordnet to be queried, the synset identifier.
Output: An array of synset's semantic relations.

getSynsetRelationsByRelation(wordnet, synsetid, relation)

Description: Provides information about the semantic relations of the synset identified by the specified id in the selected wordnet.
Input: The unique identifier of the wordnet to be queried, the synset identifier, the relation to find.
Output: An array of synset's semantic relations.

getSynsetTree(wordnet, synsetid, relation)

Description: Provides a tree structure for the requested synset, according to the requested relation, placing the requested synset as root of the tree.
Input: The unique identifier of the wordnet to be queried, the synset identifier, the semantic relation to find.
Output: The tree structure that is formed.

getNodeCount(wordnet, synsetid, relation)

Description: Provides the number of nodes contained in the tree of the requested synset.
Input: The unique identifier of the wordnet to be queried, the synset identifier, the semantic relation to find.
Output: An integer which represents the number of nodes in the tree.

getTreeDepth(wordnet, synsetid, relation)

Description: Provides the number of levels contained in the tree of the requested synset.
Input: The unique identifier of the wordnet to be queried, the synset identifier, the hierarchical relation to find (HYPONYM/HYPERNYM).
Output: An integer which represents the number of levels contained in the tree.

getDistance(wordnet, synsetid1, synsetid2, relation)

Description: Provides the distance between two synsets as the difference of their levels in the tree that is formed if for the specified hierarchical relation these synsets share the same root.
Input: The unique identifier of the wordnet to be queried, the synset identifier, the hierarchical relation to find.
Output: An integer representing the calculated distance.

4.3.2. Multilingual Services

These services can provide the same content-related retrieval of information for a given synset, but this time on a multi-wordnet level by utilizing the common point of reference that provides the BILI or another language-independent structure. The output of these services utilizes a mapping structure called the Hashtable which maps synset information to a (unique) wordnet identifier.

getMSynsetById(wordnet[], synsetid)

Description: Provides information about the synset identified by the specified id for each specified wordnet.
Input: The wordnet identifiers to search, the synset identifier.
Output: A Hashtable containing the synset information for each wordnet.

getMSynsetByLiteral(wordnet[], literal)

Description: Provides information about the synsets that contain the specified literal. by the specified id for each specified wordnet.
Input: The wordnet identifiers to search, the literal to query.
Output: A Hashtable containing the synset information for each wordnet.

4.4. WMS Clients and Applications

On top of the WMS API various clients have been realized, including a graph browser for wordnet trees, an MS .NET client, a plug-in to Microsoft Office allowing thesaurus-style access to WMS semantic data.

The graph browser is a custom application that uses the tree-like structure inherent to the wordnet, due to the interconnection created by the different kind of relations among synsets. WMS in this case is used as the provider of the relational data, leaving the actual representation of the structure to the application itself.

The Microsoft .NET Client for WMS was built as a demonstration tool, using the WSDL document that describes services that are provided by a standard WMS Server. It performs standard wordnet browsing operations, such as search by literal name and synset id, and retrieval of synset information like relations. By these means, it can be used as an ad hoc wordnet browser, but with the additional feature that it can be set to access more than one wordnet (local or remote) at a time.

The purpose of the development of the Microsoft Office plug-in was to provide access to the linguistic data inherent in the multilingual database that is formed by the wordnets of the BalkaNet project to every day applications like Microsoft Word. For this purpose, the plug-in utilizes the services provided by WMS to retrieve data like the synonyms of a given word and provide them to the user as thesaurus-like information. In this way, WMS provides the opportunity for wordnets to be used as a repository of multilingual linguistic information that is available to a multitude of every day applications like text editors, word processors and even internet browsers.

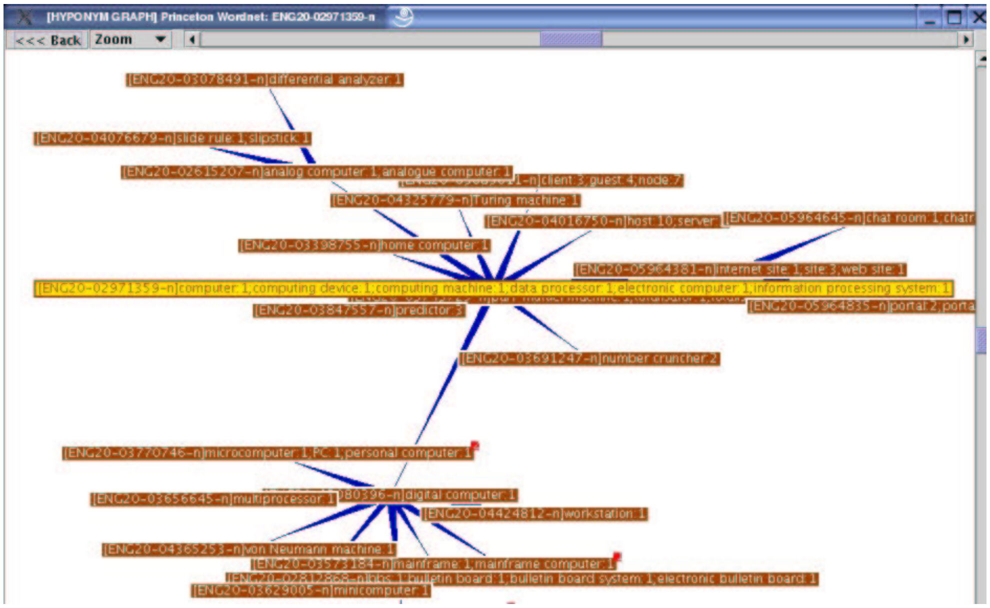

Fig. 3. Graph browsing of wordnet Trees.

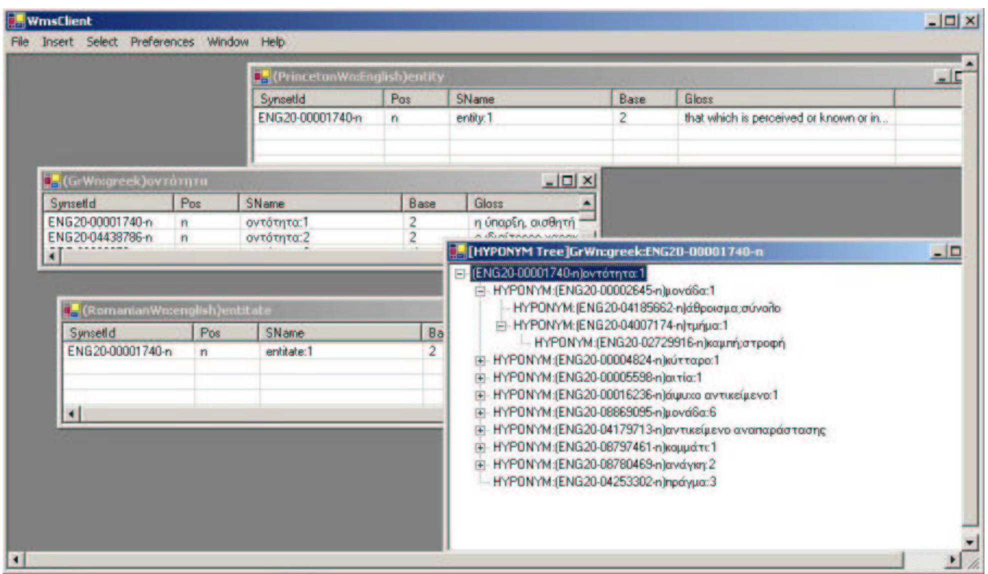

Fig. 4. .NET Client.

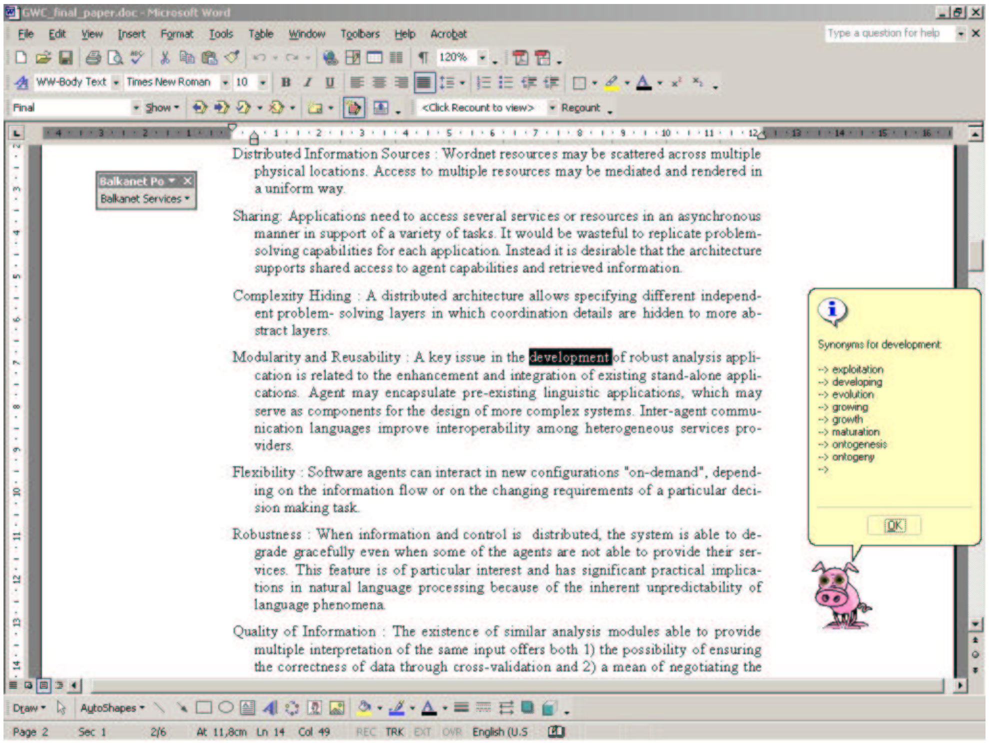

Fig. 5. Microsoft Office Plug-in.

5. WMS Added Value and Future Directions

The main advantages of Wordnet Management System are listed below:

- Open-Ended Platform
- State-of-the-Art technologies
- Distributed management and control
- Flexible access to provided services and data.
- Data Storage Independent.

Wordnet Management System's future directions include :

- Versioning of Datasources
- Full Ontology Support On CWMS.
- Wordnet Authoring

- More Multilingual services
- Incorporation of other Lexical resources.
- Standardization of Data Representation.

6. Conclusions

We have presented the design and development of the Wordnet Management System (WMS), a large scale, distributed, service-oriented Wordnet Management System. WMS attempts to overcome the limitations of standalone wordNet tools. It enables collaborative authoring and interconnection of individual cross-lingual wordnets, and provides developers with the means to facilitate the integration of wordnet resources and services into third-party applications. WMS defines a protocol for loosely coupling individual wordnets, thus exhibiting openness. Individual wordnets may participate the WMS with small cost. Such distributed nature of WMS creates also the ground to witness – with respect to multilinguality – network effects in wordnets: a situation in which the utility of an individual wordnet depends on the number of other wordnets incorporated into the WMS.

References

[1] AGIRRE, E., RIGAU, G., *Word Sense Disambiguation Using Concpetual Density*, in *Proceedings of the COLING Conference*, Copenhagen, Denmark, 1996.

[2] BAYARDO, R.J., BOHRER W., et al., *InfoSleuth: Agent-Based Semantic Integration of Information in Open and Dynamic Environments*, in *Proceedings of the ACM SIGMOD International Conference on Management of Data*, **26**, No. 2, June 1997.

[3] BERTAGNA, F., *Exploiting ItalWordNet Taxonomies in a Question-Answering Classification Task*, in *Proceedings of the 2nd Global Wordnet Conference*, Czech Republic, 2004.

[4] FELLBAUM, C. (ed.), *WordNet: An Electronic Lexical Database*, MIT Press, Cambridge, Mass., 1998.

[5] FELLBAUM, C., GRABOWSKI, J., LANDES, S., *Analysis of a Hand-Tagging Task*, in *Proceedings of the ANLP Workshop on Tagging Text with Lexical Semantics: Why, what, and How?*, Washington D.C., USA, 1997.

[6] GILARRANZ, J., GONZALO, J., VERDEJO, F., *An Approach to Conceptual Text Retrieval Using the EuroWordNet Multilingual Semantic Database*, in *Working Notes of AAAI Spring Symposium on Cross-Language Text and Speech Retrieval*, Stanford, CA, USA, 1997.

[7] GARCIA-MOLINA, H., PAPAKONSTANTINOU, Y., et al., *The TSIMMIS approach to mediation: data models and languages*, Journals of Intelligent Information Systems, **8**, 117–132, 1997.

[8] KOUTSOUBOS, I.D., ANDRIKOPOULOS, V., CHRISTODOULAKIS, D., *WordNet exploitation through a Distributed Network of Servers*, in *Proceedings of the 2nd Global WordnetConference*, Czech Republic, 2004.

[9] LEVY, A., RAJARAMAN, A., ORDILLE, J., *Querying Heterogeneous Information Sources Using Source Descriptions*, in *Proceedings of the 22nd VLDB Conference*, September, 1996.

[10] MIATIDIS, M., ASSIMAKOPOULOS, D., KOUTSOUBOS, I.D., KOUROUSIAS, G., TZAGARAKIS, M., CHRISTODOULAKIS, D., *Access to Wordnets Through API*, *Lexical DataBases, Electronic Linguistic Resources*, Meeting Thessalonica, 2001.

[11] NEGRI, M., MAGNINI, B., *Using WordNet Predicates for Multilingual Named Entity Recognition*, in *Proceedings of the 2nd Global WordnetConference*, Czech Republic, 2004.

[12] NODINE, M., BOHRER, W., HIONG NGU, A.H., *Semantic Brokering over Dynamic Heterogeneous Data Sources in InfoSleuth*, in *Proceedings of the 15th International Conference on Data Engineering*, March 1999.

[13] PANTI, M., PENSERINI, L., SPALAZZI, L., *A pure P2P approach to information integration*, *Tec. Report 2002-02-19*, Istituto di Informatica, University of Ancona, 2002.

[14] PAVELEK, T., PALA, K., *WordNet standardization from a practical point of view*, in *Proceedings of the LREC Workshop on WordNet Structures and Standardisation, and How These Affect WordNet Applications and Evaluation*, Las Palmas, Gran Canaria, 30–34, May 28th, 2002.

[15] RESNIK, P., (a) *Disambiguating Noun Groupings with Respect to WordNet Senses*, in *Proceedings of the 3rd Workshop on Very Large Corpora*, MIT, 1995.

[16] RESNIK, P., (b) *Using Information Content to Evaluate Semantic Similarity in a Taxonomy*, in *Proceedings of the 14th International Joint Conference on Artificial Intelligence (IJCAI)*, Montreal, Canada, 448–453, 1995.

[17] RICHARDSON, R., SMEATON, A.F., *Using WordNet in a Knowledge-Based Approach to Information Retrieval*, in *Proceedings of the 17th BCS-IRSG Colloquium, Crewe*, 1995.

[18] SHETH, A., LARSON, J., *Federated Database Systems for Managing Distributed, Heterogeneous and Anonymous Databases*, in *ACM Transaction on Database Systems*, **22**(3), 1990.

[19] SIMONS, S. F., *CELLAR: A Data Modeling System for Linguistic Annotation*, in *LREC 2000 Workshop*, Athens.

[20] SMEATON, A.F., QUINGLEY, A., *Experiments in Using Semantic Distances between Words in Image Caption Retrieval*, in *Proceedings of the 19th ACM SIGIR Conference*, 174–180, 1996.

[21] STAMOU, S., OFLAZER, K., PALA, K., CHRISTODOULAKIS, D., CRISTEA, D., TUFIŞ, D., KOEVA, S., TOTKOV, G., DUTOIT, D., GRIGORIADOU, M., *BALKANET: A Multilingual Semantic Network for Balkan Languages*, in *Proceedings of the GWA 1st International WordNet Conference*, Mysore, India, Jan. 21–25, 2002.

[22] VOORHEES, E., *Query Expansion using Lexical-Semantic Relations*, in *Proceedings of the 17th ACM-SIGIR Conference*, 61–69, 1994.

[23] VOSSEN, P. (ed.), *EuroWordNet: a multilingual database with lexical semantic networks for European Languages*, Kluwer Academic Publishers, Dordrecht, 1999.

[24] WIEDEHOLD, G., *Mediators in the Architecture of Future Information Systems*, IEEE Computer Magazine, **25**, 38–49, March 1992.