Integrating Ontologies into Distributed Multi-Agent System

Khaoula ADDAKIRI
Department of Mathematics and Computer Science
Université Hassan 1er, FSTS, LABO LITEN
Settat, Morocco

Mohamed BAHAJ
Department of Mathematics and Computer Science
Université Hassan 1er, FSTS, LABO LITEN
Settat, Morocco

***Abstract*—Multi-agent systems have proven to be a powerful technology because of their many advantages in distributed and complex environments however its disadvantage is that is lacks the interconnection with semantic web standards. In this paper we propose a new approach to enhance the interoperability and cooperation of Multi-Agent System (MAS) using semantic web technology (such as RDF and OWL) and we present a proposal for modeling agent based system using Unified Modeling Language.**

Keywords—Web Ontology Language (OWL); Unified Modeling Language (UML); Multi-Agent system (MAS); Ontology.

I. INTRODUCTION

Multi-Agent System (MAS) is a powerful paradigm in nowadays and is become promising means for the development of distributed systems [1, 2, 3]. In Multi-Agent System the interoperability allows agents to communicate and cooperate in order to attain their own objectives and sometimes to solve a common problem. As part of the open multi-agent system, heterogeneous agents, is designed by different organizations and in different languages and can dynamically join or leave the system. This implies a dynamic environment changing functionality, which makes the problem of interoperability of heterogeneous agents very complex.

In the MAS, knowledge is usually represented by states, rules or predicate logic [4, 5]. Although this is extremely powerful, it is hard to capture knowledge from a person or from current information systems in the form of rules or predicate logic clauses. Moreover, another difficulty is to present information and knowledge expressed in e.g. predicate logic to the end user [6]. Ontology has established as a powerful tool to enable knowledge sharing, and a growing number of applications have benefited from the use of ontology as a means to achieve semantic interoperability among heterogeneous, distributed systems. Both ontology and agent technologies are central to the semantic web, and their combined use will enable the sharing of heterogeneous, autonomous knowledge sources in a capable, adaptable and extensible manner[7].

The MAS architectures, Behavioral [8], Belief Desire Intention (BDI) [9], or FIPA [5], have a representation of the knowledge model. In BDI [9], architecture's "belief" represents the knowledge model. In the Behavioral architecture, knowledge is hidden in variables and algorithm states or can be represented by additional mechanisms (rules or ontology and the FIPA describes the knowledge model based on ontology, but leaves internal agent memory model.. In addition, FIPA defines a knowledge manipulation based on content languages such as FIPA-SL [10], FIPA-KIF, which are powerful but lack any interconnection with commercial tools and standards. This is why we see ontology description from the semantic web area (DAML+OIL [11], OWL [12]) to be more appropriate for real application. Furthermore, FIPA defines FIPA-RDF, but again only concerning a message structure. For these reasons all implementations of FIPA architecture are weak and not suitable for developers who want to build their knowledge management systems using software agents.

In order to resolve the problems above, this paper proposed a novel approach to integrate semantic web technologies into MAS using ontology which established as a powerful tool to enable knowledge sharing and to achieve the semantic interoperability among heterogeneous distributed systems. These agents are modeled using MAS-CommonKADS Agent-Oriented Methodology which ensures that many scenarios and cases are analyzed from various perspectives and KQML as an agent communication language.

The rest of the paper is organized as follows. Section 2 reviews the background work in the area of MAS and semantic web .Section 3 presents the agent a knowledge model and tools to build knowledge model in JADE. Section 4 describes the methodology and presents a proposal for modeling agent by using UML Diagram. Section 5 describes the communication language. Section 6 presents the design method .Finally, we draw conclusions in section 7.

II. Related work

At the beginning of the decade, the publishing of the agent technology roadmap [13] pointed out the lack of interconnection between multi-agent systems and semantic web technologies. Since then, several applications and frameworks have been developed to bridge this gap:

B.Schiemann et al. [14], proposed Owl-dl as a FIPA ACL content language in order to facilitate the building of simple interaction protocols that are based on information dialogues and to separate of speech act semantics from content semantic. One limitation of this approach is that they only use propositions or referential expressions in the content field of speech acts (Inform and Query-ref).Our model has no limitation in the content field of speech acts.

M.Laclavik et al. [15], presented a semantic knowledge model of an agent suitable for discrete environments as well as implementation and a use of such model using the Jena semantic web library, the JADE agent system and model it in the Protégé ontology editor. Our models presents a semantic knowledge model by using Jena, JADE , Protégé and JESS to enrich our ontology and KQML ,as well as it allow agents to maintain several dialogues at time.

A.M.Zarafin et al. [16], proposed a semantic description of multi-agent systems, showing the advantages regarding integration with Web semantic technologies. My system shows how to use and integrate ontology in multi-agent system.

Within the presented work we are trying to use the power of both the technologies to cover the following challenges:

- Providing several facilities to make a good decision about how to deal with data.
- Providing better semantic infrastructure (used ontology to support various processes within MAS).
- Make stronger connection between the MAS and existing technologies.

III. Agent Architecture

A. *Agent knowledge model*

Many agent architectures are developed to support intelligent agent such as Belief Desire Intention (BDI), reactive or behavioral architecture. The main focus in literature is on the externals of the agents, their communication with environment and other agents. The internal knowledge model is left for an agent creator. In this work we have based on behavioral architecture.

The motivation for such model came from satisfying most information and knowledge management needs in organizations. Usually in an organization discrete events can be captured as defined in the model such as e-mails received or sent, documents created or opened, activities finished .An analysis of such events can provide useful information and knowledge about organization processes, documents or communication flows. The use of agents to reason on such model is essential if we deal with large distributed or virtual organizations in heterogeneous environment.

B. *Building Knowledge Model in JADE*

Knowledge of the agents is stored in the Web Ontology Language (OWL).It rests primarily on three existing software's: JESS, JENA and Protégé. The JESS (Java Expert System Shell) [17] is a familiar rule engine written entirely in Sun's java language, provides a powerful tool for developing systems with intelligent reasoning abilities. It gives the possibility for building Knowledge in the form of declarative rules and facts, and reason about it .The JENA [18] framework, an open-source Semantic Web framework for Java that provides APIs to build semantic Web applications managing the RDF and OWL languages. It also provides methods to write and read in RDF/XML.

The Protégé is a free, open source ontology editor and knowledge-base framework. The Protégé provides methods to load and save OWL and RDF ontologies, and to edit these ontologies in a powerful way.

IV. Methodology

A. *Method for creating Ontology and agent Model*

In the field of agent systems, researchers have developed several methodologies to support the development of MAS at the analysis and design stages. The majority of these methodologies are coming from three distinct areas: object-oriented software development, knowledge engineering and knowledge-based systems development, and formal software specifications [19]. Examples of such agent-oriented methodologies are the Agent-Oriented Analysis and Design methodology, the formal framework of agent specification based on Z, the MAS-CommonKADS methodology, etc.

The MAS-CommonKADS is extended from the CommonKADS methodology that uses some OO concepts and techniques. This methodology suggests to analyze and describe an agent system in such (meta-) terms as:

- *Agent model* – to detail the characteristics of agents (reasoning capabilities, skills, etc.);

- *Task model* – to specify the tasks assigned to agents;
- *Organization model* – to describe the organizational environment where MAS will operate as well as organization of agent society itself;
- *Expertise model* – to explain what kinds of knowledge are used by agents during solving the tasks (domain, task, inference, problem solving knowledge);
- *Coordination model* – to specify interactions among agents, conversations and protocols used;
- *Communication model* – to specify interactions between agents and people.

When modeling the knowledge model for our application we follow the first three MAS-commonKADS models:

Agent Model: models agent performing actions and tasks. The agent can be a software agent or another entity performing actions and which can be monitored by the system

An important part of the Agent model is agent's context which defines current agent's environment and agent's resources which are the results of accomplished goals. Thus, the Agent model includes definitions of algorithms for the agent context and resources automatic updating defined in the agent model.

Task model: models tasks, activities and actions relevant to actors. Tasks are often related to some resources that we need to define functions for updating the agent context and the resources mentioned above.

Organization model*:* models the environment of application and that of agent. Thus, modeling a resource element needs to be extended by new types of resources such as documents, contacts or services. The resources considered as results of actor's goals have to be modeled as well. Usually results of agents' goals can be considered as resources e.g. in searching agents (found documents).

Defining ontology (developed in Protégé in the OWL format) of mentioned models must be followed by iterative refining to include all needed elements in the model. Refining the model requires combination of actions, resources and actors to be captured as events and all possible events could be created from defined ontology elements

B. Design of Agent Based System

The design methodology of the system is based on three UML diagram types, similarly to those in the object oriented programming:

- Use Case Diagram
- Sequence Diagram
- Class Diagram.

The Use Case Diagram: According to UML standards, the use case definition is[20]:"A use case is a coherent unit of functionality provided by a system manifested by sequences of messages exchanged among the system and one or more actors together with actions performed by the system." (Figure 1).

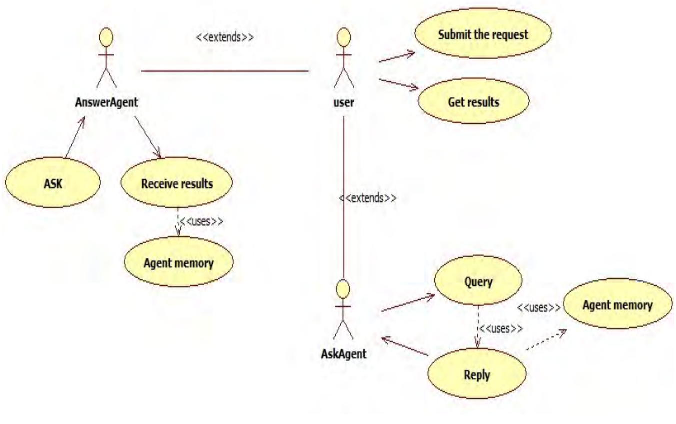

Figure 1:Agents Use Case diagram

The Sequence Diagram: It represents communication between agents and also interaction with external systems (GUI or other interfaces) (Figure 2).

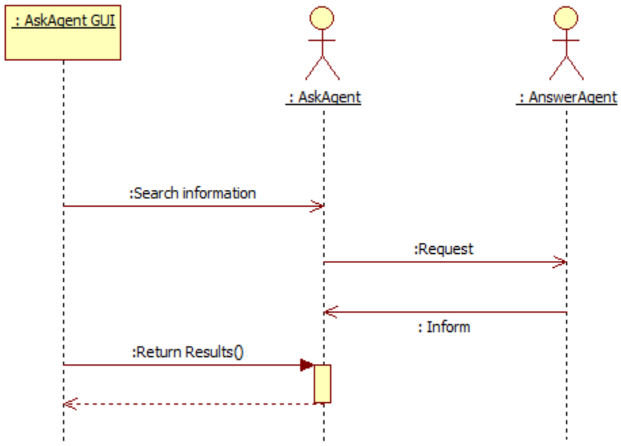

Figure 2:Agent Sequence diagram

The Class Diagram: The purpose of a class diagram is to represent the design of the agent type by describing all agent behaviors. Behaviors can be considered as something owned by agents and should be displayed as methods in the agent class (Figure 3).

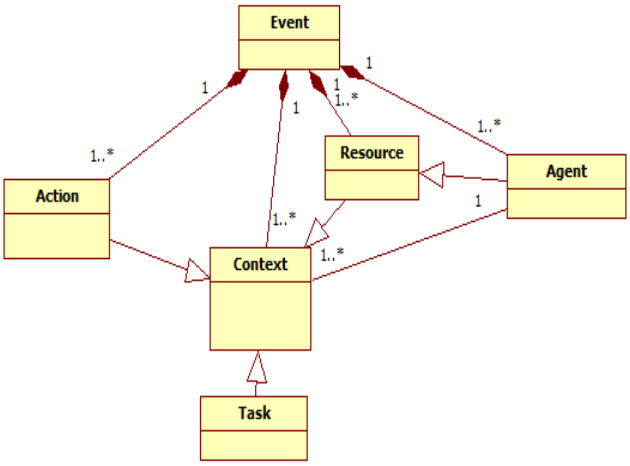

Figure 3:Agent Class Diagram

V. AGENT COMMUNICATION WITH KQML

Knowledge Query and Manipulation Language (KQML) is a language and protocol for exchanging information and knowledge independent of content syntax and applicable ontology [21].

A KQML message has three layers [22].

- Content - is the actual content of the message in the program's own representation language.
- Communication - encodes a set of features to the message which describe the lower-level communication parameters.
- Message - identify the network protocol to be used to deliver the message and supply a speech act or performative that the sender attaches to the content.

A KQML message is called a performative. Following are the parameters of a performative:

:sender	The actual sender of the performative.
:receiver	The actual receiver of the performative.
:from	The origin of the performative in :content when forward is used.
:to	The destination of the performative in :content when forward is used.
:in-reply-to	The expected label in a response to a previous message.
:reply-with	The expected label in a response to the current message.
:language	The name of the representation language of the :content.
:ontology	The name of the ontology assumed in the :content parameter.
:content	The information about which the performative expresses an attitude.

A KQML message from AskAgent representing a query about the course offered by Department to AnswerAgent is formed as following:

(ask-one
:sender AskAgent
:content (Course IA? Coursename)
:receiver AnswerAgent
:reply-with CourseofferedByDepartment
:language prolog
:ontology Onto1
)

The Institute agent sends a reply to Course agent as the following KQML message.

(tell
:sender AnswerAgent
:content (course IA fsts.)
:receiver AskAgent
:in-reply-to CourseOfferedByDepartment
:language prolog
:ontology Onto1
)

VI. DESIGN METHODS

We extend this idea of [23] in which an agent based system design need to consider levels and those levels need to be considered such as communication component (detail level, intermediate level and policy level), the agent knowledge component (abstract level), the control component and the execution monitoring component (organization level).Fig 7 shows the different level of agent -based system design.

Organization level	Planning algorithm
Abstract level	OWL ontology
Policy Level	KQML
Intermediate level	
Detail level	TCP/IP

VII. CONCLUSION

Multi-Agent System (MAS) is a powerful paradigm in nowadays and is become promising means for the development of distributed systems. However its disadvantage is that it lacks the interconnection with semantic web standards such as Ontology Web Language (OWL). This paper describes how semantic web technologies as Ontology web Language (OWL) can be applied to build an intelligent JADE agent and leverages the power of both ontology and Multi-agent system. So, an agent knowledge model was created using protégé and using JESS language to enrich the ontology. In this system, we have developed agents, they are communicating via one of the most popular agent communication language, KQML. The combined use of Ontologies and MAS enable the sharing of heterogeneous, autonomous knowledge sources in a capable, adaptable and extensible manner. The idea offered by KQML is that of having specialized agents, called facilitators, which with the use of the appropriate KQML performatives can help agents to find other agents that can perform desired task for them. Also the UML Diagrams allow using a high level of abstraction and can well describe a great population of agents and visualizing their mutual communications.

REFERENCES

[1] D.Chess,C.Harrison, A.Kershenbaum,"Mobile Agentes: Are They a Good Idea?", IBM ResearchReport, 1998 .

[2] J.Kelemen,"The Agent Paradigm" – Foreword. In Computing and Informatics,Vol. 22, 2003, No. 6, p. 513–519.

[3] D.Lavbic, R.Rupnik "Multi-Agent System for Decision Support in Enterprises", JIOS, Vol. 33, No. 2 (2009).

[4] G.Caire, "JADE Tutorial Application-defined Content Languages and Ontology".http://jade.cselt.it

[5] FIPA: Foundation for Intelligent Physical Agents website. http://www.fipa.org/.

[6] M.Laclavik,"Ontology and Agent based Approach for Knowledge Management", Ph.D. Thesis; Institute of Informatics, 2005.

[7] P.M.Dolia,"Integrating Ontologies into Multi-Agent Systems Engineering (MaSE) for University Teaching Environment", Journal of Emerging Technologies in Web Iintelligence, 2010.

[8] Telecom Italia: JADE (Java Agent DEvelopment Framework) Website. http://jade.cselt.it/

[9] M.Wooldridge,"Introduction to MultiAgent Systems", 2002.

[10] FIPA: FIPA SL Content Language Specification, 2001

[11] DARPA: DAML Website. http://www.daml.org/.

[12] W3C: Web Ontology Language (OWL). http://www.w3.org/TR/owl-features/.

[13] M. Luck, P. McBurney, and C. Preist, "Agent technology: Enabling next generation computing", AgentLink II, 2003.

[14] B.Schiemann, U.Schreiber, "OWL DL as a FIPA ACL content language", In Nicola Guarino Roberta Ferrario and Laurent Prevot, editors, Proceed-ings of the Workshop on Formal Ontology for Communicating Agents (FOCA),18th European Summer School of Language, Logic and Information, pages 73–80, University of Malaga, July 2006.

[15] M.Laclavik, Z.Balogh, M Babik,L.Hluchy, "AgentOwl: semantic knowledge model and agent architecture", Computing and Informatics, Vol. 25, 2006, 421–439.

[16] A.M.Zarafin, A. Zimmermann,O.Boissier, "Integrating Semantic Web technologies and Multi-Agent Systems: a Semantic Description of Multi-Agent Organizations", 15-16 October 2012, Dubrovnik, Croatia.

[17] JessWeb Site [Online]. Available: http://www.jessrules.com

[18] Semantic Web Framework for Java. http://jena.sourceforge.net/

[19] V. Abramov, N. Szirbik, J. Goossenaerts, T. Marwala, P. De Wilde, L. Correia, P.Mariano, and R. Ribeiro. Ontological basis for open distributed multi-agent system.In Symposium on Adaptive Agents and Multi-Agent Systems, 2001.

[20] A.Chella, M. Cossentino, U. L.Faso:"Applying UML Use Case Diagrams to Agents Representation", Proc. of AI*IA 2000 Conference. (Milan, Italy, Sept. 2000).

[21] T.Finin, Y.Labrou and J. Mayfield, "KQML as an agent communication language",http://www.cs.umbc.edu/kqml/papers/kqmlacl.pdf.

[22] D.Roman, "Agents Communication Languages (ACL)",http://www.engr.uconn.edu/%7Eibrahim/publications/acl.html.

[23] T.Finin et al." KQML :An Introduction and Knowledge Exchange Protocol'', in Knowledge Building and Knowledge and Knowledge sharing,Ohmasha and IOS Press