Agent Communication In DAML World [1]

Youyong Zou, Tim Finin, Yun Peng, Anupam Joshi, Scott Cost

Computer Science and Electrical Engineering
University of Maryland Baltimore County
{yzou1,finin,peng,joshi, cost}@cs.umbc.edu

Abstract. Darpa Agent Markup Language (DAML) [7] is the newest effort for Semantic Web [5]. It can be used to create ontologies and markup information resource like web pages. The information resource can be read by human and understood by agent programs. We believed DAML could be used to markup agent communication content and promote knowledge sharing and exchanging between agents. This paper also suggested an alternative model to connect web and agent together. We defined the necessary ontologies for agent communication in DAML language and described the agent communication scenario occurred in the ITTalks Project.

Keyword. Agent Communication Language, Ontology, Semantic Web, Software Agents, DAML

1 Introduction

Semantic Web activities from W3C and other groups like On-To-Knowledge, Darpa, defined specifications and technologies to support knowledge sharing across applications. DAML is the joint effort from US DAML group and Europe Semantic Web Technologies, supported by US Darpa. DAML Language is based on RDF, RDF Schema and benefit from SHOE [10] and OIL [11]. The goal is to markup and embed ontology into the massive web pages so that they are not only viewable by humans, also understandable by programs.

Agent Communication Language (ACL) provides agents with a means of exchanging information and knowledge. Knowledge Query and Manipulation Language (KQML) and the Foundation for Intelligent and Physical Agents (FIPA) ACL are two most widely used ACLs. The work described in this paper was based on FIPA ACL. The FIPA ACL [2, 3] specification consists a set of message types and the description of their effects to the sender and receiver agents. It can be used build high-level interaction protocols, such as contract net and auctions.

FIPA ACL RDF Content Language [1] and Agent toolkit Jade [8] defined ontologies for ACL message content and modeled ACL message in RDF language. As the data model intended for describing metadata, RDF is not strong enough to represent the rich forms of content. Research works in [6] suggested building an Abstract On-

tology Representation(AOR) for agent language (include FIPA-SL, FIPA-KIF and FIPA-ACL) using DAML language, and used the AOR to support multilingual agents. The using of SL and KIF improved the opinion of semantic. However, it is difficulty to efficiently processed and conveyed multiple languages. Also, SL and KIF did not fit well with context of XML-based web. This paper suggested using DAML to encode ACL message. We defined the ontology of objects, propositions and actions in DAML language. Agents communicated with each other by exchanging DAML-encoded documents. Compared with RDF and XML, DAML can express richer meaning and support knowledge sharing among agents using different ontologies. Compared with SL and KIF, DAML's style is more readable and easier for agent programs to parse and understand.

The next section revealed the model of agents working in semantic web. Section 3 introduced the agent communication scenario in ITTalks[2] project. The design of encoding ACL in DAML language was presented in Section 4. Section 5 concluded our work and pointed out future works.

2   Agent that speaks for web pages

The semantic web [5] is a vision in which web pages are augmented with information and data that expressed in a way that facilitates its understanding by machines. The current human-centered web is mostly encoded in HTML, which mainly focuses on how text and images are rendered for human viewing. Over the past few years we have seen a rapid increase in the use of XML, which is intended primarily for machine processing, as an alternative encoding. The machine that processed XML documents could be the end consumers of the information or could transform the information into a form appropriate for human understanding (e.g., HTML, graphics, synthesized speech). As a representation language, XML provided essentially a mechanism to declare and use simple data structures and thus leaved much to be desired as a language of expressing complex knowledge. The enhancements to basic XML, such as XML Scheme, addressed some of the shortcomings, but still did not result in an adequate language for representing and reasoning about the kind of knowledge essential to realizing the semantic web vision. RDF (Resource Description Framework) and RDFS (RDF Schema) attempted to address these deficiencies by building on the top of XML. They provided representation frameworks that were roughly the equivalent to semantic networks in the case of RDF and very simple frame languages in the case of RDFS. However, RDFS was still quite limited as a knowledge representation language, lacking support for variables, general quantification, rules.

DAML is the attempt to build on XML, RDF and RDFS and produces a language that is well suited for building the semantic web. The goal of the DAML program [7], which began in August 2000, is to develop a universal semantic web

---

[2] http://www.ittalks.org

markup language that is sufficiently rich to support intelligent agents and other applications. DAML can dramatically improve traditional ad hoc information retrieval because its semantic improves the quality of retrieval results. Also, DAML allows the intelligent agents to retrieve and manipulate the information on the semantic web.

The exact role of agent and its relationship to the knowledge encoded in documents on the semantic web is one part of the semantic web vision that has not yet been fully articulated. Most works to date had followed one of two models described below, each of which had serious shortcomings. The semantic web assumed that information and knowledge were encoded in a semantic rich web language (e.g., DAML, OIL and RDF) and made available in the form of web pages. The information may ultimately reside in databases, knowledge bases, but it must be accessible in the form of documents partially or completely marked up in a semantic web language.

Agents using semantic information used two models. The active model used by SHOE[10] system assumed that one knowledge acquiring agent existed which was responsible for searching the web for possible semantic marked web pages of interest. All found pages were processed and knowledge inside was loaded into the SHOE system's knowledge base (KB). Subsequently, the KB could answer other agent's questions about the information on the pages. The passive model was that the agent with inference ability located the related pages of interest at runtime whenever a question arose. The desired knowledge was extracted from marked pages, loaded into inference engine. Hence the question was answered by inference engine.

The two models both suggested that one agent in the system carried all the responsibility of finding and understanding the massive semantic information of the entire web. The website may defined and used its own ontologies that agent had no prior knowledge. The rules and policies used inside the website may be complex and closed to outside agents for security reason.

We were pursuing an alternative distributed model in which the web page marked up in semantic web language would appointed an existing agent as service agent. Any question related to the page could be answered by this agent. The web site supporting this model would provider both web based content for human and agent based services for other agents.

The benefits of the distributed model included:
- Distributed agent environment fit massive web environment. There is no centralized agent that has to search all web pages and understand every ontology;
- The best agent to ask question was always present. This model didn't depend on web service discovery or agent services search.
- Ontologies and rules were stored in local and accessed only by local agent. We could define personalized ontologies and rules. It would help resolving the problem of security and trust.

3 Agent Communication Scenario

This section talked about the agent communication scenario occurred in the ITTalks project (http://www.ITTalks.org). ITTalks was designed by UMBC as part of the DAML Project. It provided users with numerous details about the IT events, including location, speaker, hosting organization, and topic. Unlike other event web sites, ITTalks employed DAML for knowledge base representation, reasoning, and agent communication. The use of DAML enabled more sophisticated functions that would otherwise be unavailable. For example, a simple representation scheme might be able to provide the user with talks based on interest, time and location. When both IT-Talks and the user agreed on a common ontologies, the ITTalks web portal was able to perform further filtering based on sophisticated inference. In addition to enhancing knowledge representation and reasoning, DAML is used for all communication, including agent messages and queries. Moreover, ITTalks offered the capability for each user to use his/her personal agent to communicate with ITTalks on his/her behalf and provided a higher level of service

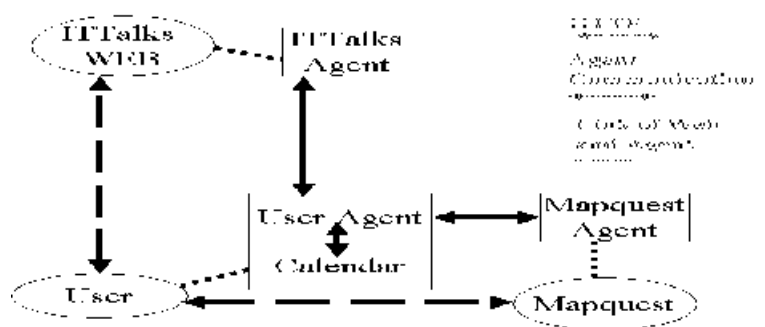Consider the following scenario (Figure 1) which include three agents.

Figure 1: ITTalks Scenario

User Jim already registered with ITTalks, and has left instructions with the system to be notified of the occurrence of certain types of talks. The editor of ITTalks added a new IT talk into ITTalks web site. Based on information in Jim's preferences, IT-Talks agent thought Jim would be interested in this talk and decided to notify Jim's User Agent.

First task of the ITTalks agent is to find Jim's user agent. This was done by searching Jim's DAML-encoded personal profile, which was submitted when Jim registered at ITTalks. ITTalks agent sended out the new talk announce to Jim's User Agent and asked whether he would join or not. The announce was encoded in DAML language. The query ACL message was delivered to the user agent using agent-based messaging system. Upon receiving the message, Jim's User Agent parsed the DAML content and loaded into an inference engine. User Agent consulted with Jim's Calendar agent to determine his availability, and contacted the MapQuest agent to find out

the driving distance from Jim's predicted location at the time of the talk. There could be more sophisticated interactions. For example, Calendar agent and User agents may decide to alter Jim's calendar to resolve the schedule conflict, contact the other individual's User agent for advisory. After all, the User Agent made decision and sent the notification message back to the ITTalks agent indicating whether Jim plan to attend. The ITTalks agent made the appropriate adjustments to event attendant at the ITTalks site.

4 Approach

To accomplish the ITTalks scenario, the DAML ontology for agent communication was needed. Physically, the FIPA ACL message consists of three layers: Content, ACL and Agent Message Transport.

4.1 DAML encoded ACL Message Content

FIPA ACL were designed to use multiple content languages [3]. This had generally been seen as a good design choice in that it provided agents with the flexibility to choose a content language best suited to its domain of discourse, so long as it is one that is mutually understood by its conversational partners. FIPA defined a mechanism by which one can offer a particular content language to the FIPA community for inclusion in the FIPA Content Language Library (CLL) [3]. Inclusion in the library means that certain minimum requirements were met and that there was a specification document for the canonical version of the language to which implementers could refer.

We designed the preliminary version of a specification of DAML ontology as a FIPA compliant content language. To be fully general and support all of the FIPA communicative acts, our ACL DAML Content Language included:

- Objects: RDF Content Language [1] assumed that both an ACL object and an RDF resource were defined as descriptions of a certain identifiable entity. RDF resource identifiers and references could be used as ACL object identifiers and references. Same idea could be used in RDF-based DAML. Here, we used resource as ACL object.
- Propositions: statements expressing that some sentences is true or false
- Actions: express an activity that carried out by an object. It included three properties: the act identifies the operative part of the action; the actor identifies the entity that performs the act; the argument identifies the entity that used by the actor to perform the act.
- Rule and Query: For the rule, the if statement included Prem part and Conc part. For the query, the statement included question part and result part.

4.2 DAML Encoded ACL

ACL described the conversation part of the ACL. In the ontology for FIPA ACL, we treated performative as DAML Class and treated attribute-value pairs as property of the performative Class. Following is a example of inform message. It included agent identifier of sender and receiver, ontology used and content language.

```
<acldaml: inform>
        <acldaml: sender>
    <fipa:agent-identifier>
                <fipa:name>sender@bar.com</fipa :name>
                <fipa:addresses>

            <fipa:url>http://bar.com/acc</fipa:url>
                </fipa:addresses>
                </fipa:agent-identifier>
        </acldaml: sender>
        <acldaml: receiver> ……</acldaml: receiver>
                <acldaml: language value =  FIPA-DAML  />
        <acldaml:content-length value =  12 />
            <acldaml:ontology value =ITTalks />
        </acldaml:inform >
```

4.3 DAML Encoded Agent Message Transport

The ontology of Agent Message Transport defined Message Transport Envelope. FIPA supported HTTP, IIOP, WAP as transport protocol. Following was an example of message using HTTP protocol. It included the agent identifier of sender and receiver, ACL representation language and information about already received message.

```
<fipa :envelope>
<amt:params index="1">
  <amt:to>
   <fipa:agent-identifier>
    <fipa:name>user@foo.com</fipa :name>
    <fipa:addresses>
     <fipa:url>http://foo.com/user </fipa:url>
    </fipa:addresses>
   </fipa:agent-identifier>
  </amt:to>
  <amt:from> ……   </amt:from>
 <amt:acl-representation>fipa.acl.daml</amt:acl-representation>
 <amt:payload-encoding>US-ASCII</amt:payload-encoding>
  <amt:received >
        <amt:received-by value="http://foo.com/user" />
        <amt:received-id value="123456789" />
  </amt:received>
 </amp:params>
</fipa:envelope>
```

4.4 **Parsing of DAML Message**

Figure 2 showed how the DAML file parsing was done in our system. We used XSB system [9] as the inference engine and defined XSB rules for RDF, RDF Schema and DAML. The agents could submit DAML file, which would be parsed into triples and asserted into XSB inference engine. The facts in the XSB can also be retracted when expired. The XSB system supported query of the facts through a frame based language defined in our system. The user agent could own more complex rules. The XSB Engine determined the agent's behavior based on default rules and user agent's own rules.
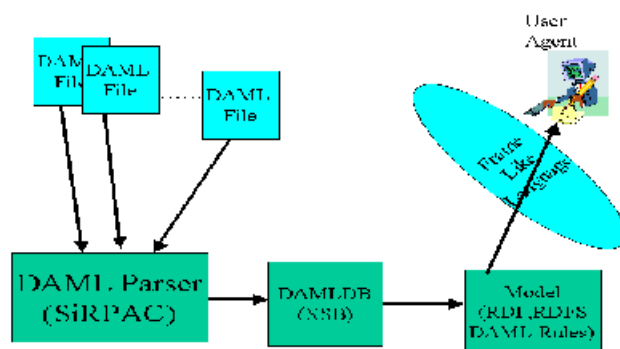

Figure 2. DAML file parsing

5 Conclusion

This paper talked about the agent communication scenario utilizing DAML language, defined ACL message in DAML Language and implemented the reasoning engine for the purpose of parsing DAML messages. To define Agent Message Transport ontology using DAML language and support popular FIPA platforms are one of the future works. The popular FIPA platforms like Jade and FIPA-OS support HTTP, IIOP, WAP as message transport protocol. SOAP and Message Queue may become transport protocol for future agent systems. In this paper, we defined simple rule and query using DAML language. In the future, we will consider enrich this part by using new development like ruleML [4], DQL. Ontology sharing and resolving semantic differences between heterogeneous agents could also be another future work. Forcing all agents to use a common vocabulary defined in one or more shared ontology is an oversimplified solution especially when these agents are designed and deployed independently of each other.

There is no doubt DAML, as the name "Agent Markup Language" suggested, will be an important actor in the agent world.

References

1. FIPA Report XC00011B: FIPA RDF Content language, Foundation for Intelligent Physical Agents (2001).
2. FIPA Report XC00008D: FIPA SL content language, Foundation for Intelligent Physical Agents (2001).
3. FIPA Report XC00009A: FIPA CCL Content Language, Foundation for Intelligent Physical Agents (2001).
4. Harold Boley, Said Tabet, and Gerd Wagner: Design Rationale of RuleML: A Markup Language for Semantic Web Rules, Proc. SWWS 01, Stanford, (2001)
5. Tim Berners-Lee, James Hendler and Ora Lassila: The Semantic Web, Scientific American, May (2001)
6. Steven Willmott, Ion Constantinescu, Monique Calisti, Multilingual Agents: Ontology, Language and Abstractions, OAS (2001)
7. Dan Connolly, Frank van Harmelen, Ian Horrocks, Deborah L. McGuinness, Lynn Andrea Stein: DAML+OIL Reference Description, W3C Note (2001)
8. Fabio Bellifemine, Agostino Poggi, Giovanni Rimassa : JADE – A FIPA-compliant agent framework, in Proceedings of PAAM'99, London, April (1999), 97-108.
9. Kostantinos Sagonas , et. al. XSB as an efficient deductive database engine, In ACM Conference on Management of Data (SIGMOD) (1994).
10. Heflin, J. and Hendler, J. Searching the Web with SHOE. In Artificial Intelligence for Web Search. Papers from the AAAI Workshop. AAAI Press, (2000). 35-40.
11. D. Fensel et al.: OIL in a nutshell In: Knowledge Acquisition, Modeling, and Management, Proceedings of the European Knowledge Acquisition Conference (2000).