Semantic Knowledge Model and Architecture for Agents in Discrete Environments

Michal Laclavik and **Marian Babik** and **Zoltan Balogh** and **Emil Gatial** and **Ladislav Hluchy**[1]

1 Introduction

Multi-Agent Systems (MAS) is a powerful paradigm for distributed heterogeneous information systems when representation and reasoning using knowledge is needed. According to the Agent Technology Roadmap [3], which is the result of AgentLink, there are a number of broad technological challenges for research and development over the next decade in the agent technology. Within the presented work we are trying to partially cover some of them. These are:

- Providing better semantic infrastructure
- Apply basic principles of software and knowledge engineering
- Make stronger connection with existing commercial technologies

In the MAS, knowledge is usually represented by states, rules or predicate logic [12] [2]. Although this is extremely powerful, it is hard to capture knowledge from a person or from current information systems in the form of states or predicate logic clauses. Moreover, another difficulty is to present information and knowledge expressed in e.g. predicate logic to the end user [8]. Ontology as understood in the Semantic Web is closer to current information systems. It is based on XML/RDF [9], which can be more easily captured or returned from/to a person and existing information systems because information systems usually have XML based interfaces and XML can be easier presented to user after XSL transformations [10].

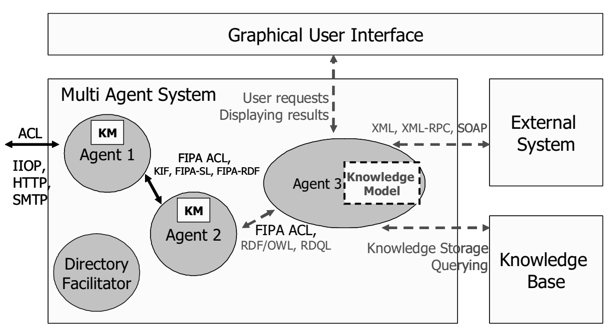

Figure 1. Missing features of FIPA compliant agent architecture

The current MAS platforms follows FIPA [12] standards. FIPA [12] describes the knowledge model based on ontology, but leaves internal agent memory model, its manipulation and understanding of design of agent on developer's decisions. In addition FIPA defines a knowledge manipulation based on content languages such as FIPA-SL [12], FIPA-KIF, which are powerful but lack any interconnection with commercial tools and standards. JADE agent system for example supports FIPA-SL language for message passing, but no FIPA-SL query engine or repository of such knowledge model is available. This is why we see ontology description from the semantic web area (RDF/OWL [9]) more appropriate for real application. Therefore we have decided to integrate semantic web technologies into MAS and create architecture, methodology and software for such integration. In addition, we have developed a generic ontology suitable for representing an agent knowledge model sutable for discrete environment. This model can be further extended, thus enabling its use in many areas especially for the knowledge and experience management [14], [6], [8].

2 Agent Knowledge Model

Most of the agent architectures are combinations of basic architecture types - the so called hybrid architectures. We focused on behavioral architecture, where an agent memory model used for behaviors implementations was created within this work. Proposed Knowledge Model is generic and suitable for applications with discrete, fully observable environments where environment changes can be captured by events.

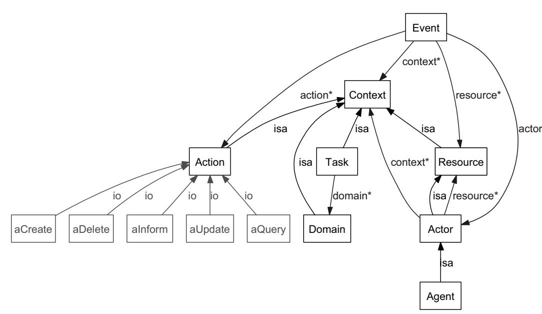

Figure 2. Basic Ontology for Knowledge Modeling

Our model is based on five main elements: Resources, Actions, Actors, Contexts and Events. Figure 2 shows formal graph representation of a model. The proposed model is described also using Description Logic [7] compatible with OWL-DL [9]. In proposed model we expect that all agents share same ontology.

Here we describe only the *Actor* class using Description Logic, which consists of important properties: *context.Actor*, *resource.Actor*.

[1] Institute of Informatics, Slovak Academy of Sciences, Dubravska cesta 9, 845 07 Bratislava, Slovakia, email: laclavik.ui@savba.sk

$$
\begin{aligned}
&Actor \subseteq \\
&\quad resource.Actor(\{resource\})\cap \\
&\quad context.Actor(\{actor\})\cap \\
&\quad Resource
\end{aligned} \tag{1}
$$

The *context.Actor* represents actor's current context. The system or application environment is based on stored events. The events model the environment state. Current state of the environment or actor related environment/context is thus affected by relevant new events.

$$
\begin{aligned}
&context.Actor(\{actor\}) = \\
&\quad f_C(\forall event; actor.Event(\{actor\}) \in \{event\})
\end{aligned} \tag{2}
$$

The *resource.Actor* property stands for all current resources of the actor. These resources are results of actors' intentions or objectives. Such resources are dependable on current actors' environment state/context (*context.Actor*).

$$
\begin{aligned}
&resource.Actor(resource) = \\
&\quad f_R(contextActor(\{actor\}))
\end{aligned} \tag{3}
$$

Functions/algorithms for context and resource updating are specified by (2) and (3). An advantage of such model is that it enables achievement of better results when such algorithms are changed in the future, using the same model and data. Due to storing all events we can model the environment at any moment from past and process it later from any starting point with improved algorithms for context and resource updating. In addition, this model can be successfully used outside the MAS [6]. It can be used in the knowledge intensive application, where we need to model actors and their knowledge model. Often this is the case when we need to model users of the system who are mostly main actors in any application.

3 Modeling and Development Methodology for Agent Design

The developed methodology follows CommonKADS methodology [4]. However CommonKADS is not tied with any modeling tool, knowledge representation or ontologies, we represent knowledge based on the OWL ontology and we model it in the Protégé ontology editor similar as in [4]. Design of the system is based on UML, AUML and MAScommonKADS. When using this methodology, good results can be archived only after several iterations of the process and remodeling after the first developing, use and evaluation of the first system version. Modeling a knowledge model for an application we follow first three CommonKADS models: Organizational or in our case the Environment Model; the Task Model; the Agent or the Actor Model. Modeling the knowledge model we have to extend the generic agent model (Figure 2) with new elements and relations. Results of models is the knowledge model which consists of :

- Ontology developed in Protégé in the OWL format.
- Algorithms for each agent (actor) updating actors' context f_C (2) and resources f_R (3). Often such algorithms are similar or the same for all actors.

The design methodology of a system is based on three UML diagram types in a similar way as known in object oriented programming: Use Case Diagram; Sequence Diagram; Class Diagram.

4 Design and Specification of Agent Software Library

The developed library is based on the JADE agent system [1] and the Jena [13]. It covers functionalities which we identified as missing in current agent architectures such as JADE, the Agent Knowledge Model based on RDF/OWL; Action resource as basics for communication and incorporated in the OWL ontology; Sending ACL based RDF/OWL messages; Receiving ACL based RDF/OWL messages; Incorporation of received information into a model; XMLRPC receiving messages; XMLRPC returning RDF and XML; An inference or RDQL messages handling. The JADE based agent can be developed using this library to support the Jena OWL model as the Agent memory and furthermore it is possible to include XML-RPC based functionalities for presenting knowledge or receiving events from external systems as RDF messages based on a used ontology. Moreover it allows communication among agents based on RDF/OWL.

The developed library is published on the JADE website as the 3rd party software [5] as a way to put together Jena RDF/OWL and JADE Multi-agent system features. The developed library is quite popular and in year 2005 was downloaded 577 times.

5 CONCLUSION

The paper describes how semantic web technologies can be applied in MAS. An agent knowledge model was created enabling the possibility to model the agent environment, agent context and its results. The agent library to support such an agent knowledge model was developed and extends the JADE agent system, which is currently the most popular MAS toolkit. It was proved that such model and the agent architecture are implementable and can be used in different knowledge intensive applications [6], [8] in discrete environments, which was tested in several R&D projects, mainly in the Pellucid IST project. More details with demonstration examples can be found in [8] or on library website [5].

ACKNOWLEDGEMENTS

This work is supported by projects K Wf-Grid IST FP6-511385, SPVV 1025/2004, APVT-51-024604, VEGA No. 2/6103/6.

REFERENCES

[1] Telecom Italia, *JADE Website*, http://jade.cselt.it/, 2004.
[2] G. Caire, *JADE Tutorial Application-defined Content Languages and Ontology*, http://jade.cselt.it/, 2002
[3] M. Luck, P. McBurney, C. Preist, *Agent Technology:Enabling Next Generation Computing, A Roadmap for Agent Based Computing*, 2003
[4] G. Schreiber, M. Crubezy, M. Musen, *A Case Study in Using Protege-2000 as a Tool for CommonKADS*,2001
[5] M. Laclavik, AgentOWL: *OWL Agent memory model*, 2005, http://jade.tilab.com/community-3rdpartysw.htm#AgentOWL
[6] M. Laclavik et al., *Experience Management Based on Text Notes (EMBET)*, Innovation and the Knowledge Economy, Volume 2, Part 1; IOS Press, pp.261-268. ISSN 1574-1230, ISBN 1-58603-563-0,2005
[7] F. Baader et al., *Description Logic Handbook*, ISBN:0521781760, 2003
[8] M. Laclavik: *Ontology and Agent based Approach for Knowledge Management*; PhD Thesis; II SAS, field: Applied Informatics, June 2005
[9] W3C, *OWL*, 2004, http://www.w3.org/TR/owl-features/
[10] M. Laclavik et al., *Methods for Presenting Ontological Knowledge to the User*, Proceedings of Znalosti 2005, pp.61-64. ISBN 80-248-0755-6
[11] M. Wooldridge, *Introduction to MAS*, ISBN:047149691X, 2002
[12] *Foundation for Intelligent Physical Agent*, 2002, http://www.fipa.org/
[13] HP Labs, *Jena Semantic Web Library*, 2004, http://www.sf.net/
[14] J. Kitowski at al., *Model of Experience for Public Organizations with staff Mobility*, KMGov2004, Krems, Austria, May 17-19, 2004