Int. J. Contemp. Math. Sci., Vol. 1, 2006, no. 6, 255 - 266

Carrying Updated DAML Ontology to Improve Agent Communication

Chin-Swern Huang and Jason Jen-Yen Chen
Dept. of Computer Science and Information Engineering
National Central University
Chung-Li, Taiwan

Abstract

This paper presents a new approach to improve agent communication. DARPA Agent Markup Language (DAML) ontology is carried and updated by agent, mobile or stationary. Basic ontology reflects main architecture of DAML. It is updated regularly. On the other hand, extended ontology refers to the domain knowledge constructed using basic ontology. If needed, it is updated when a mobile agent arrives at a website. Its benefits appear to be: 1) improved agent communication in terms of speed, precision, and stability; 2) minimal ontology update; 3) ontology recovery in case of malicious destruction; and 4) precise web page access. An advanced traveler information system (ATIS) is included to demonstrate its feasibility.

Keywords: ontology, agent, DAML.

1. Introduction

Agent communication is currently an active agent research area. Every agent system has its specific agent communication protocol. For example, MiLog [5] uses its own communication predicates to realize communication. Bee-gent [12] and ITTalks [9] use, respectively, KQML (Knowledge Query and Manipulation Language) and Foundation for Intelligent Physical Agents (FIPA) [2] Agent Communication Language (ACL) to communicate among agents.

Like ITTalks, this paper adopts FIPA/ACL. Further, the ontology is built using DARPA Agent Markup Language (DAML) [7] [8]. Most of all, mobile agent carries basic ontology (to be explained shortly) and extended ontology (downloads or updates from another agent system when it travels there). Then, the agent uses the extended ontology to construct content of ACL that should be understood by agents of another agent system. This is just like how human communicates with persons from another community.

Moreover, agents can precisely access the web page encoded in DAML. This realizes the ideal of semantic web [13].

Section 2 describes related work such as Milog, Bee-gent, and ITTalks, including their communication protocols. Section 3 depicts architecture of this implementation, including DAML ontology. Section 4 gives an intelligent transportation system (ITS) example to demonstrate the feasibility of this architecture. Finally, Section 5 draws conclusions and looks into the future.

2. Related Works

This section compares three agent-systems, namely, MiLog, Bee-gent, and ITTalks with regard to ACL format and communication scheme, respectively.

1) ACL format:

1. MiLog [5] uses its own ACL format. However, with the growing adoption of FIPA/ACL as the standard, MiLog will have a hard time integrating with other agent systems. In other words, this restricts MiLog from further development and wide acceptance.

2. Bee-gent [12] uses KQML format with XML content. KQML was an ACL used in early days. With the growing acceptance of FIPA as the standard these days, KQML will be difficult to integrate with other agent systems. In addition, XML is less capable in describing resources than DAML. This makes agents difficult to interpret XML content in terms of lower speed and poorer quality.

3. ITTalk [4] uses FIPA format with DAML content. Its ACL is similar to that presented in this paper. However, they differ greatly in

communication mechanism that will be covered next.

2) Communication mechanism:

1. MiLog [5] uses fixed communication predicates make agent communication less flexible that has restricted agent communication. This is just like people can only use fixed vocabulary to communicate with each other. This will result in poor communication. Contrarily, extended ontology is used in this paper to construct content that will make agent more human-like.

The example below shows how MiLog agents communicate. In MiLog, an agent (agentA) want to know another agent (agentB) how it feels now, agentA will use the predicate below to query:

query (agentB, feeling (Now))

Because that the number of predicates (such as "query"), actions (such as "feeling"), and arguments (such as "Now") are fixed, this communication mechanism is rather inflexible.

2. Bee-gent [12] uses mediation agent (a mobile agent) to assist stationary agents in communication. It uses XML to construct the ontology. However, XML ontology in different agent systems are difficult to be understood, because different basic tags (see "basic ontology" to be described shortly) may be used in different ontology. More importantly, it has no mechanism to carry and update ontology that makes it impossible to resolve the problem with different ontology.

3. Every agent in ITTalks [9] administrates its own ontology, but it doesn't utilize the mechanism that carries and updates ontology to assist agent communication and mobile agent. Thus, the agents can only communicate with very limited constructs that are available to all the agents. Moreover, without mobile agent, network congestion will seriously affect performance.

3. Carrying updated DAML ontology

The so-called “basic ontology” [1] reflects main architecture of DAML. It is based on Resource Description Framework (RDF) and Resource Description Framework Schema (RDFS), and it benefits from SHOE [10] and OIL [6]. It is a group of some defined tags called “basic tags”. And “extended ontology” is the ontology that agent system uses basic ontology to further construct according to its needs.

Every agent, whether mobile or stationary, must carry basic ontology with regular updates. Fig. 1 reveals the content of basic ontology. The first line is the comment showing that the basic ontology is version 1.7. Lines 3-6 are XML namespaces (xmlns). Beneath line 8 are some more detailed description, and the definition of elements, attributes and properties.

```
1     <!-- $Revision: 1.7 $ of $Date: 2001/06/06 01:38:21 $. -->
2
3     <rdf:RDF
4        xmlns:rdf ="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
5                              . . .
6     >
7
8    <rdf:Description rdf:about="">
9    <versionInfo>$Id: daml+oil.daml,v 1.7 2001/06/06 01:38:21 mdean Exp
10    $</versionInfo>
11   <imports rdf:resource="http://www.w3.org/2000/01/rdf-schema"/>
12    </rdf:Description>
13
14    <!-- (meta) classes of "object" and datatype classes    -->
15
16    <rdfs:Class rdf:ID="Class">
17       <rdfs:label>Class</rdfs:label>
18       <rdfs:comment>
19          The class of all "object" classes
20       </rdfs:comment>
21       <rdfs:subClassOf
22    rdf:resource="http://www.w3.org/2000/01/rdf-schema#Class"/>
23    </rdfs:Class>
24                             . . .
25    </rdf:RDF>
```

Fig. 1. Basic ontology

There are two ways to carry updated ontology:

1) Agents, stationary or mobile, will regularly check agent version on the Agent update site for new agent version with updated basic ontology. This is similar to the update function in Windows System. If the website has the new agent version, the agents will download it to update themselves.

2) After mobile agent travels to another website, it will check the extended ontology on the website. If it is different, the agent may or may not discard, based on user setting, original extended ontology and download it.

Extended ontology refers to the domain knowledge constructed using basic ontology. Fig. 2 reveals an extended ontology. Lines 1-12 are some description and namespaces. Most importantly, domain knowledge in agent system is shown below line 13.

After a mobile agent has the same extended ontology as that of a stationary agent or a DAML-enabled Web, it can construct DAML-based ACL to directly communicate with the stationary agent and, furthermore, to access the information and services on the DAML-enabled Web [3].

Fig. 3 shows the content constructed using “NCURoom” extended ontology. Based on it, agent precisely understands all the information of 0B-23 classroom that it wants to find and precisely access that information.

Figures 4-5 reveals the whole agent communication scheme. Every agent will regularly check agent version on the Agent update site to update itself (see Fig. 4). As shown in Fig. 5, on every web service site, there is a stationary agent (ordinary agent or a wrapper agent.) Or, the website may be a Middle Agent Site or a DAML-enabled website. When a mobile agent representing a user travels on the network in order to access information or carry out some actions, it will carry basic ontology to travel to the destination and download the extended ontology on the website. Then, the mobile agent can communicate with the stationary agent or precisely access information on the DAML-enabled website.

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
  <!DOCTYPE uridef (View Source for full doctype...)>
<!--
This document uses entity types as a shorthand for URIs.
Download the source for a version with unexpanded entities.
  -->
<rdf:RDF xmlns:rdf=http://www.w3.org/1999/02/22-rdf-syntax-ns#
. . .
>
  <daml:Ontology about="">
. . .

  </daml:Ontology>
  <rdfs:Class rdf:ID="ClassRoom">
   <rdfs:label> ClassRoom </rdfs:label>
   <rdfs:subClassOf
rdf:resource="http://www.daml.org/2001/03/daml+oil#Thing" />
   </rdfs:Class>
. . .

</rdf:RDF>
```

Fig. 2. Extended ontology

```
<rdf:RDF
xmlns:NCURoom = "http://www.csie.ncu.edu.tw/ontologies/ClassRoom#">
< NCURoom:ClassRoom>
< NCURoom:firstNum>0B</ NCURoom:firstNum>
< NCURoom:lastNum>23</ NCURoom:lastNum>
< NCURoom:RoomName>Computer ClassRoom</ NCURoom:RoomName>
</ NCURoom:ClassRoom > ....
```

Fig. 3 DAML-based ACL

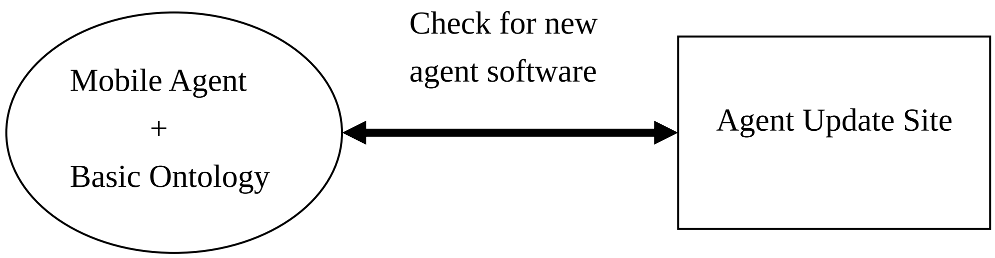


Fig. 4. Mobile Agent (or stationary agent) will regularly check agent version on the Agent update site for new agent version with updated basic ontology

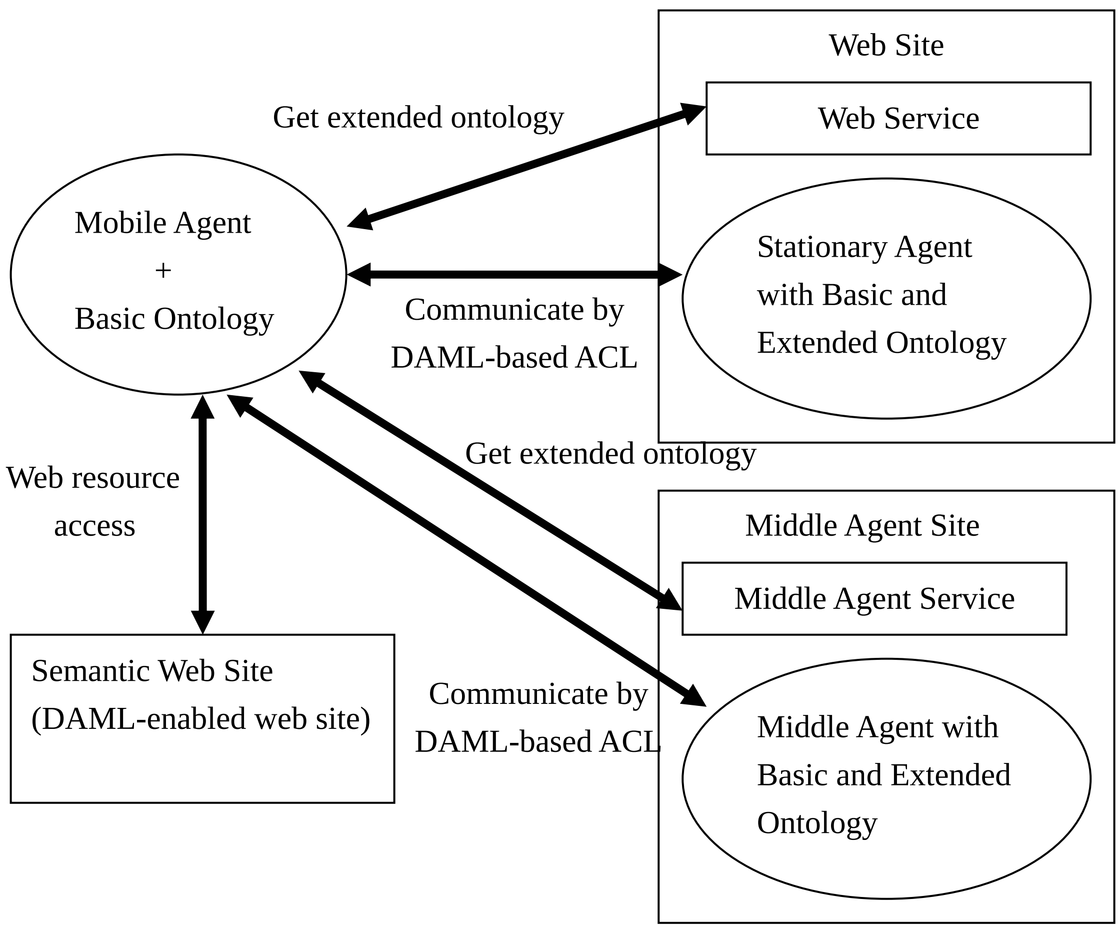


Fig. 5 (above) Mobile Agent will download the extended ontology on the website Then, the mobile agent can communicate with the stationary agent or precisely access information on the DAML-enabled website.

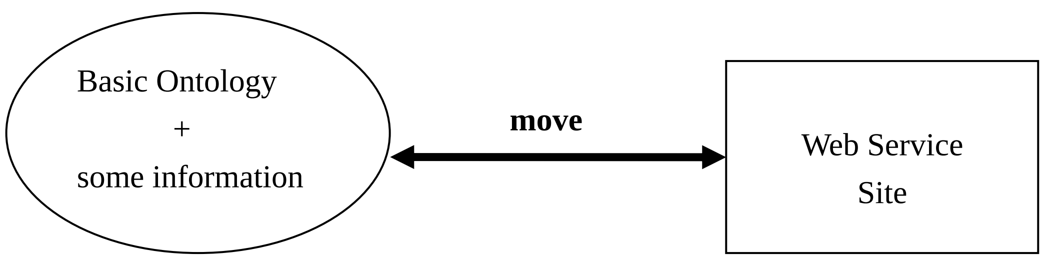
4. ATIS Example

After depicting the agent communication protocol of this system, an example of ATIS (Advanced Traveler Information Systems) will be given next.

The services spread on the Hsin-chu Science Park are web services and middle agent services. Each stationary agent on the service site represents the service and owns both basic ontology and extended ontology.

Fig. 6 reveals the scenario. When a user plans to travel to the Science Park, he/she only needs to send personal mobile agent carrying basic ontology and some necessary information to internet in advance (see step1 in Fig. 6).The agent will travel to the web service site and query the stationary agent on it about which route is faster, based on the carried information. Of course, interactions between the agents are the mobile agent first updates extended ontology, and then uses it to construct the content of ACL. Having done this, the mobile agent is able to communicate with the stationary agent (see step2 in Fig. 6).

Step1



Step2

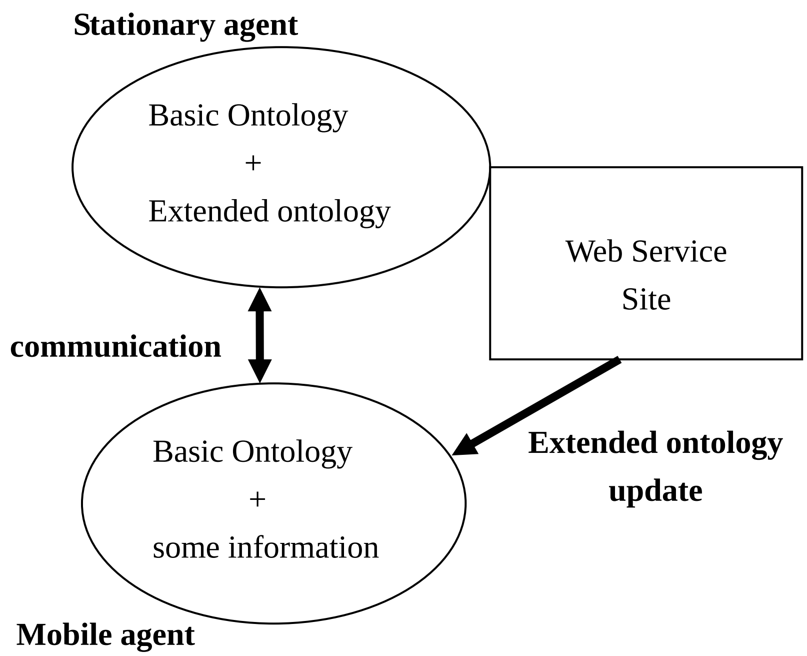


Fig. 6: Movement and communication of the agents

If the mobile agent needs some web service (such as room reservation), but does not know where to request, it can query the middle agent to find the service. The communication between the agents would be rather fast, as it does not require downloading ontology through network. Therefore, quick processing can be expected.

If the web service site has no relevant information, the mobile agent will travel to another site. It only needs to check whether or not the extended ontology on the site is the same as its ontology. If it is, the mobile agent can directly communicate with the stationary agent. If it is not, the agent may or may not discard, based on user setting, original extended ontology and download the extended ontology on the site. Finally, after getting the needed information, the mobile agent returns and reports back to user. Notice that agents will not carry out correct behaviors even after ontology update – unless programmers design agents to do so.

5. Conclusions

There appear to be four benefits of using this approach as below:

1. Improved agent communication:

   First, the mobile agent can interact with stationary agent without network. And, because of carrying basic ontology and extended ontology, the agent can handle ACL message without downloading ontology. This makes fast processing. Second, by using the specifics of DAML, the agent can precisely construct ontology. Further, it can construct precise content any time by using carried ontology. This ensures precise communication. Third, in case that the internet is down, agent can still works. This makes stable communication.

2. Minimal ontology update:

   Supposing that mobile agent travels to another website, after it downloaded the extended ontology on the website, it only needs to check whether or not the extended ontology on another web site is the same as its extended ontology. If they are the same, it can directly communicate with the stationary agent on the website. That means the extended ontology does not have to be downloaded every time. This will make minimal download. In a sense, this is human-like, as human “download” (learn) only when he/she travels to unfamiliar communities.

3. Secure ontology recovery:

   If the ontology on some website is destroyed maliciously, the mobile agent can notify the stationary agent and help to recover the ontology on the website after arriving there. This will protect agent communication from the impediment.

4. Precise web page access：

   Agent can precisely access web page on the DAML-enabled websites. It only needs to check whether or not the ontology that is used to markup the web page is the same as the ontology it carries. If they are the same, the agent can precisely access the web page.

At last, let us look into the future. As ontology is becoming popular and its development is easier than before – thanks to the emergence of DAML, it is expected that widely-accepted ontology will be available in the near future. For example, the DAML-based Web Service Ontology (DAML-S) shows this trend. With different versions of ontology, DAML-S continuously adds new web service tags to it for people to better describe web services. Consequently, agents can find the web services more precisely and easily.

This is like human carrying fixed vocabulary. As time passes by, some new vocabulary is added to it. Similarly things happened to ontology. By using these ontologies, agents have a set of communication vocabulary to follow. Hence agents can find the web services they want quickly and correctly.

Acknowledgments

This work was supported by the National Science Council in Taiwan under Grant NSC93-2213-E-008-036, the Ministry of Economic Affairs in Taiwan under Grant 93-EC-17-A-02-S1-029, and the Ministry of Education in Taiwan under Grant EX91-E-FA06-4-4.

REFERENCES

[1] Basic ontology, http://www.daml.org/2001/03/daml+oil#

[2] Foundation for Intelligent Physical Agents.(2002, December). [online], FIPA Agent Communication Language Specification. Available: http://www.fipa.org/repository/aclspecs.html

[3] G. Denker, J.R. Hobbs, D. Martin, S. Narayana and W. Waldinger, “Accessing Information and Services on the DAML-Enabled Web,” Proc. of The Second International Workshop on the Semantic Web - SemWeb'2001, Workshop at WWW10, May 2001.

[4] Natalya Fridman Noy and Deborah L. McGuinness, “Ontology Development 101: A Guide to Creating Your First Ontology,” Stanford Knowledge Systems Laboratory Technical Report KSL-01-05 and Stanford Medical Informatics Technical Report SMI-2001-0880, March 2001.
[5] N. Fukuta, T. Ito, and T. Shintani, MiLog: A Mobile Agent Framework for Implementing Intelligent Information Agents with Logic Programming, Proc. of the First Pacific Rim International Workshop on Intelligent Information Agents (PRIIA'2000) , pp.113-123, 2000.
[6] Ontology Inference Layer http://www.ontoknowledge.org/oil/
[7] Roxane Ouellet, and Uche Ogbuji, Introduction to DAML: Part I, January 30, 2002.
[8] Roxane Ouellet, and Uche Ogbuji, Introduction to DAML: Part II, March 13, 2002.
[9] R. Scott Cost, Tim Finin, Anupam Joshi, ITTALKS: A Case Study in the Semantic Web and DAML, IEEE Intelligent Systems, 17(1):40–47, January/February 2001.
[10] Simple Html Ontology Extensions http://www.cs.umd.edu/projects/plus/SHOE
[11] The DAML Services Coalition(2002). [online] DAML-S: Semantic Markup for Web Services. Available: http://www.daml.org/services/
[12] T. Kawamura, T. Hasegawa, A. Ohsuga, S. Honiden, Bee-gent: Bonding and Encapsulation Enhancement Agent Framework for Development of Distributed Systems, Proceedings of the 6th Asia-Pacific Software Engineering Conference (APSEC 99), pp. 260-267, 1999.
[13] World Wide Web Consortium (2001, May), [online], Semantic Web. Available: http://www.w3.org/2001/sw/

Received: October 11, 2005