Integrating Ontologies into Multi-Agent Systems Engineering (MaSE) for University Teaching Environment

Dr. Prashant M. Dolia

Sr. Lecturer, Department Of Computer Science & Applications, Gaurishankar Lake Road, Bhavnagar University, Bhavnagar 364022 Gujarat. India
prashant_dolia@rediffmail.com

Abstract---Multi-agent systems have received much attention in recent years because of their many advantages in complex and distributed environments. There are a number of methodologies have been proposed for multi-agent engineering process such as Multi-agent System Engineering (MaSE). I have used the MaSE engineering process for the development of my ontology based multi-agent system for the Academic Institute. In the era of Semantic Web, the ontology has established as a powerful tool to enable knowledge sharing and it is an important means in Semantic Web to achieve the 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. Ontology is used throughout the multi-agent system to assist the interactions among different agents as well as to improve the quality of the service provided by each agent.

This paper focuses on the utilization of combining both Ontology and Multi-Agent System (MAS) structure towards system integration for University teaching environment. In this paper, I include system prototype for ontology based multi-agent system. I have used Web Ontology Language (OWL) for the development of domain ontology for the Academic Institute and KQML as an agent communication language. Finally I have developed ontology-based multi-agent system for the university teaching environment by taking the benefits of both renowned technologies.

Index Terms-- Ontology, Multi-agent systems, MaSE, OWL, KQML

I. Introduction

An agent is a software program that automatically performs tasks on behalf of the user [7]. A multi-agent system is a system composed of multiple interacting intelligent agents. One of the current factors fostering multi-agent development is the increasing popularity of the Internet, which provides the basis for an open environment where agents interact with each other to reach their individual or shared goals. A multi-agent system is a loosely coupled network of problem-solver entities that work together to find solution to problems that are beyond the individual capabilities or knowledge of each entity [9].

The agent paradigm is successfully employed in those applications where autonomous, loosely-coupled, heterogeneous, and distributed systems need to interoperate in order to achieve a common goal. In a multi-agent system the agents communicate between them in order to fulfill the global goal or their local goals. Also, 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.

This is particularly important for multi-agent systems, where the content of messages exchanged among agents must conform to some ontology in order to be understood. Through the collaboration between different agents, my aim is to achieve a highly efficient, flexible, customizable system that provides better communication, interaction and management among all users engaged in the academic institutions. Ontology is used throughout the multi-agent system to assist the interactions among different agents as well as to improve the quality of the service provided by each agent [5].

Various systems and models have been developed so far to achieve the advantages of both multi-agent systems and ontology paradigm. Mihaela Oprea [13] represents Ontology Mapping in Open Multi-Agent Systems. Cu D. Nguyen, Anna Perini and Paolo Tonella [14] show Ontology-based Test Generation for Multi-agent Systems. Rosario Girardi, Carla Gomes de Faria, Leandro Balby[15] represents Ontology-based Domain Modeling of Multi-Agent Systems. Victoria Iordan, Antoanela Naaji, Alexandru Cicortas [10] shows how to derive ontology using multi-agent systems. My system shows how to use and integrate domain ontology in a multi-agent system.

My system developed a prototype which fulfills basic needs of users of academic institutes by using the power of both the technologies. Different agents providing several facilities help applications and users to make

doi:10.4304/jetwi.2.1.42-47

better decisions about how to deal with the data. Ontology represents the domain knowledge and can be used to support various processes within a multi-agent system. Ontology is high expressive knowledge models and as such increase the expressiveness and intelligence of a system. Ontology provides a common way of understanding between different agents. Here, ontology is passed as a parameter between different agents during their communication process.

II. MULTI-AGENT SYSTEMS ENGINEERING METHODOLOGIES

There are number of methodologies that have been proposed for multi-agent engineering process. In this paper, I have used the MaSE engineering process for the development of my ontology based multi-agent system for Academic Institute because other methodologies do not adequately address the information domain of the system.

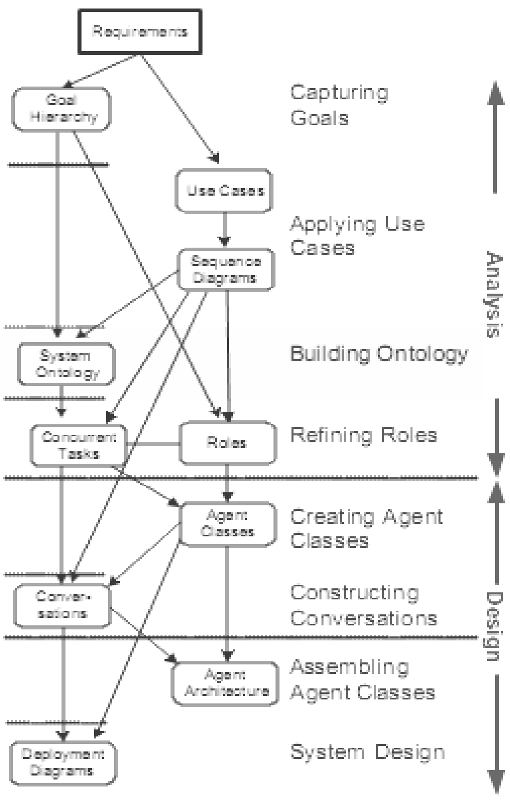

Fig. 1. Extended MaSE Phases, Steps and Models

Multi-agent Systems Engineering (MaSE) has been developed at AFIT to assist in the development of multi-agent systems by leading the designer from the initial system specifications to a set of formal design documents [DeLoach and others 2001]. The MaSE divides the development of Multi-agent systems into analysis, design and implementation phases [1]. MaSE originally consisted of three steps in the analysis phase and four steps in the design phase. The developers of MaSE intended for these phases and steps to be applied iteratively. During system implementation the models from the analysis and design phases are used to program the system into code. The transformations from each step in MaSE are formally defined and provide the engineering approach needed for multi-agent system engineering. Despite its benefits in multi-agent systems design, however, MaSE fails to address the design of the information domain [4].

As shown in figure.1, Jonathan DiLeo, Timothy Jacobs and Scott DeLoach have expanded MaSE to include ontology as a mechanism for specifying the information domain of the system and the individual agents and corrects this deficiency by extending MaSE to include steps to specify the information domain and to use the objects defined in that domain, providing the designer with a complete set of design documents [3]. Ontology is used to specify the classes, properties, object constants, and axioms that a system and its components use to represent the domain in which they operate.

III. BENEFITS OF USING ONTOLOGY IN MULTI-AGENT SYSTEMS

The term ontology is defined as “an explicit specification of a conceptualization” (Gruber, 1994) or “a set of types, properties, and relationship types” (Garshol, 2004). Ontology defines concepts in a specific area and their relationships; however, ontology is more than an agreed-on term. It has a set of well-defined constructs that can be leveraged to build structured knowledge. Although taxonomy enhances the semantics of terms in a vocabulary, ontology includes richer relationships among terms (Smith, 2004). Ontology is the framework of the semantic web, and permits intelligent navigation (Information Intelligence, 2004). For humans, ontologies enable better access to information and promote reuse and shared understanding; for computers, ontologies facilitate comprehension of information and more extensive processing (Ontology Engineering) [6]. Ontology defines the terms used to describe and represent an area of knowledge. The ontology specifications can be passed as parameters in agent conversations. The benefits of integrating agent specification and system ontologies are defined as following [4].

Communication: The KQML (Knowledge Query and Manipulation Language) is an effort to standardize the communication language between software agents. The standard defines the semantical meaning of ASCII expressions, being exchanged in an agent community and makes abstraction of transport issues. From the outside, it appears as if each agent manages its own knowledge base, which typically consists of 'beliefs' and 'goals' (determining the agent's behavior). Part of this knowledge may be shared with other agents, which can query or manipulate it.

Competent Reuse: Ontology also allows the reuse of agents. Components that are designed for separate domains will not be able to directly communicate with each other some type of translating component must build. If ontology is specifies for the component initially, future designers would normally know if a translator is needed for the component to properly operate once integrated into a new software project.

IV. Ontology-Based Communication Model

The FIPA communication model defined in [FIPA00023] is based on the assumption that communicating agents share ontology of communication defining speech acts and protocols (see Figure 2). In order to have fruitful communication, agents must also share ontology of their domain of application. In an open environment, agents are designed around various ontologies (either implicit or explicit). For allowing their communication, explicit ontologies are however necessary, together with a standard mechanism to access and refer to them.

Without explicit ontology, agents need to share intrinsically the same ontology to be able to communicate and this is a strong constraint in an open environment where agents, designed by different programmers or organizations, may enter into communication.

An explicit ontology is considered to be declaratively represented as opposed to implicitly, procedurally encoded. It can be considered as “a referring knowledge” and, as a consequence, could be outside the communicating agents; managed by a dedicated ontology agent [12].

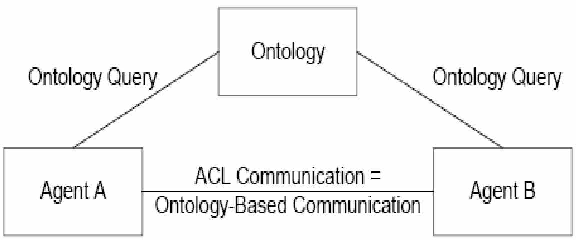

Figure 2. Ontology-Based Communication Model

There are many applications that benefit from having a dedicated agent that manages and controls access to a set of explicit ontologies.

V. System Prototype for Ontology Based Multi-Agent System

The overall architecture of the ontology based multi-agent system is presented in Figure 3. We have followed the FIPA agent management specification (FIPA SC00023K 2004), which is one of the most widely adopted agent management standards available. The Foundation for Intelligent Physical Agents (FIPA) was formed in 1996 to produce software standards for heterogeneous and interacting agents and agent-based systems. FIPA is a non-profit organization who dedicates their efforts to the standardization of agent-based technologies and multi-agent systems. FIPA specifications represent a collection of standards which are intended to promote the interoperation of heterogeneous agents and the services that they can represent.

As shown in Figure 3, the system consists of seven major components: Directory Facilitator Agent, Agent Management Service Agent, Interface Agent, Faculty Agent, Institute Agent, Course Agent and Ontology Agent. All those agents interact and work together to provide different services to the end users, related to Academic Institute.

In figure 3, the Agent Management Service Agent and the Directory Facilitator Agent are used from the FIPA agent management specification. According to FIPA agent management specification, the Agent Management Service Agent is responsible to manage basic operation of an agent platform (FIPA SC00023K 2004) like creation and/or deletion of agents, query platform profile, authentication of agents, and the registration and/or deregistration of agents. In short it provides the basic control of agents for the multi-agent system.

According to FIPA Agent Management Specification, a Directory Facilitator (DF) Agent is an optional component of the Agent Platform, but if it is present, it must be implemented as a DF service. The DF provides yellow pages services to other agents. Agents may register their services with the DF or query the DF to find out what services are offered by other agents. Multiple DFs may exist within an Agent Platform and may be federated. The DF is a reification of the Agent Directory Service in [FIPA00001]. At any time, and for any reason, the agent may request the DF to modify its agent description. An agent may search in order to request information from a DF.

According to FIPA Ontology Service Specification [11] an Ontology Agent (OA) is a dedicated agent, whose role in the community is to provide some or all of the following services: discovery of public Ontologies in order to access them, maintain (for example, register with the DF, upload, download, modify) a set of public Ontologies, translate expressions between different Ontologies and/or different content languages, respond to query for relationships between Ontologies, and, facilitate the identification of a shared ontology for communication between two agents.

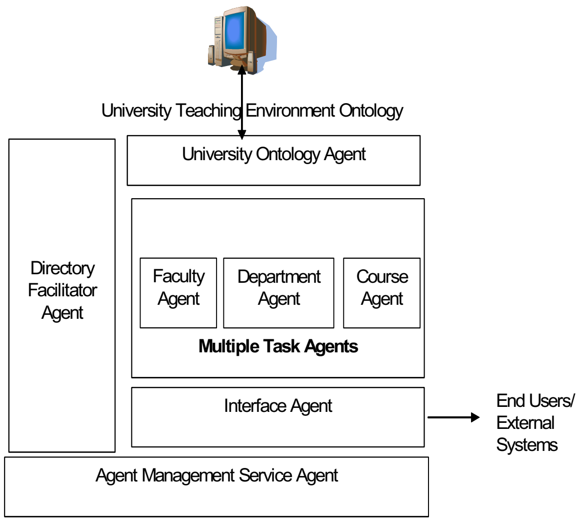

Figure 3. Ontology based Multi-agent System Architecture

The faculty agent is used to provide the response for various types of queries such as identification number of faculty, area of interest, faculty belongs to which department, faculty is assigned to which course etc. The institute agent is responsible to provide the different types of information such as identification number, various courses offered by institutes and name of the institute. The course agent offers the information such as course identification number, course name and it is offered by which institute.

The Interface Agent is used to connect end users or external systems together with the multi-agent system. It reacts to different request made by the end user or external system, and translates these commands into agent understandable requests, and sends them to appropriate agents [5].

VI. DOMAIN ONTOLOGY

The Domain Ontology is the guideline in defining how data is transferred between the agents. The Academic Institute Ontology is developed by defining these concepts and the relationships that exist within its domain.

Figure 4 shows the UML class diagram which includes the concepts of Academic Institute Ontology. It also includes the sub-concepts such as permanent faculty and ad-hoc faculty. It reveals the properties of each concepts and sub-concepts.

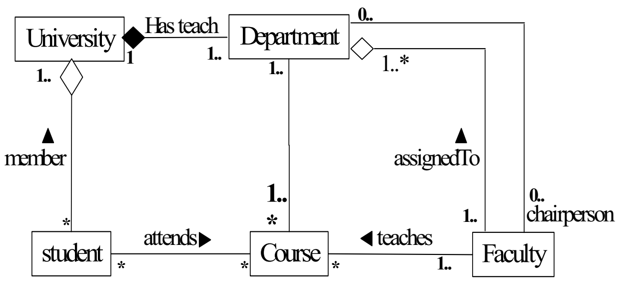

Figure 4. Domain Ontology for University Teaching Environment

Since UML is only a modeling tool, therefore the Domain Ontology needs to be translated into a machine readable language and for this we have utilized OWL to achieve this goal [5]. As OWL is written in XML, therefore it has all the benefits that XML can provide. It allows information to be exchanged easily across different platforms on top of that; it also allows the exchange to be taken place between different applications. OWL also captures information relating to classes, properties, attributes as well as the constraints displayed by UML and OCL together.

The Web ontology language (OWL) can formally describe the semantics of classes and properties used in Web documents. For machines to perform useful reasoning tasks on these documents, the language must go beyond the basic semantics of RDF Schema.

VII. ONTOLOGY IN OWL LANGUAGE

In Academic Institute domain ontology, I have defined the following OWL classes for University Teaching Environment.

<owl: Class rdf:ID=”University”/>
<owl: Class rdf:ID=”Faculty”/>
<owl: Class rdf:ID=”Department”/>
<owl: Class rdf:ID=”Course”/>
<owl: Class rdf:ID=”Student”/>

In OWL, I can also define the relationship between the classes. Following are the statements in OWL, which expressed the relationship between classes of Academic Institute.

<owl: ObjectProperty rdf:ID=”hasTeach”>
<rdfs: domain rdf:resource=”#Faculty”/>
<rdfs:domain rdf:resource= “#Course”/>
</owl: ObjectProperty>

Data type can also be defined in OWL through, owl: DatatypeProperty. For example, facultyID is a string; therefore it can be expressed as:

<owl: DatatypeProperty rdf: ID=”facultyID”>
<rdfs:domain rdf:resource=”#Faculty”/>
<rdfs:range rdf:resource=”&xsd; string”/>
</owl: DatatypeProperty>

Ontology is usually expressed in a logic-based language, so that detailed, accurate, consistent, sound, and meaningful distinctions can be made among the classes, properties, and relations. Some ontology tools can perform automated reasoning using the ontologies, and thus provide advanced services to intelligent applications such as conceptual/semantic search and retrieval, software agents, decision support, speech and natural language understanding, knowledge management, intelligent databases, and electronic commerce [12].

VIII. Agent Communication with Knowledge Query and Manipulation Language (KQML)

KQML is a high-level, message-oriented communication language and protocol for information exchange independent of content syntax and applicable ontology [8].

A KQML message has three layers [2].

- Content - bears the actual content of the message in the program's own representation language.
- Communication - encodes a set of features to the message that describe the lower-level communication parameters.
- Message - identify the network protocol with which 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 final destination of the performative in *:content* when forward is used.
:in-reply-to - the expected label in a response to a previous message (same as the value of the 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 (e.g., set of term definitions) assumed in the *:content* parameter.
:content - the information about which the performative expresses an attitude.
A KQML message from Course agent representing a query about the course offered by Department to Academic Institute agent is formed as following.

```
(ask-one
    :sender Course
    :content (Course MCA? Coursename)
    :receiver Department
    :reply-with CourseOfferedByDepatment
    :language prolog
    :ontology Academic
)
```

Within a time, Institute agent sends a reply to Course agent as the following KQML message.

```
(tell
    :sender Department
    :content (course MCA Bhav Uni.)
    :receiver Course
    :in-reply-to CourseOfferedByDepartment
    :language prolog
    :ontology Academic
)
```

IX. Conclusion

This paper is aimed at applying ontology based Multi-agent approach to provide different services like course registration and scheduling in academic institutions. Also, it discovers a seamless information processing system across an organization that the user can access from any location. This system leverages the power of both ontology and Multi-agent system. In this system, I have developed here six agents. They are communicating via one of the most popular agent communication language, KQML. In KQML, I am passing academic institute ontology as a parameter. The combined use of Ontologies and Multi-agent technologies 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 find other agents (or be found by other agents) that can perform desired task for them.

The implementation of a computer understandable representation of the semantics of academic programs is complex. That's why academic institutions struggle in implementing pervasive information systems that offer services to help all actors in this context. These services are demanded by, *e. g.*, students who want to plan their curricula correctly, or who want to know which courses can be used for a different academic program or at a different academic institution. In this paper, I introduced a distributed ontological approach to represent the semantics of academic programs and their examination regulations, the universities' supply, and individual results. It allows academic institutions to implement applications that offer the demanded services and that use these Ontologies as a common basis.

References

[1] DeLoach, S., M. Wood and C. Sparkman, "Multiagent Systems Engineering", The International Journal of Software Engineering and Knowledge Engineering, Vol 11 no. 3, (June 2001)

[2] Dumitru Roman, Digital Enterprise Research Institute "Agents Communication Languages (ACL)" http://www.engr.uconn.edu/%7Eibrahim/publications/acl.htm

[3] Jonathan DiLeo, Timothy Jacobs, Scott DeLoach, "Integrating Ontologies into Multiagent Systems engineering" http://ftp.informatik.rwth-aachen.de/Publications /CEUR-WS/Vol-57/id-3.pdf.

[4] Scott A. DeLoach, Mark F. Wood, and Clint H. Sparkman "Multi-agent System Engineering"

International Journal of Software Engineering and Knowledge Engineering. Vol. 11, No. 3 (2001) 231-258.

www. macr.cis.ksu.edu/publications

[5] Kai Yang , Amanda Lo , Dr. Robert Steele on "Ontology-Based Multi-agent System for the Accommodation Industry" http://ausweb.scu.edu.au/aw07/papers/refereed/yang/paper.html

[6] M. Kifer, J. de Bruijn, H. Boley, and D. Fensel. "A realistic architecture for the Semantic Web". In Proceedings of the International Conference on Rules and Rule Markup Languages for the Semantic Web (RuleML-2005), Ireland, Galway, November 2005

[7] S. genesereth M. and Ketchpe (1994), "Software agents", Comm. of ACM. Vol. 37, No. 7, pp. 48-53.

[8] Tim Finin, Yannis Labrou and James Mayfield, "KQML as an agent communication language" http://www.cs.umbc.edu/kqml/papers/kqmlacl.pdf

[9] V.R., Durfee, E.H., Lesser and Corkill, D.D., "Trends in Cooperative Distributed Problem Solving", IEEE Transactions on Knowledge and Data Engineering, Vol. March 1989, KDE-1(1), pages 63-83.

[10] Victoria Iordan, Antoanela Naaji, Alexandru Cicortas, "Deriving ontology using multi-agent systems". WSEAS Transactions on Computers, Volume 7, Issue 6 (June 2008)

[11] FIPA "Ontology Service Specification" Document Number : XC00086D] Document Source: FIPA Architecture Board www.fipa.org

[12] Peter V. Mikhalenko "The benefits of Web Ontology language in Web Applications" TechRepublic http://articles.techrepublic.com.com/5100-10878_11-5060266.html

[13] Bd. Bucuresti "Ontology Mapping in Open Multi-Agent Systems" *Department of Informatics, University Petroleum-Gas of Ploiesti* http://www.ici.ro/SIC/sic2007_2/

[14] Cu D. Nguyen, Anna Perini and Paolo Tonella "Ontology-based Test Generation for MultiAgent Systems (Short Paper)" http://www.aamas-conference.org/Proceedings/aamas08/proceedings/pdf/paper/AAMAS08_0528.pdf

[15] Rosario Girardi, Carla Gomes De Faria, Leandro Balby "Ontology-based Domain Modeling of Multi-Agent Systems" Department of Computer Science, Federal University of Maranhao

AUTHOR'S BIOGRAPHY

Dr. Prashant M. Dolia is Sr. Lecturer at Department of Computer Science & Applications, Bhavnagar University, Bhavnagar, Gujarat, India. He has done B.Sc (Physics), M.C.A. and Ph.D. (Computer Science) from the Bhavnagar University, Bhavnagar, Gujarat, India. He is having 10 years of teaching experience at Post Graduate level. His area of research includes Web Intelligence, Data mining & Warehousing, Wireless technology, Linux platform and Kernel Programming. Dr. Prashant Published 5 papers in the International Journal and 2 papers in the National Journal.

Dr. Prashant M. Dolia