# The Usable Ontology: An Environment for Building and Assessing a Domain Ontology

Michele Missikoff[1], Roberto Navigli[2], Paola Velardi[2]

[1] IASI-CNR, Viale Manzoni 30, Roma, Italy
missikoff@iasi.rm.cnr.it

[2] Dipartmento di Scienze dell'Informazione,
via Salaria 113, 00198 Roma, Italy
{navigli,velardi}@dsi.uniroma1.it

**Abstract.** Experience shows that the quality of the stored knowledge determines the success (therefore the effective usage) of an ontology. In fact, an ontology where relevant concepts are absent, or are not conformant to a domain view of a given community, will be scarcely used, or even disregarded. In this paper we present a method and a set of software tools aimed at supporting domain experts in populating a domain ontology and obtaining a shared consensus on its content. "Consensus" is achieved in an implicit and explicit way: implicitly, since candidate concepts are selected among the terms that are frequently and consistently referred in the documents produced by the virtual community of users; explicitly, through the use of a web-based groupware aimed at consensus building.

## 1 Introduction

The development of the *Semantic Web* [25], aimed at improving the "semantic awareness" of computers connected via the Internet, requires a systematic, computer-oriented representation of the world. Such a world model is often referred to as an ontology. Though the role of ontologies in the Semantic Web solutions is widely recognized, several barriers must be overcome before they become practical and *usable* tools. Once the formal principles and the basic domain concepts have been assessed (a result eventually achieved in many projects), ontology engineers must face the time-consuming and expensive task of populating the ontology and making it accessible to the users of a given virtual community. The absence of powerful tools to support and speed-up this process is a major obstacle to a wide-spread usage of ontologies in web applications. In this paper we present the results of a project aiming at developing a set of integrated methods for semi-automatic learning, verification and maintenance of domain ontologies. Two European projects, *Fetish* [22] and *Harmonise* [23], both in the Tourism domain, provided an application test bed to verify the effectiveness and usability of the proposed methods.

### 1.1 The Usable Ontology

Creating ontologies is a difficult process that involves specialists from several fields. Philosophical ontologists and Artificial Intelligence logicists are usually involved in the task of defining the basic kinds and structures of concepts (objects, properties, relations, and axioms) that are applicable in every possible domain. The issue of identifying these very few "basic" principles, referred to as the *top ontology* (TO), is not a purely philosophical one, since there is a clear practical need of a model which has as much generality as possible, to ensure reusability across different domains [13]. Domain modelers and knowledge engineers are involved in the task of identifying the key domain conceptualizations, and describing them according to the organizational *backbones* established by the Top ontology. The result of this effort is referred to as the *upper domain ontology* (UDO), which includes usually a few hundred application-domain concepts.

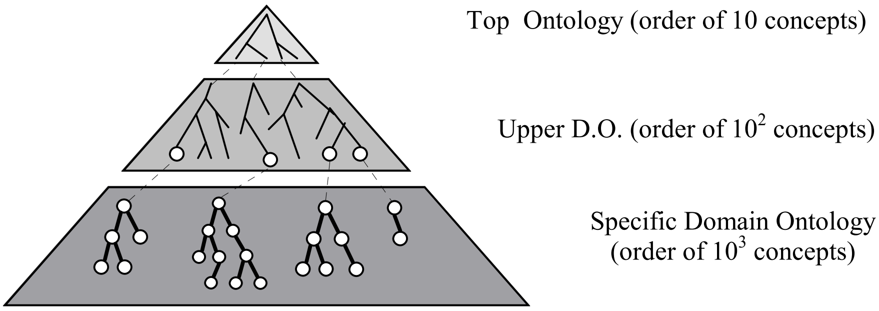
Top Ontology (order of 10 concepts)
Upper D.O. (order of $10^2$ concepts)
Specific Domain Ontology (order of $10^3$ concepts)

**Figure 1**. The three levels of generality of a Domain Ontology.

While many ontology projects eventually succeed in the task of defining an Upper Domain Ontology[1], populating the third level, that we call the *specific domain ontology* (SDO), is the actual barrier that very few projects could overcome (e.g. Wordnet [3], Cyc [6] and EDR [18]), at the price of inconsistencies and limitations.
It turns out that, although ontologies are recognized as crucial resources for the Semantic Web, in practice they are not available, and when available they are <u>not used</u> outside specific research environments[2].
Which features are mostly needed to build *usable* ontologies?

**Coverage**: the domain concepts <u>must be there</u>: the SDO must be sufficiently (for the application purposes) populated. Tools are needed to extensively supporting the task of identifying the relevant concepts and the relations among them.
**Consensus**: decision making is a difficult activity for one person and it gets even harder when there is a group of people that must reach the consensus on a given issue

[1] In fact many ontologies are already available on the Internet including a few hundred more-or-less extensively defined concepts.

[2] For example Wordnet is widely used in the Computational Linguistics research community, but large scale IT applications based on WordNet are not available.

and, in addition, the group is geographically dispersed. When a group of enterprises decide to cooperate in a given domain, they have firstly to agree on many basic issues, i.e., they must reach a consensus of the business domain. Such a common view must be reflected by the domain ontology.

**Accessibility**: the ontology must be easily accessible: tools are needed to easily integrate the ontology within an application that may clearly demonstrate the advantage of the ontology, e.g., improving the ability to share and exchange information through the web.

In this paper we present a general architecture and a battery of systems to foster the creation of such "usable" ontologies. Consensus is achieved both in an *implicit* and *explicit* way. Implicit, since candidate concepts are selected among the terms that are frequently and consistently referred in the documents produced by the virtual community of users; explicit, through the use of a web-based groupware aimed at consensual construction and maintenance of an ontology. Within this frame, the proposed tools are: *OntoLearn*, for the automatic extraction of domain concepts from the thematic web sites, *ConSys*, for the validation of the extracted concepts, and *SymOntoX*, that is the ontology management system.

### 1.2 An Ontology Engineering Environment

Hereafter we shortly outline the proposed software environment. The above mentioned systems have been developed and are being tested in the context of two European projects, Fetish [15] and Harmonise [16], where they are used as the basis of a semantic interoperability platform for small and medium-sized enterprises, operating in the tourism domain.

Figure 2 sketchily reports the proposed ontology engineering method, i.e., the sequence of steps and the intermediate output that are produced in building a domain ontology.

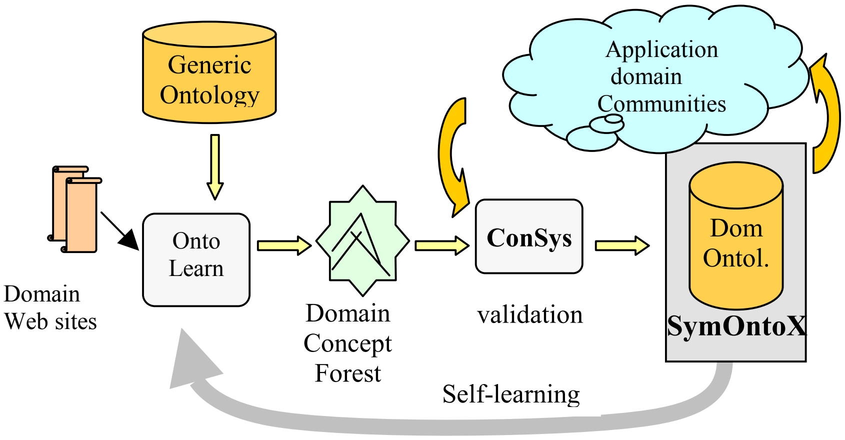

**Figure 2**. The Ontology Engineering Chain.

As shown in Figure 2, *ontology engineering* is an iterative process involving machine concept learning (*OntoLearn*), machine-supported concept validation (*Consys*) and ontology management (*SymOntoX*).
*OntoLearn* explores available documents and related web sites to learn domain concepts, and to detect taxonomic relations among them (*Specific Domain Ontology*). Initially, a generic lexical database (WordNet) is used as a background knowledge.
The subsequent processing step in Figure 2 is *ontology validation*. This is a continuous process supported by a web-based *groupware* aimed at consensus building, called *ConSys* [9]. It is used to achieve a thorough ontology validation with experts and representatives of the communities that are active in the application domain.
*ConSys* operates in connection with *SymOntoX* [27], an *ontology management* system, used by the ontology engineers to define and maintain the concepts and their mutual connections, thus allowing a semantic net to be constructed. SymOntoX uses a knowledge representation method, referred to as *OPAL* (Object, Process, Actor modeling Language) [10], that is an extension of XML based methods, such as DAML+OIL [19]. The ontology engineers use the environment provided by SymOntoX to attach automatically learned concepts sub-trees under the appropriate nodes of the upper domain ontology, to enrich concepts with additional information, and to perform consistency checks.
Figure 2 shows that in the ontology engineering chain several cycles are necessary: the learning cycle highlights the progressively growing role of the domain ontology as a background knowledge for learning new concepts; the validation cycle highlights the many interactions that are necessary between knowledge engineers and domain experts in order to assess the information represented in the domain ontology.
The main focus of this paper is the description of a tool, *OntoLearn*, aimed at extracting knowledge from electronic documents to support the rapid construction of a domain ontology. A brief account of the *ConSys* system is also provided.
The rest of the presentation is organized as follows: in Section 2 we describe in more detail the OntoLearn system: Section 2.1 describes the method to extract terminology from web sites, Section 2.2 presents the knowledge-based semantic interpretation method, along with a summary of the knowledge representation scheme, Section 2.3 describes the creation of a Specific Domain Ontology, and Section 2.4 presents an evaluation of *OntoLearn*. Section 3 briefly describes the validation groupware, *Consys*. Further research and expected outcomes are discussed in the Conclusion.

## 2 The OntoLearn System

Figure 3 shows the architecture of the *OntoLearn* system. There are three main phases: First, a domain terminology is extracted from available documents in the application domain (specialized web sites or documents exchanged among members of a virtual community), and filtered using statistical techniques and documents in different domains for contrastive analysis. Second, terms are semantically interpreted, i.e., we associate unambiguous *concept* names to the extracted terms. *Automatic semantic interpretation is a novel aspect of our research, since in the literature the task of associating terms to concepts is a burden of the ontology engineers* [7, 16].

Third, taxonomic (i.e., generalization/specialization) and similarity relations among concepts are detected, and a *Specific Domain Ontology* (hereafter SDO) is generated. Ontology matching (i.e., the integration of SDO with the existing upper ontology) is performed in connection with *SymOntoX* and *ConSys*.

Initially, we assume that only a small *domain upper ontology* is available (a realistic assumption indeed), therefore a semantic interpretation is based on external (non domain-specific) knowledge sources, such as *WordNet* [3, 28] and the semantically tagged corpus *SemCor* [26]. WordNet is a large lexical knowledge base (described later in more detail), whose popularity is recently growing even outside the computational linguistic community. SemCor is a corpus of semantically annotated sentences, where every word is annotated with a sense tag, selected within the WordNet sense inventory for that word.

As soon as the ontology engineering and validation processes result in a sufficiently rich *domain ontology*, the role of the latter in automatic concept learning progressively overcomes that of WordNet. Eventually, new terms are semantically disambiguated and taxonomically organized using only the information already stored in the domain ontology.

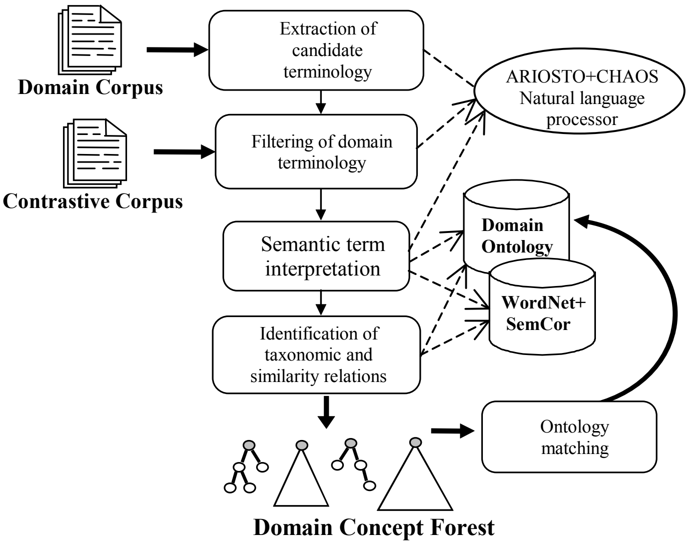
Domain Corpus
Extraction of candidate terminology
ARIOSTO+CHAOS Natural language processor
Contrastive Corpus
Filtering of domain terminology
Semantic term interpretation
Domain Ontology
WordNet+ SemCor
Identification of taxonomic and similarity relations
Ontology matching
Domain Concept Forest

**Figure 3**. Architecture of OntoLearn.

## 2.1 Identification of Relevant Domain Terminology

Terminology is often considered as the surface appearance of relevant domain concepts. The objective of this phase is to extract from the available documents a domain terminology. The domain-related documents are retrieved browsing web sites with an initial set of domain terms[3], and then progressively specializing the search when new terms are learned.

We use a linguistic processor, ARIOSTO+CHAOS [1], to extract from the domain documents a list of *syntactically plausible* terminological patterns, e.g., compounds (*credit card*), prepositional phrases (*board of directors*), adjective-noun relations (*manorial house*). Then, two measures based on information theory are used to filter out non-terminological (e.g., *last week*) and non-domain specific terms (e.g., *net income* in a Tourism domain). The first measure, called *Domain Relevance*, computes the conditional probability of occurrence of a candidate term in the application domain (e.g., Tourism), relative to other corpora that we use for a contrastive analysis (e.g., Medicine, Economy, Novels, etc.). The second measure, called *Domain Consensus*, computes the *entropy* of the probability distribution of a term across the documents of the application domain. The underlying idea is that only terms that are *frequently* and *consistently* referred in the available domain documents reflect some *implicit* *consensus* on the use of that term. These two measures have been formally defined and extensively evaluated in [14] and [15].

Let *T* be the terminology extracted after the filtering phase. Using simple string inclusion, we generate a *forest* of *lexicalized trees*. Figure 4 is an example of lexicalized tree $\mathfrak{I}$ extracted from our Tourism corpus.

However, lexicalized trees do not capture many taxonomic relations between terms, for example between *public transport service* and *bus service* in Figure 4.

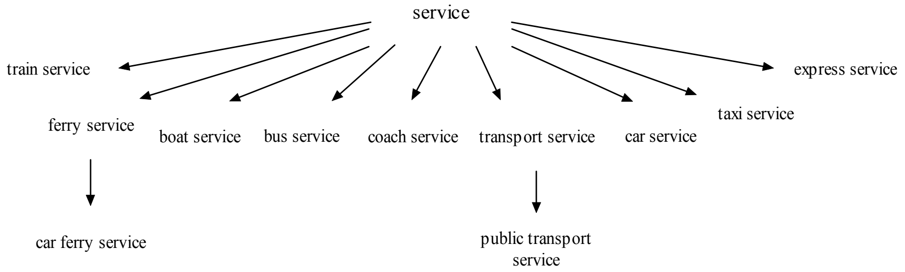

**Figure 4.** A lexicalized tree.

## 2.2 Semantic Interpretation of Terms

The process of *semantic interpretation* is one that associates to each term $t = w_n \cdot \ldots \cdot w_2 \cdot w_1$ (where $w_i$ is an atomic word) the appropriate *concept name*. The idea is that, though the string *t* is usually not included in the start-up ontology, we expect to find a conceptual entry for the various senses of $w_i$ ($i$=1, …, $n$): for example, though there are no concepts for "*room service*", we may find concept descriptions for "*room*" and "*service*" individually. Therefore, it should be possible to *compositionally*

[3] In our application domain, an initial upper ontology of about 300 terms was available.

*create a definition* for *t*, selecting the *appropriate* (given the context *t*) concept definition for each $w_i$ ($i$=1, …, $n$).
As we said, we use WordNet as a start-up ontology, since the upper domain ontology is initially quite poor. In WordNet, a word sense is uniquely identified by a set of terms called *synset* (e.g., for the sense #3 of *transport*: { transportation#4, shipping#1, transport#3 }), and a textual definition called *gloss* (e.g. "*the commercial enterprise of transporting goods and materials*"). Synsets are taxonomically structured in a lattice, with a number of "root" concepts called *topmost* (e.g., { entity#1, something#1 }). WordNet includes over 120,000 words and over 170,000 synsets, with an average ambiguity of 1.4, but very few domain terms are present: for example, *transport* and *company* are individually included, but not "*transport company*" as a unique term.
Formally, a *semantic interpretation* is defined as follows: let $t = w_n \cdot \ldots \cdot w_2 \cdot w_1$ be a valid term belonging to a lexicalized tree $\mathfrak{I}$. The process of semantic interpretation is one that associates to each word $w_k$ in *t* the appropriate WordNet synset $S^k$. The *sense* of *t* is hence defined as:

$$S(t) = \bigcup_k S^k,\ S^k \in Synsets(w_k) \text{ and } w_k \in t.$$

where $Synsets(w_k)$ is the set of synsets each representing a sense of the word $w_k$.

For instance:

*S*("transport company") = { { transportation#4, shipping#1, transport#3 },
{ company#1 } }

corresponding to sense #1 of *company* ("*an institution created to conduct business*") and sense #3 of *transport*, previously reported.

Semantic interpretation is achieved by intersecting semantic information associated to each alternative sense of the words in *t*, and then selecting the "best" intersection. Semantic information is extracted from WordNet and represented in the form of a semantic net fragment, according to a representation scheme described in the next sub-section.

### 2.2.1 Semantic Representation of Concepts

For each sense of a word, several other types of semantic relations are supplied in WordNet, though these relations are not systematically and formally defined. As a first effort, we tried to establish a connection between semantic relations in WordNet and the concept representation method adopted in OPAL.
According to OPAL [10], an ontology is a *semantic net*, constructed by supplying a set of concepts and their semantic relationships. The list of relationships is briefly reported in what follows. In each description, a reference is made to the linguistic counterpart in WordNet, and a graphic symbol is reported. The latter will be used in constructing the diagrams (semantic nets) presented in the next sub-sections.

*Generalization* – This is an asymmetric relation, often indicated as *ISA* relation, that links a concept to its more general concepts (e.g. Hotel ISA Accomodation).
Its inverse is called *specialization*. In the linguistic realm this relation, defined between *synsets*, is called *hyperonymy* ($\xrightarrow{@}$) and its inverse *hyponymy* ($\xrightarrow{\sim}$).
*Aggregation* – This is an asymmetric relation that connects a concept representing a whole to another representing a component. It is often indicated as *PartOf* relation (e.g. Reception PartOf Hotel).
Its inverse is called *decomposition*. In the linguistic realm this relation, defined between *synsets*, is called *meronymy* ($\xrightarrow{\#}$), and *holonymy* ($\xrightarrow{\%}$) its inverse.
*Similarity* – This is a symmetric relation that links two concepts that are considered similar in the given domain. A similarity degree is often indicated (e.g. Hotel SimilarTo[0.8] Motel).
In the linguistic realm this relation, defined between *synsets*, is called *synonymy* when the similarity degree is 1[4], while *similarity* ($\xrightarrow{\&}$) and *correlation* ($\xrightarrow{\wedge}$) are used to indicate progressively weaker levels of similarity. In WordNet there is also a *dissimilarity* relation, *antonymy* ($\xrightarrow{!}$), for example *liberal* and *conservative*, indicating a degree of similarity =0. Furthermore, the relation *pertonymy* ($\xrightarrow{\backslash}$) relates the nominal and adjectival realization of a concept (e.g. *mother* and *maternal*).
*Relatedness* – This is a semantic relation that connects two concepts symmetrically related in the given domain. This relation assumes specific, domain dependent, interpretations. For example, in: Hotel RelatedTo Airport, the relation subsumes *physical proximity*. This weakly defined relation does not have a counterpart in WordNet, but it can be induced from concept definitions and from semantically annotated sentences in the SemCor corpus. Parsing the definitions (glosses) of a given concept, and the semantically annotated sentences including that concept, we generate a linguistic counterpart of "relatedness", represented by the *gloss* relation ( $\xrightarrow{gloss}$ ) and the *topic* relation ( $\xrightarrow{topic}$ ). The idea is that, if a concept $c_2$ appears in the definition of another concept $c_1$, or if $c_2$ appears in the near proximity of $c_1$ in an annotated sentence including $c_1$, then $c_1$ and $c_2$ are "related", i.e. $c_1 \xrightarrow{gloss} c_2$ or $c_1 \xrightarrow{topic} c_2$, respectively. For example, parsing the SemCor sentence: "*The rooms(#1) were very small but they had a nice view(#2)*" produces: *room*#1 $\xrightarrow{topic}$ *view*#2, while parsing the WordNet gloss for tourist#1 "someone who travels for pleasure" produces : tourist#1 $\xrightarrow{gloss}$ travel#2. Notice that the labels "gloss" and "topic" only refer to the *source* of the detected relation (SemCor, or WordNet glosses), not to its meaning. The semantic nature of the relation remains underspecified.

[4] Strict synonyms are those belonging to the same synset.

### 2.2.2 Concept Disambiguation

In order to provide a semantic interpretation for a complex term, all its atomic components must be disambiguated, i.e., the correct sense (given the context) for each word must be identified. To disambiguate the words in a term $t = w_n \cdot \ldots \cdot w_2 \cdot w_1$ we proceed as follows:

a) If $t$ is the first analyzed element of $\mathfrak{I}$**,** manually disambiguate the root node ($w_1$ if $t$ is a compound) of $\mathfrak{I}$.

b) For any $w_k \in t$ and any synset $S_i^k$ of $w_k$ (the $i$-th synset that WordNet defines for $w_k$) create a *semantic net SN*. Semantic nets are automatically created using the semantic relations described in the previous sub-section, extracted from WordNet and SemCor (and possibly from the Upper Domain Ontology).

To reduce the size of a SN, concepts at a distance greater than 3 arcs from the SN center, $S_i^k$ , are not considered. Figure 5a is an example of SN generated for sense #1 of *airplane*.

Let then $SN(S_i^k)$ be the semantic network for sense $i$ of word $w_k$.

c) Starting from the "head" $w_1$ of $t$, and for any pair of words $w_{k+1}$ and $w_k$ ($k$=1,…,$n$-1) belonging to $t$, intersect alternative pairs of SNs. Let $I = SN(S_i^{k+1}) \cap SN(S_j^k)$ be one such intersection for sense $i$ of word $k$+1 and sense $j$ of word $k$. Note that in each step $k$, the word $w_k$ is already disambiguated, either manually (for $k$=1) or as a result of step $k$-1.

d) For each alternative intersection, identify common *semantic patterns* in $I$, and select the sense for $w_{k+1}$ producing the “strongest” intersection[5].

To identify common semantic patterns several heuristic rules are used, e.g.:

$$\text{(1)}\ \exists\, G,M \in Synset_{WN} : S_1 \xrightarrow{gloss} G \xrightarrow{@\ \le 3} M \xleftarrow{\le 3\ @} S_2$$

where $Synset_{WN}$ is the whole set of synsets in WordNet and the heuristic (named "gloss+parallelism") reads: "*given two central concepts $S_1$ and $S_2$, there exist two concepts G and M in $SN(S_1) \cap SN(S_2)$ such that G appears in the gloss of $S_1$ and both G and $S_2$ reach the concept M through a hyperonymy path of length ≤3* ".

Figure 5b is an example of one such intersection for *transport#3* and *company#1*. The bold arrows identify a pattern matching the "gloss+parallelism" heuristics (rule 1 above):

$$\text{transport\#3} \xrightarrow{gloss} \text{enterprise\#2} \xrightarrow{@\ 1} \text{organization\#1} \xleftarrow{2\ @} \text{company\#1.}$$

[5] The algorithm is here oversimplified for sake of space. 11 heuristics are used to identify different semantic paths between SNs, and some amount of backtracking occurs between steps c) and d). Each SN intersection is evaluated by a score vector where each heuristic contributes to one of its component. The “strongest intersection” is given by the maximum score vector.

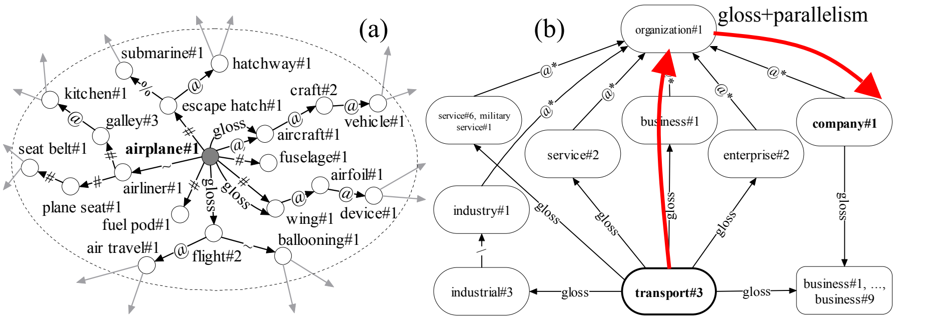

**Figure 5**. a) example of semantic net for *airplane#1* b) example of intersecting semantic patterns for *transport#3* and *company#1*.

## 2.3 Creating a Specific Domain Ontology

Initially, all the terms in a tree $\mathfrak{I}$ are independently disambiguated. Subsequently, taxonomic information in WordNet (or in the upper domain ontology) is used to detect *is-a* relations between *concepts*, e.g., *ferry service* $\xrightarrow{@}$ *boat service*.

In this phase, since all the elements in $\mathfrak{I}$ are jointly considered, some interpretation error produced in the previous disambiguation step is corrected. In addition, certain concepts are *fused* in a unique concept on the basis of pertonymy, similarity and synonymy relations (e.g. respectively: *manor house* and *manorial house*, *expert guide* and *skilled guide*, *bus service* and *coach service*). Notice again that we detect semantic relations between *concepts*, not words. For example, *bus#1* and *coach#5* are synonyms, but this relation does not hold for other senses of these two words. Each lexicalized tree $\mathfrak{I}$ is finally transformed in a *domain concept* tree $\Upsilon$. Figure 6 shows the concept tree obtained from the lexicalized tree of Figure 4.

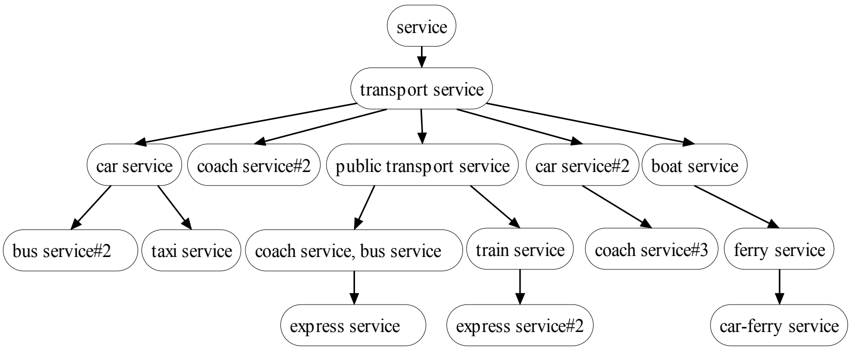

**Figure 6**. A Domain Concept Tree.

For clarity, in Figure 6 concepts are labeled with the associated terms (rather than with synsets), and numbers are shown only when more than one semantic interpretation holds for a term, as for *coach service* and *bus service* (e.g., sense #3 of "bus" refers to "old cars" in WordNet).

### 2.4 Evaluation of the OntoLearn System

OntoLearn is a knowledge extraction system aimed at improving human productivity in the time-consuming task of building a domain ontology. Though a complete field evaluation is still in progress within the Harmonise project using the Consys groupware (see next Section), some crude fact indicates the validity of our method. Our experience in building a tourism ontology for the European project Harmonise reveals that, after one year of ontology engineering activities, the tourism experts were able to release the most general layer of the tourism ontology, comprising about 300 concepts. Then, we decided to speed up the process developing the *OntoLearn* system, aimed at supporting the ontology engineering tasks. This produced a significant acceleration in ontology building, since in the next 6 months[6] the tourism ontology reached 3,000 concepts.

The OntoLearn system has been also evaluated independently from the ontology engineering process. We extracted from a 1 million-word corpus of travel descriptions (downloaded from Tourism web sites) a terminology of 3840 terms, manually evaluated[7] by domain experts participating in the Harmonise project. We obtained a precision ranging from 72.9% to about 80% and a recall of 52.74%. The precision shift is motivated by the well-known fact that the intuition of experts may significantly differ[8]. The recall has been estimated by submitting a list of 6000 syntactic candidates (first step of Section 2.1) to the experts, requiring them to mark truly terminological entries, and then comparing this list with that obtained by our statistical filtering method described in Section 2.1.

The authors personally evaluated the semantic disambiguation algorithm described in Section 2 using a test bed of about 650 extracted terms, which have been manually assigned to the appropriate WordNet concepts. These terms contributed to the creation of 90 syntactic trees. The entire process of semantic disambiguation and creation of domain trees has been evaluated, leading to an overall 84.5% precision. The precision grows to about 89% for highly structured sub-trees, as those in Figure 6. In fact, the phase described in Section 2.3 significantly contributes at eliminating disambiguation errors (in the average, 5% improvement). We also analyzed the individual contribution of each of the heuristics mentioned in Section 2.2.2 to the performance of the method, but a detailed performance report is omitted here for sake of space. The results of this performance analysis led to a refinement of the algorithm and the elimination of one heuristic.

[6] The time span includes also the effort needed to test and tune OntoLearn. Manual verification of automatically acquired domain concepts actually required few days.

[7] Here manual evaluation is simply deciding whether an extracted term is relevant, or not, for the tourism domain.

[8] This very fact stresses the need of a consensus building groupware, as Consys.

## 3. Creating a "Consensus": the ConSys System

As we mentioned in the previous sub-section, a full evaluation of the Tourism ontology, called *OntoTour*, is still in progress. A specific groupware has been conceived to facilitate decision-making and the creation of consensus about the *content* of the ontology. The system, called *ConSys*, is briefly described in this section. More details are given in [9].

Group decision-making is a very difficult activity. Difficulties even increase if the participants do not meet face-to-face, but are geographically dispersed and work mainly asynchronously, communicating via the Internet.

*ConSys* aims at supporting the group of domain experts in the discussion and decision-making process required by ontology building. Essentially, *ConSys* creates a virtual space where the members of the decision group can meet and interact. The main characteristics of *ConSys* are:

- Distributed, open environment, accessible via the Internet, by using a web browser;
- Facilitated organizational communication, by means of predefined interaction templates;
- Decision-making support, provided by specific functions and roles;
- Enhanced document management, for consultation during decision-making;
- Group dynamics carefully conceived, drawn from approaches like: MDM (Multi-participant Decision-Making), GDSS (Group Decision Support System), Formal Consensus techniques [2], speech act theory [12];
- Discussion rules enforcement, e.g., message length or frequency.

The group operates in a virtual space organized into four different virtual rooms (See Figure 7), specialized in terms of activities performed therein.

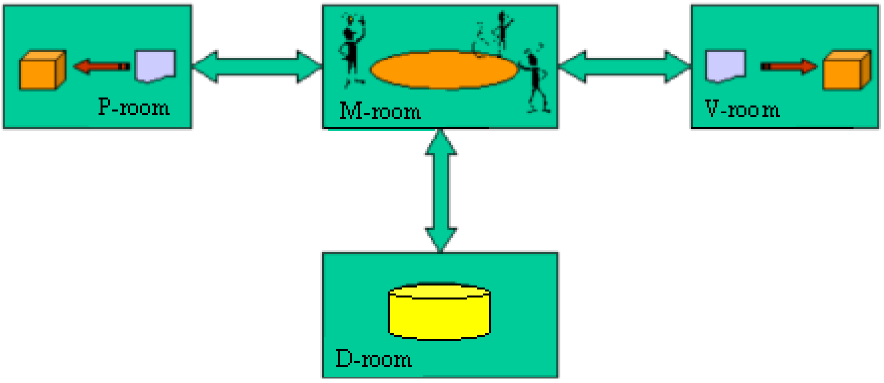

**Figure 7**. The rooms of ConSys.

**Meeting Room** - This is the main room (*M-room*), where the participants meet to debate the issue at hand, express their opinion, make proposals and ask the group to agree. There are precise rules that must be followed by the participants in the M-room. Primarily, they must interact according to a number of predefined *speech acts*, as detailed in [9].

**Documentation Room** – This is a space (*D-room*) where all the relevant documents are stored and made available to all the group members. To support their positions, members may include documents, referred to as *supporting evidence*[9]. In the *D-room* there are also complete accounts of previous decisions, with records of discussions and related supporting evidence.

**Polling Room** – This room (*P-room*) is opened when polling is necessary. The result of a poll has an orientation value, therefore its outcome is not mandatory nor binding for the group.

**Voting Room** – The primary goal of *ConSys* is to facilitate the reaching of a large consensus on the ontology entries. However, if the group gets stalled with two (or more) opposing positions, a voting is required. Then, the participants are asked to enter the *voting room* (*V-room*) to cast their ballot. The result of voting will decide which proposals will be actually included in the ontology.

The construction of an ontology typically starts from a first set of concepts (the Upper Domain Ontology *kernel*) that is generally less controversial and therefore can be approved by the group in a short time. At the same time, it represents a sort of training set for the group, allowing members to progressively know each other, and to get acquainted with the fundamental mechanisms that regulate the group activities. Starting with a least controversial matter is very useful, since initially problems mainly emerge from misunderstandings, unclear statements, involuntary violation of the rules. Therefore, in the first phase, the controversy will be more formal than substantial.

Once the group has accepted the ontology kernel, the actual ontology validation activity starts. The interactions among participants take place according to a predefined set of *speech acts* [17]. Each member of the decision group must check in at the M-room, then he/she can send memos to the group. A memo is classified according to the speech act that represents the intent of the sender.

## Conclusions

In this paper we presented an ontology engineering architecture aimed at facilitating the task of populating domain ontologies and building a shared consensus about their actual content.

The work presented in this paper is novel in several aspects:

- Many methods have been proposed to extract domain terminology or word associations from texts and use this information to build or enrich an ontology. *Terms* however are put in a one-to-one correspondence with domain *concept*

[9] For example, the supporting evidence of a concept proposed by OntoLearn is the set of automatically retrieved sentences including its lexical realization (terminology), and its related definition (Section 2.2).

*names*, while we perform a *semantic interpretation*. By doing so, we can automatically determine that, for example, *swimming pool* is a kind of *hotel facility*, and *bus service* is a kind of *public transportation service*. This has clear implications, for example, in automatic document indexing[10].

- Thanks to semantic interpretation, we are able to detect not only taxonomic, but also other types of relations (e.g., similarity, pertonymy). The amount of extracted semantic relations is being extended in our on-going work, exploiting the information obtained from the intersections of semantic nets SNs (see Figure 5b).
- Though WordNet is not an ontological standard for the Semantic Web, it is *de facto* one of the most widely used general purpose lexical database, as also witnessed by the considerable funding, devoted by the European Community to its extension (for example, the EuroWordNet project [21]). An explicit relation between a domain ontology and WordNet may favour interoperability and harmonization between different ontologies.
- Ontology learning issues have been considered in strict connection with ontology engineering and validation issues. We put a special emphasis on the notion of *consensus*, since an ontology, where the relevant concepts are absent or are not conformant with a domain view of a given community, will be scarcely used, or even disregarded. In our system consensus is achieved both in an implicit and explicit way. Implicit, since the relevant concepts are captured based on their systematic appearance in the documents shared by a virtual community of users. Explicit, since we developed a groupware aimed at the ontology building process.

## References

1. Basili R., Pazienza M.T. and Velardi P. *An Empirical Symbolic Approach to Natural Language Processing*, Artificial Intelligence, n. 85, pp.59-99, (1996).
2. Buttler C.T. and Rothstein, A. *A guide to Formal Consensus*. Food not Bombs Publishing, (1987).
3. Fellbaum, C. *WordNet: an electronic lexical database*, Cambridge, MIT press, (1995).
4. Hirst M., St-Onge D. *Lexical chains as representations of context for the detection and correction of malapropisms*. In C. Fellbaum, editor, WordNet: An electronic lexical database and some of its applications. The MIT Press, Cambridge, MA, (1997).
5. Harabagiu S. and Moldovan D. *Enriching the WordNet Taxonomy with Contextual Knowledge Acquired from Text*. AAAI/MIT Press, (1999).
6. Lenat, D.B. *CYC: a large scale investment in knowledge infrastructure*, in Communication of the ACM, vol. 3, N. 11.
7. Maedche A. and Staab S. *Semi-automatic Engineering of Ontologies from Text* Proceedings of the Twelfth International Conference on Software Engineering and Knowledge Engineering (SEKE'2000) (2000).

[10] Automatic document indexing is an application we are currently looking at, in order to measure the effectiveness of our term extraction and interpretation method.

8. Milhalcea, R. and Moldovan. D. *eXtended WordNet: progress report.* NAACL 2001 Workshop on WordNet and Other Lexical Resources, Pittsbourgh, June (2001).

9. Missikoff M., Wang X.F., *Consys - A Group Decision-Making Support System For Collaborative Ontology Building*, in Proc. of Group Decision & Negotiation 2001 Conference, La Rochelle, France, (2001).

10. Missikoff M., *OPAL - A Knolwedge-Based Approach for the Analysis of Complex Business Systems*, LEKS, IASI-CNR, Rome, (2000).

11. Morin E., *Automatic Acquisition of semantic relations between terms from technical corpora*, Proc. of 5th International Congress on Terminology and Knowledge extraction, TKE-99, (1999).

12. Searle, J.R and Vanderveken, D. *Foundations of Illocutionary Logics*, Cambridge University Press, (1985).

13. Smith, B. and Welty, C. *Ontology: towards a new synthesis*, Formal Ontology in Information Systems, ACM Press, (2001).

14. Velardi P., Missikoff M. and Basili R. *Identification of relevant terms to support the construction of Domain Ontologies*. ACL-EACL Workshop on Human Language Technologies, Toulouse, France, July (2001).

15. Velardi P., Missikoff and P. Fabriani *Using Text Processing Techniques to Automatically enich a Domain Ontology* . Proc. of ACM Conf. On Formal Ontologies and Information Systems, ACM_FOIS, Ogunquit, Maine, October (2002).

16. Vossen P. *Extending, Trimming and Fusing WordNet for technical Documents*, NAACL 2001 workshop on WordNet and Other Lexical Resources, Pittsbourgh, July (2001).

17. Winograd, T. *A Language/Action Perspective on the Design of Cooperative Work*, in Computer Supported Cooperative Work: A Book of Readings, I. Greif (ed) Morgan Kauffmann, (1988).

18. Yokoi T. *The EDR electronic dictionary*, Communications of the ACM, vol. 38, N. 11.

## Web Sites Citations

19. DAML+OIL http://www.daml.org/2001/03/daml+oil-index

20. ECAI-2000 1st Workshop on *Ontology Learning* http://ol2000.aifb.uni-karlsruhe.de/

21. EuroWordNet http://www.hum.uva.nl/~ewn/

22. Fetish EC project ITS-13015 http://fetish.singladura.com/index.php

23. Harmonise EC project IST-2000-29329 http://dbs.cordis.lu

24. IJCAI-2001 2nd Workshop on *Ontology Learning* http://ol2001.aifb.uni-karlsruhe.de/

25. Semantic Web Community Portal http://www.semanticweb.org/index.html

26. SemCor *The semantic concordance corpus* http://mind.princeton.edu/wordnet/doc/man/semcor.htm

27. SymOntos, a symbolic ontology management system http://www.symontos.org

28. WordNet 1.6 http://www.cogsci.princeton.edu/~wn/w3wn.html