Semantic Resolution for E-Commerce

Yun Peng[1], Nenad Ivezic[2], Youyong Zou[1], and Xiaocheng Luan

[1] Department of Computer Science and Electrical Engineering
University of Maryland Baltimore County
Baltimore, MD 21250, USA
{ypeng, yzou1, xluan1}@cs.umbc.edu
[2] National Institute of Standards and Technology
100 Bureau Drive, Gaithersburg, MD 20899, USA
nivezic@nist.gov

Abstract. In this paper we describe our on-going research project on resolving semantic difference for multi-agent systems (MAS) in electronic commerce. The approach we are taking can be characterized by the following: 1) agents in a MAS are allowed to have their own specific ontologies, which are defined on top of a shared base ontology; 2) concepts and their classes in these ontologies are represented as frame-like structures based on DAML+OIL semantic markup language; 3) the semantic differences between heterogeneous agents are resolved in runtime through inter-agent communication; and 4) the resolution is viewed as an abductive reasoning process, and thus necessarily involves approximation and default reasoning.

1 Introduction

Understanding the meaning of messages exchanged between software agents has long been recognized as one of the key problems to achieving interoperability in multi-agent systems (MAS). Forcing all agents to use a common vocabulary defined in one or more shared ontologies is an oversimplified solution especially when these agents are designed and deployed independently of each other. This is particularly the case for agent applications in E-Commerce whose business operation environment can be seen as 1) a huge, ***open*** marketplace (accommodating many companies, each capable of entering and leaving the market freely); 2) involving ***dynamic*** relationships (with partnerships formed and dissolved easily and frequently); and 3) containing ***heterogeneous*** representations of individual agents for different enterprises [4]. It is therefore practically infeasible and unwise to restrict all agents to using the same vocabulary (and with the same semantic understanding of the terms in the vocabulary) for exchanging information. It is also impractical to require inter-ontology translation services be available between involved ontologies prior to the deployment of the agent system. Semantic differences should be allowed and be resolved when they arise during agent interaction. These points are captured by the following assumptions we make for our work. (Similar assumptions have also been made by others [1, 19]):

- **Assumption 1:** interacting agent share a base ontology or a set of base ontologies.
- **Assumption 2**: agents are allowed to use different ontologies, defined on top of the base ontology.
- **Assumption 3**: runtime semantic resolution is unavoidable.

Assumption 1 is necessary because no one can understand the other without sharing basic understanding of the world and some common vocabulary. The base ontology should be general and stable. It can be either defined in some agreed upon ontology specification languages (e.g., Ontolingua [7] and DAML+OIL [5]) or represented in some other forms (e.g., WordNet, a natural language-based taxonomy, as in work in [1]). Assumption 2 allows each agent to develop its own specialized vocabulary. Usually, the agent specific ontologies are changed or updated more frequently than the base ontology. Since these ontologies are defined on top of the base ontologies, they are also called *differentiated ontologies* in the literature [19].

Research work on ontology attempts to provide semantics to information exchanged over the internet [5, 6, 12].The most noticeable recent development in this direction is the *Semantic Web* effort joint launched by W3C [2, 16], the DARPA Agent Markup Language Project [5], and EU's Information Science and Technology Program' [12]. One result from this effort is DAML+OIL specifications, a language for ontology definition, manipulation, and reasoning [5]. Although the technologies developed in this effort are primarily aimed at making web pages understandable by programs, they may serve as a basis (both as language standards and theoretical foundation) for resolving semantic differences between heterogeneous agents. However, additional methodology and mechanisms need to be developed if semantic resolution is to be done dynamically at runtime (through agent interaction). This is the primary objective of our ongoing project on semantic resolution for MAS applications in E-Commerce jointly by The Laboratory for Advanced Information Technology at UMBC and Internet Commerce for Manufacturing at NIST.

2 A Simple E-Commerce Scenario

Consider the following simple E-Commerce scenario of **RFQ** (Request For Quote) involving two agents, the buyer A1 representing a whosaler of computers, and seller A2, representing a computer manufacturer. Both A1 and A2 share a common ontology ONT-0, which gives semantics of some very basic terms (e.g., those describing business transactions such as RFQ and generic names for computer systems and components such as notebooks, CPU, memory, etc.). Each agent has its own specialized ontology (i.e., ONT-1 that gives the semantics or data model of products to order for A1, organized based on the needs or ointended usage of its customers, and ONT-2 that gives data model for the product catalog for A2, based on types of computer systems).

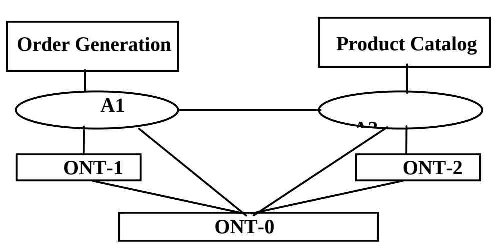


Fig. 1. A simple RFQ scenario involving two agents

Suppose A1 sends an RFQ to A2 for a number of “PC_for_Gamers” product, which is a term defined in ONT-1. Before A2 can determine a quote, it needs to understand what A1 means by this term and if a semantically similar term is in its catalog (as defined in ONTO-2). We use phrase “***Semantic Resolution***” to reference the process of identifying the meaning of terms defined in different ontologies and matching them semantically.

3 Ontology Representation

Although some researchers have used full first-order logic for ontology representation (e.g., Ontolingua [7]), the current trend has been to use DL of different flavors [5, 6, 17 - 19]. DAML+OIL can be seen as a combination of DL and web standards such as RDF, RDF Schema [10], and XML. It provides rich modeling primitives commonly found in frame-based languages such as classes, super-classes, properties, sub-properties, and constraints on cardinalities, values, domains, ranges etc. The language also has a clean and well defined semantics (model-theoretic), similar to the one for DL. DAML+OIL uses XML to encode ontologies. One feature particularly interesting us is the unique namespaces adopted from XML schema, which can be used to reference individual (both base and agent specific) ontologies. We use ns0, ns1, and ns2 as namespaces for the three ontologies ONTO=0, ONTO-1, and ONTO-2, respectively.

The following is an example of XML encoded DAML+OIL definition of a class of “PC_for_Gamers” in ONTO-1. Here symbols starting with “#” are terms defined in the home ontology ONTO_1, whose namespace ns1 is omitted, and prefix symbols “daml” and “rdfs” denote namespaces for DAML and RDF Schema specifications whose URI (e.g., xmlns:daml = ”http://www.daml.org/2001/03/daml+oil#”) are given as part of XML Schema at the beginning of the ontology definition.

In essence, this definition says that the concept or the class of “PC_Fof_Gamer” is a sub-class of “Computers-to-order” in ONTO-1 and sub-class of “Workstations, desktop-computers” defined in ONTO-0, with “good video card”, “good sound card”, and “fast CPU” (the meanings of these terms are defined in ONTO-1, the home ontology).

```
<daml:Class rdf:ID=”PC_For_Gamer “>
   <rdfs:subClassOf rdf:resource=”#Compuetrs-to-order”/>
   <rdfs:subClassOf rdf:resource=”
          ns0: Workstations, desktop-computers “/>
   <rdfs:subClassOf>
          <daml:Restriction>
                <daml:onProperty rdf:resource=”ns0:hasVideoCard”/>
                     <daml:hasValue rdf:resource=”#GoodVideoCard “/>
            </daml:Restriction>
   </rdfs:subClassOf>
      <rdfs:subClassOf>
         <daml:Restriction>
               <daml:onProperty rdf:resource=”ns0:hasSoundcard”/>
               <daml:hasValue rdf:resource=”#GoodSoundcard”/>
         </daml:Restriction>
   </rdfs:subClassOf>
      <rdfs:subClassOf>
         <daml:Restriction>
               <daml:onProperty rdf:resource=”ns0:hasCPU”/>
               <daml:hasValue rdf:resource=”#FastCPU “/>
```

4 Operations for Semantic Resolution

The semantic resolution process consists of two steps, *Semantic Querying* and *Semantic Mapping*, each of which involves its own research issues. We briefly describe each in the following subsections, and address technical issue involved in the subsequent sections.

Semantic Querying. Following the example scenario in the previous section, since A2 only understand ONTO-0 and ONTO-2, it does not understand the term ns1:*PC_for_Gamers* in the RFQ from A1. Similar to the conversation of two strangers in a human society, A2 would ask what does A1 mean by this term via some agent communication language (ACL). We call this process of obtaining the description of a term from a different ontology *Semantic Querying*, and the two agent specific ontologies ONTO-1 and ONTO-2 in our example are called the *source* and *target* ontologies, respectively, with respect to semantic querying. The description of a source term include both slot name and filler name of each slot in its definition in the source ontology. In our example, the first semantic query to A1 gives A2 the following information (with proper namespace designation).

ns1:PC_for_Gamers

List of primitive super-classes

- ns1: Computers-to-order
- ns0:Workstations, desktop-computers

List of properties

- ns0:HasGraphics_card = ns1:GoodGraphicCard
- ns0:HasSound_card = ns1:GoodSoundCard
- ns0:HasCPU= ns1:FastCPU
- ns0:Memory=ns1:BigMemory

Additional queries on ns1 terms in the above description gives

ns1:PC_for_Gamers

List of primitive super-classes

- ns1: Computers-to-order
- ns0:Workstations, desktop-computers
- ns0:Computers

List of properties

- ns0:HasGraphics_card = (ns0:size > 1000)
- ns0:HasSound_card = (ns0:size > 24)
- ns0:HasCPU = (ns0:size > 1000)
- ns0:Memory = (ns0:size > 256).

This can be viewed as an extended normal form of the given ONTO-1 concept with respect to ONTO-0[1].

Semantic Mapping. The extended normal form of ns1:PC_for_Gamers from the semantic querying step provides much information to A2. However, for A2 to truly understand this concept, it needs to map or re-classify this description into one (or more) concept defined in its own ontology ONTO-2. This is accomplished by the *Semantic Mapping* step. Note that due to the structural difference, concepts from different ontologies are likely to match each other only partially in semantic mapping. All partially matched target concepts are considered candidate maps of the source concept. If the best candidate is satisfactory, then a quote is generated by A2 and sent to A1. Otherwise, additional steps of inter-agent interactions may be taken. For example, if the best candidate, although unsatisfactory, is sufficiently better than all others, then its description is sent back to A1 for confirmation. If the first few leading candidates have similar level of satisfaction, then some questions that discriminate some candidates over others will be sent to A1. The details of the algorithms are described in Section 6.

5 Communication Protocol for Semantic Resolution

To support agent communication for both semantic querying and semantic mapping, we need to have 1) an agent communication language (ACL) to encode messages, 2) a content language to encode the content of a message, and 3) a communication protocol that specifies how these messages can be used for meaningful conversations. For reasons including clearly defined semantics and standardization support, we have selected FIPA ACL [9] as the ACL for our project over KQML [10], an ACL. popular in early days of agent research. We choose DAML+OIL as the content language because it is also the language for ontology specification.

The most relevant work on developing agent communication protocol for semantic resolution between two different ontologies can be found in [1]. The

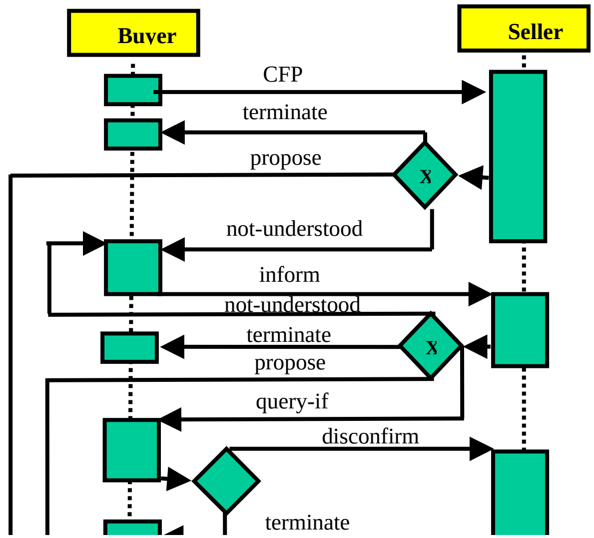
[1] In description logics, a normal (or canonical) form of a concept *C* consists of two lists, a list of all of *C*'s primitive super-classes, and a list of all of *C*'s properties, including those inherited from its super-classes.

protocol, called ONP (Ontology Negotiation Protocol), is an extension of KQML with additional performatives, such as *Request Clarification*, *Clarification*, *Interpretation*, *Confirmation*, etc. Our protocol for ontology-based semantic resolution combines a FIPA-based negotiation protocol for RFQ process in e-commerce we developed earlier [4] and the work in [1]. The design follows FIPA Interaction protocol convention, which requires the definitions of: 1) the acts involved in interaction processes; 2) the roles played by the actors in interaction processes; and 3) the phase transitions of the interaction process. There are two players in our *Semantic Resolution Protocol* (it may be easily extended to involving multiple players), the buyer (A1) and the seller (A2). The buyer plays the role of *the initiator* while the seller the *participant.*

Performatives used in the protocol represent the communicative acts intended by the players. The following FIPA performatives are selected for the protocol.

- **call-for-proposal** (CFP): the action of calling for proposals to perform a given action. This is used for the buyer to ask the seller to propose a quote for a given RFQ.
- **Propose**: the action of submitting a proposal to perform a certain action, given certain preconditions. This is used to turn a proposed quote
- **accept-proposal**: the action of accepting a previously submitted proposal to perform an action
- **reject-proposal**: the action of rejecting a previously submitted proposal to perform an action
- **terminate**: the action to finish the interaction process.
- **inform**: the action of informing that certain propositions are believed to be true.
- **not-understood**: the action of informing the other party that its message was not understood. This is used by the seller to request the buyer to send the description of a term it does not understand in the previous message.
- **query-if**: The action of asking another agent whether or not a given proposition is true. This is used by the seller in semantic mapping to ask the buyer to confirm if a candidate concept is an acceptable match for the given source concept.
- **confirm**: the action of confirming that given propositions are believed to be true. This is used by the buyer to confirm a target concept received in the incoming "query-if" message from the seller.
- **disconfirm**: the action of informing that given propositions are believed to be false

The first 5 are for RFQ, the rest are for semantic querying and mapping. (See [9] for a detailed description of these perfromatives.) The phase transitions in the protocol is given in the following message flow diagram.



6 Algorithms for Semantic Mapping

The primary objective of semantic resolution is to find a concept in the target ontology whose description best matches the description of a given concept defined in the source ontology. Because agent specific ontologies often have different structures and use different concept names, concept matching involving here are seldom exact. Partial matches, which can occur even if a single ontology is involved, become more prevalent when different agent specific ontologies are involved, and the simple techniques used in DL for partial matches (e.g., most gneal subsumees and most specific subsumers) are no longer adequate. Approximate reasoning that at least gives a ranking for all partially matched target concepts is required. Commonly used approximate reasoning techniques include *rough set theory* [18], f*uzzy set theory* [15], and *probabilistic classification* [13, 15]. In many applications, these more formal approaches may not work, either because the assumptions made for them cannot be met or the information (e.g., fuzzy membership functions or statistical information) needed is not available. Heuristic approximation becomes necessary [19].

In this section, we focus on heuristic methods for approximating partial matches. The main algorithm *subsumption*(*A, B, theta*) is an extension of the structural comparison for subsumption operation in DL, and it returns a numeric score theta in [0, 1] for the degree that concept A (or a description) subsumes concept B. In DL, A subsumes B if and only if every object in A is also an object in B. Structural comparison approach [3, 11] works with normal forms of concepts , which include a list of all primitive super-classes *P* and a list of all properties *R* for a concept, and requires that 1) *Pa* is a subset of *Pb*, and 2) constraints on *Rb* is compatible with that of *Ra*. These requirements cannot be established logically if the normal forms of A and B involve terms from different ontologies. This can be seen by comparing the extended normal form of ns1:PC_for_Gamers in Section 4 obtained via semantic querying operation and the extended normal form for ns2:Professional_Use_Desktop given below. Besides ns0 terms, these two normal forms contain ns1 and ns2 terms from ONTO-1 and ONTO-2, respectively.

ns2:Professional_Use_Desktop

List of primitive super-classes

- ns2:Desktop
- ns0:Workstations, desktop-computers
- ns2:Copmuter_Systems
- ns0:Computers

List of properties

- ProductName = “xxx4”
- ProductNumber = “yyy4”
- ns0:HasSound_card = (ns0:size = 24)
- ns0:HasCPU = (ns0:size = 1800)
- ns0:Memory = (ns0:size = 512)
- ns0:Price = (ns0:size = 2300)
- ns2:HasColorMonitor = subproperty(ns0: HasMonitor ns0:size = 19)

One may suggest that we ignore all of these ns1 and ns2 terms and conduct subsumption operation based solely on those ns0 term. However, doing so would overlook the important information on the structural difference, and it is generally believed that if two concepts are far apart in structure, they are less likely to match each other, even if they agree well on terms of the base ontology. In what follows we briefly descrbe the methods to compute measure to compare two concepts’ *P* and *R* lists and the method to combine them into a single score.

Comparing the superclass lists *Pa* and *Pb*. The objective of this comparison is to obtain a measure for the degree that *Pa* is a subset of *Pb*. More than one heuristic rules can be used for approximating set. Here we choose a simple but general one

inclusion_measure(X, Y) = |X n Y| / |X|.

This measure is 1 when X is a subset of Y, 0 if none of the members of X is also a member of Y. One benefit of this heuristic rule is that is can be viewed as the conditional probability *Pr*(x in Y | x in X) when members of X and Y are treated as sample points from the same space. This allows us to generalize the measure with more sophisticated probabilistic computation when the interdependency of these members are known. When logical inconsistency occurs (e.g., any member in *Pa* is known to be disjoin with any member in *Pb*), this measure returns –1. Applying this function to our example of ns1:PC_for_Gamers and ns2:Professional_Use_Desktop, we have the inclusion index of 2/3 because 2 of the 3 member in *P* list of the former are members of *P* list of the latter.

Comparing the property lists *Ra* and *Rb*. This comparison is done in two steps. First, we identify all matching pairs between members in *Ra* and in *Rb*. *Ra_i* in *Ra* matches *Rb_j* in *Rb* if 1) they have the identical property name, including the name space, or 2) *Rb_j* is a sub-property of *Ra_i*. Second, for each matching pair, we compute a compatibility measure of their constraints (e.g., cardinalities and value ranges). If *Rb_j* and *Ra_I* are incompatible (e.g., the logical expressions of their constraints are not satisfiable simultaneously), then A cannot subsume B and the measure = -1, otherwise, the measure is determined by the degree they agree with each other. For example, to compute the compatibility measure for the value range constraints of a pair, we can use the following function

$$\textbf{\textit{compatibility\_measure}} = \begin{cases} -1 & \textbf{\textit{if}} \quad \textbf{\textit{range\_j}} \cap \textbf{\textit{range\_i}} = \emptyset \\ 1 & \textbf{\textit{if}} \quad \textbf{\textit{range\_j}} \subseteq \textbf{\textit{range\_i}} \\ \alpha_{ij} & \textbf{\textit{otherwise}} \end{cases}$$

where α_{ij} is the overlapping ratio between *range_i* and *range_j*, which can be computed by additional rules that handle different types of value ranges. In our running example, all properties, except ns0:HasGoodGraphicCard, in ns1:PC_for_Gamers match with those in ns2:Professional_Use_Desktop, and for each matched pair, the value range in latter is a subset of that of the former. Therefore, each pair has a compatibility measure of 1.

Not every property in either *Ra* or *Rb* can be matched. We will ignore unmatched properties in *Rb* since B as the subsumee is more specific than A, the subsumer, and thus may contain more properties than A. Unmatched properties in *Ra* should in general affect the overall measure. To account for this we modify the compatibility measure with the ratio of the number of unmatched properties in *Ra* over the size of *Ra*. Since only 1 out of 4 properties of the source concept does not have a match, the ratio is ¾, and the modified compatibility measures for the three are ¾.

Combining comparison results. Let $x_1,\ldots,x_n$ be *n* measures of the comparison results (including both inclusion measure for super-class comparison and compatibility measures for each matched property pair), and $w_1,\ldots,w_n$ be their weights or importance. The combined measure is given as

$$\textbf{\textit{theta}} = 1 - \Pi_{i=1}^{n}(1 - x_i \cdot w_i)\,.$$

This formula, known as the Bernoulli law in probability theory, has been widely used in evidential reasoning to combine the influence of individual evidences when the influence of evidences are additive and independent of each other [14]. Applying this formula to our example the combined score for *subsumption*(*ns1:PC_for_Gamers, ns2:Professional_Use_Desktops, theta*) is computed as (assuming all weights = 1)

$$\textbf{\textit{theta}} = 1 - (1 - 2/3)(1 - 3/4)^3 = 0.9947\,.$$

Search for the plausible subsumees. The semantic resolution seeks a most plausible target concept B that either approximately subsumes or is subsumed by A, measured by the heuristic score theta. Finding the most plausible subsumee can be done by a depth-first search plus backtracking or more efficiently by a best-first style search of the target ontology graph. Candidate target concepts are normalized when they are generated during the search. The ranking of candidate target concepts can also be used to support additional process such as *hypothesize-and-test* as mentioned earlier.

7 Conclusions

We believe that the joint Semantic Web effort by W3C and the DAML will have far-reaching influence on the agents R&D and applications. However, the agent research community must keep in mind that the full potential of the agent approach will be delivered only if the agents truly are able to live in open, dynamic, heterogeneous environments. Therefore, the semantic resolution issue needs to be kept at the forefront of the research agenda. Because of the difficult nature of this issue, a solid, sustained research effort will be needed that spans multitude of efforts from software engineering to fundamental research.

The work presented in this paper represents an initial phase of our effort toward a comprehensive solution to the problem of semantic resolution. Many issues, both practical and theoretical, remain to be addressed. To answer some of them, we will to continue our project along the following directions. First, we plan to build a prototype agent system based on the approach outlined in this paper. This system will be used as a testbed to validate the methods we develop and to test various tools and specifications developed by the DAML project. It can also serve as a bridge connecting the research community and the industry by incorporating ontologies of real world enterprises engaged in E-Commerce activities. Secondly, we plan to develop a more formal treatment for approximating semantic mapping with partial matched concepts. One approach we are considering is to incorporate probability theory, in particular the Bayesian belief network [13, 14], into the ontology class hierarchies. Finally, we plan to develop a theoretical model of semantic resolution for differentiated ontologies. This model views the semantic resolution as evidential reasoning, in which the evidences are incrementally accumulated (via semantic querying) and the solution gradually emerges through semantic mapping with each new evidence.

References

1. Bailin, S.C. and Truszkowski,W., Ontology Negotiation between agents supporting intelligent information management, *Workshop on Ontologies in Agent Systems,* 2001.
2. Berners-Lee, T. What the Semantic Web can represent, http://www.w3.org/DesignIssues/RDFnot.html, 1998.

3. Borgida, A., *et al.* CLASSIC: A Structural Data Model for Objects. In *Proceedings of ACM SIGMOD International Conference on Management of Data*, Portland, OR, June 1989, 59-67.
4. Chen, Y., *et al.* A Negotiation-Based Multi-agent System for Supply Chain Management. In *Proc. of The International Conference on Autonomous Agents*, Seattle, May, 1999..
5. DAML home page, http://www.daml.org/.
6. Decker, S. et al, Knowledge representation on the Web. *Proceedings of the 2000 International Workshop on Description Logics*.
7. Farquhar, A., Fikes, R., and Rice, J. The Ontolingua server: a tool for collaborative ontology construction. *Intl. J. Of Human-Computer Studies* **46**(6):707-727, 1997.
8. Finin, T., Labrou, Y., and Mayfield, J. KQML as an Agent Communication Language, in Jeff Bradshaw (ed.), *Software Agents*, MIT Press, Cambridge, MA, 1997.
9. FIPA (The Foundation of Intelligent Physical Agents) home page, http://www.fipa.org
10. Lassila, O. and Swick, R. Resource Description Framework (RDF) Model and Syntax Specification, http://www.w3.org/TR/1999/REC-rdf-syntax-19990222/, Feb, 1999.
11. MacGregor, R.M. The Evolving Technology og Classification-based Knowledge Representation Systems. In *Principles of Semantic Networks: Explorations in the Representation of Knowledge,* J. Sowa (ed.), Morgan Kaufmann, 1991.
12. OIL (Ontology Inference Layer) home page, http://www.ontoknowledge.org/oil/.
13. Pearl, J. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*, Morgan Kaufman, San Mateo, CA, 1988.
14. Peng Y and Reggia J: *Abductive Inference Models for Diagnostic Problem Solving*, Springer-Verlag, New York, NY,1990.
15. Russell, S. and Norvig, P. *Artificial Intelligence, A Modern Approach.* Prentice Hall, Englewood Cliffs, NJ, 1995.
16. Semantic Web home page, http://www.w3.org/2001/sw/
17. Stuckenschmidt, H., Uing OIL for semantic information Integration. *Proceedings of the ECAI workshop on Ontologies and PSMs 2000*
18. Stuckenschmidt,, H. and Visser. U., Semantic translation based on approximate re-classification. *Proceedings of the Workshop "Semantic Approximation, Granularity and Vagueness, KR'00*
19. Weinstein, P. and Birmingham, W.P., Comparing concepts in differentiated ontologies. Proceedings of the Twelfth Workshop on Knowledge Acquisition, Modeling and Management (KAW'99)..