Semantic Resolution for E-Commerce

Yun Peng
Dept. of Comp. Sci. and El. Eng.
University of Maryland Baltimore Co.
Baltimore, MD 21250, USA

ypeng@cs.umbc.edu

Youyong Zou
Dept. of Comp. Sci. and El. Eng.
University of Maryland Baltimore Co.
Baltimore, MD 21250, USA

yzou1@cs.umbc.edu

Xiaocheng Luan
Dept. of Comp. Sci. and El. Eng.
University of Maryland Baltimore Co.
Baltimore, MD 21250, USA

xluan1@cs.umbc.edu

Nenad Ivezic
Nat'l Institute of Standards and Tech.
100 Bureau Drive
Gaithersburg, MD 20899, USA

nivezic@nist.gov

Michael Gruninger
Nat'l Institute of Standards and Tech.
100 Bureau Drive
Gaithersburg, MD 20899, USA

gruning@nist.gov

Albert Jones
Nat'l Institute of Standards and Tech.
100 Bureau Drive
Gaithersburg, MD 20899, USA

ajones@nist.gov

ABSTRACT
We describe a research project on resolving semantic differences for multi-agent systems (MAS) in electronic commerce. The approach can be characterized as follows: (1) agents in a MAS may have their own specific ontologies defined on top of a shared base ontology; (2) concepts in these ontologies are represented as frame-like structures based on DAML+OIL language; (3) the semantic differences between agents are resolved at runtime through inter-agent communication; and (4) the resolution is viewed as an abductive inference process, and thus necessarily involves approximate reasoning.

Categories and Subject Descriptors
I.2.11 [**Artificial Intelligence**]: Distributed Artificial Intelligence – *multiagent systems*.

General Terms
Algorithms, Design, Theory

Keywords
Semantic resolution, e-Commerce, multi-agent system

1. INTRODUCTION
Understanding the meaning of messages exchanged between software agents has long been recognized as a key challenge to interoperable multi-agent systems (MAS). Forcing all agents to use a common vocabulary defined in shared ontologies is an oversimplified solution. This is the case for agent applications in e-Commerce which (1) is a huge, *open* marketplace accommodating many companies capable of entering and leaving

the market freely; (2) involves *dynamic* partnerships formed and dissolved easily and frequently; and (3) contains *heterogeneous* representations of agents for different enterprises.

Semantic differences should be resolved when they arise during agent interaction. These points are captured by the following assumptions: (1) interacting agents share base ontologies; (2) agents may use different ontologies defined on top of the base ontologies; and (3) runtime semantic resolution is unavoidable [1].

Research work on ontology engineering attempts, in part, to provide semantics for information exchanged over the Internet. One result is the set of DAML+OIL specifications, a language for ontology definition, manipulation, and reasoning [2]. These specifications may form a basis for resolving semantic differences between heterogeneous agents. However, additional methodology needs to be developed if semantic resolution is to be done at runtime. This is the primary objective of our investigation.

2. APPROACH
2.1 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 wholesaler of computers and seller A2 representing a computer manufacturer. Both A1 and A2 share a common ontology ONT-0, which gives semantics of some basic terms that describe business transactions such as RFQ and generic names for computer systems and components such as notebooks, CPU, and memory. Each agent has its own specialized ontology. ONT-1 defines semantics of products to order for A1, organized to meet the usage of its customers. ONT-2 defines items in the product catalog for A2, based on types of manufactured computer systems.

2.2 Ontology Representation
Figure 1 gives an example of an XML-encoded DAML+OIL definition of a class of "PC_for_Gamers" in ONT-1. Here symbols starting with "#" are terms defined in the home ontology ONT-1, while ns0 indicates ONT-0 terms. Prefix symbols "daml" and "rdfs" denote namespaces for DAML and RDF Schema.
```
<daml:Class rdf:ID="PC_for_Gamers ">
```

```
    <rdfs:subClassOf rdf:resource=
      "#Computers-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 "/>
      </daml:Restriction>
    </rdfs:subClassOf>
  </daml:Class>
```

Figure 1. An example of a class definition in DAML+OIL

The definition says that the concept of "PC_for_Gamers" is a sub-class of "Computers-to-order" in ONT-1 and sub-class of "Workstations, desktop-computers" defined in ONT-0, with "good video card", "good sound card", and "fast CPU" -- the meanings of these terms are defined in ONT-1 ontology.

2.3 Operations for Semantic Resolution

The semantic resolution process we investigate consists of two steps, *Semantic Querying* and *Semantic Mapping,* each of which involves its own research issues.

Semantic Querying. Since A2 only understands ONT-0 and ONT-2, it would not understand term ns1:*PC_for_Gamers* in an RFQ from A1. A2 would ask what A1 means by this term via some agent communication language. We call this process *Semantic Querying*. The description of a source term includes both slot name and filler name of each slot in its definition in the source ontology. In our example, the answer to the first semantic query to A1 gives A2 the following information.

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).

Semantic Mapping. The above 'extended normal form' of ns1:PC_for_Gamers from the previous step provides much information to A2. However, in this step, A2 needs to re-classify this description in its own ontology ONT-2.

Due to the structural differences, concepts from different ontologies are likely to match only partially. The simple techniques used in description logics (DL) for partial matches (e.g., most general 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 (e.g., *rough set theory*, *fuzzy set theory*, and *probabilistic classification*). In many applications, more formal approaches may not work, either because the assumptions made for them cannot be met or the information needed is not available. Heuristic approximation becomes necessary.

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. It returns a numeric score, theta, in [0, 1] that quantifies the degree that concept A subsumes concept B.

3. CONCLUSIONS

The work presented in this paper is our initial effort toward a comprehensive solution to the problem of semantic resolution. Our future plans include building 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. It will connect the research community and the industry by incorporating ontologies of real-world enterprises engaged in e-Commerce activities.

4. DISCLAIMER

Certain commercial software products are identified in this paper. These products were used only for demonstration purposes. This use does not imply approval or endorsement by NIST, nor does it imply that these products are the best available for the purpose.

5. REFERENCES

[1] Bailin, S.C. and Truszkowski,W., Ontology Negotiation between agents supporting intelligent information management, *Workshop on Ontologies in Agent Systems* '01.

[2] DAML Website: www.daml.org