

©2009 IEEE. Personal use of this material is permitted. However, permission to reprint/republish this material for advertising or promotional purposes or for creating new collective works for resale or redistribution to servers or lists, or to reuse any copyrighted component of this work in other works must be obtained from the IEEE.

Use and Modeling of Multi-agent Systems in Medicine

Maja Hadzic, Darshan S. Dillon, Tharam S. Dillon

Curtin University of Technology

Research Lab for Digital Health Ecosystems

Perth 6845, Western Australia, Australia

e-mail: d.dillon@curtin.edu.au, m.hadzic@curtin.edu.au, t.dillon@curtin.edu.au

Abstract- Multi-Agent System (MAS), and more specifically, ontology-based MAS, are increasingly being proposed and used within the medical domain. In this paper we represent an ontology-based multi-agent system specifically designed to intelligently retrieve information about human diseases. The human disease ontology is organized according to the four dimensions: disease types, symptoms, causes and treatments. The multi-agent system consists of four different types of agent: Interface, Manager, Information and Smart agent. We use of UML 2.1 to model social and goal-driven nature of agents. We believe that UML 2.1 has not only provided a way for standardized notation of MAS, but also for effective representation of the dynamic processes associated with these MAS.

*Keywords-*UML, modeling of multi-agent systems, ontology-based multi-agent systems, information retrieval.

I. INTRODUCTION

Use of agent-based systems enables us to propose and design intelligent and dynamic medical information systems [1]. One of the main advantages of agents is their autonomy. They can act independently from the user and from the rest of the system, and make decisions on their own. Even though the agent is able to act autonomously, it has to be sociable and collaborate with other agents of the system in order to address more complex problems. Only when cooperating, coordinating their actions, sharing tasks and results with other agents, can multi-agent systems reach their full potential. The collaborative nature of agents enables the multi-agent system to find solutions to complex problems through carrying out distributed problem solving. Some agents are mobile i.e. they are capable of migrating to different places. This feature can increase dynamics and efficiency of the whole system.

Being goal-driven is a feature of many different agents. The fact that an agent has an overriding goal, regardless of the specifics of its processing, endows it with many other features. Specifically, it is proactive, i.e. it takes actions on its own initiative, and it is intelligent, i.e. it reasons and chooses to perform the most beneficial actions towards achieving its goals.

Ontologies are increasingly being used in association with multi-agent systems. Ontology captures and represents knowledge of a certain domain in machine-readable form [2]. Within multi-agent systems, ontologies can be used to support various intelligent actions.

One of the ongoing issues within the modelling of multi-agent systems is the lack of a standardized modelling language. Some research groups have proposed the use of UML to model agent-based systems [3]. Others have

extended UML to suit modelling requirements specific to multi-agent systems. We have noticed a number of problems with the existing approaches. The major problems relate to the inconsistent semantics of the existing UML Diagrams, and the unintuitive and complex notation (see Section 2).

In Section 2, we will discuss a number of multi-agent systems designed for medical domain and a number of approaches to modeling such multi-agent systems. We will explain the Generic Human Disease Ontology (GHDO) structure and Multi-Agent System (MAS) which is based on this ontology in Section 3. In Section 4, we will illustrate how the sociable and goal-driven nature of the GHDO-based MAS can be represented using UML2.1 [4] Sequence and Composite Structure Diagrams respectively. This paper is concluded in Section 5.

II. RELATED WORK

A. Multi-agent systems in medical domain

Multi-agent systems are increasingly being used in the medical domain. Some of these systems are designed to use information available through a specific medical and health institution, others systems use information from Internet.

As examples of institution-specific multi-agent systems we mention Agent Cities [5], AADCare [6] and MAMIS [7]. The Agent Cities [5] agents enable the user to access his/her medical record, to search for medical centres on the basis of a given set of requirements, or to request and make appointments. AADCare [6] is a decision support system used by physicians. It matches the patient's record against the predefined domain knowledge. This domain knowledge can contain knowledge regarding a specific disease, clinical management plans, patient records etc. MAMIS [7], Multi-Agent Medical Information System, provides ubiquitous information access to physicians and health professionals.

BioAgent [8], Holonic Medical Diagnostic System [9] and Web Crawling Agents [10] are some examples of internet-based multi-agent systems. BioAgent [8] is a mobile agent specifically designed to retrieve information about genome analysis. It that travels among multiple locations, collects information from each location and integrates information before deploying answer to the user. Holonic Medical Diagnostic System [9] matches the comprehensive computer readable patient record (computer readable patient pattern, CRPP) against the information available through Internet in order to provide enough evidence for correct diagnosis of the patient. Different web crawling agents [10] use information

available on mutated genes to fetch information about associated diseases.

The need for design a multi-agent system for the purpose of dynamic information retrieval regarding common knowledge of human diseases has been explained in [1,11]. Providing a systematic overview of the different aspects of human disease will bring a significant contribution to the advancement of human disease research and practice.

B. Modeling of multi-agent systems

A number of researchers have attempted to use UML in modeling Multi-agent systems [3, 12, 13, 14, 15]. This great initiative has contributed significantly in development of standardized methodology for modeling multi-agent systems. However, a number of issues have been identified and these need to be addressed. These issues include:

(1) undetailed presentation. For example, VisualAgent [14] can be used to present some preliminary ideas, but it virtually lacks existing UML diagrams or stereotypes.

(2) changing the semantics of the sequence diagram. For example, Kavi *et al.* [12] use smiley faces, thought clouds, and the like, which have not been defined by the UML semantics. Da Silva *et al.* [13] change the semantics of rectangles without the use of a stereotype. Odell *et al.* [15] define a rectangle to be a Agent/Role combination in sequence diagrams which means that a single Agent can be represented by multiple rectangles, each rectangles representing a single role.

(3) no use of stereotypes. For example, Odell *et al.* [15] did not use stereotypes to define an agent.

Our work makes following contributions to the existing modeling approaches:

(1) use of different UML diagrams to define different aspects of multi-agent systems. This allows for a more detailed presentation. For example, sequence diagrams describe social nature of agents while the composite diagrams define each agent in greater detail.

(2) no change in semantics of the existing UML diagrams. For example, we use a single rectangle to represent a composite class (which corresponded to an Agent) with ports (see UML 2.1) to represent roles.

(3) use of a stereotype to define an agent.

III. GHDO-BASED MULTI-AGENT SYSTEM

A. Generic Human Disease Ontology (GHDO)

We have proposed an ontology-based multi-agent system to support intelligent information management, sharing and retrieval for human disease study and research community. Using ontologies over knowledge bases is preferred because ontologies, as highly expressive knowledge models, increase system expressiveness and intelligence [16] and sharability of this knowledge. The proposed technological solution incorporates an ontology framework of the Generic Human Disease Ontology which contains generic information regarding human diseases [11]. The ontology is designed as

superset of four subontologies capturing Disease Types, Phenotype (Symptoms), Causes and Treatments.

B. Generic Human Disease Ontology (GHDO)-based MAS

We propose a distributed multi-agent system with the agents as its processing nodes [1]. The agents of GHDO-based MAS are sociable. This means that they are able to interact with each other in order to cooperate, collaborate and negotiate with respect to information, knowledge and services. The agents are cooperatively working on different levels within this multi-agent system and are dependent on each other with the respect to the same goal. To reduce the complexity of the overall tasks, it is subdivided among various agents. Individual agents work only on their aspect of the problem. In the final steps, the subsolutions are combined into a final solution.

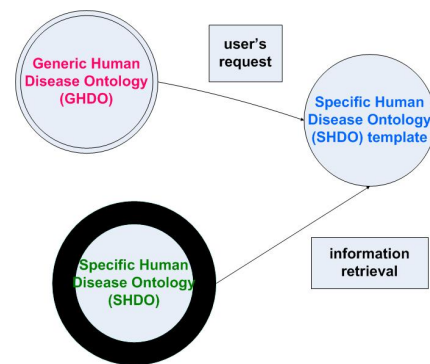


Figure 1. GHDO, SHDO template and SHDO

The GHDO ontology is used to support the information retrieval process within the multi-agent structure. GHDO is used to derive the Specific Human Disease Ontology template (SHDO template). The SHDO template specifies the information an user is interested in. For example, if a user is interested only in causes of a specific disease, the SHDO template will contain only the Causes subontology of the GHDO. When agents feed instances into this SHDO template after information retrieval, the SHDO template gets converted into a Specific Human Disease Ontology (SHDO) which is presented to the user as the answer to his/her query. The SHDO contains information about a specific disease such as bipolar disorder, SARS, diabetes and so on. With the focus on ontology, this process is shown in Figure 1.

We propose the use of Generic Human Disease Ontology as the brain for agents of the multi-agent system to carry out the search activities. Agents interact with online users. For each online inquiry, the ontology provides a road map for agents to access and retrieve relevant information.

We differentiate 4 different agent's types within the GHDO-based MAS:

(1) *InterfaceAgent* constructs Specific Human Disease Ontology (SHDO) templates from GHDO. The SHDO template is sent to the ManagerAgent and to the SmartAgent.

The SHDO template corresponds to the user's query; it specifies information the user is interested in. During the whole information retrieval process, the collaborative working agents will fill in the SHDO template by the relevant information.

(2) *ManagerAgent* assigns tasks to the various InformationAgents. ManagerAgent must have the appropriate expertise to do this, namely, it must have knowledge of the task structure. This is specified by SHDO template.

(3) *InformationAgents* retrieve the requested information from a wide range of biomedical databases. Depending on the requested information, an agent may be required to retrieve information about disease types, symptoms, causes and/or treatments. The InformationAgents send the retrieved information to the SmartAgent.

(4) *SmartAgent* collects and analyzes the information received from InformationAgents, select the relevant information and assemble it according to the structure specified by the SHDO template. This step results in the Specific Human Disease Ontology (SHDO). The SHDO is sent to the Interface agent to be presented to the user as answer to his/her query.

A GHDO-based Multi-Agent System can be used as an information infrastructure to support worldwide scientific research on human diseases. This system can be used to provide answers to researchers or doctors about the latest scientific research results on any type of disease, the phenotype (the symptoms), disease causes and treatments.

IV. USING UML2.1 TO MODEL GHDO-BASED MAS

Each agent that we model is usually capable of playing many roles. Each role could correspond to an interface which could be implemented in a variety of ways. In our modeling paradigm we use a technique called method lifting [19] to form composite classes which implement interfaces which enable agents to play roles. A composite class is formed by a selection of methods being taken from a hierarchy and aggregated to provide an interface.

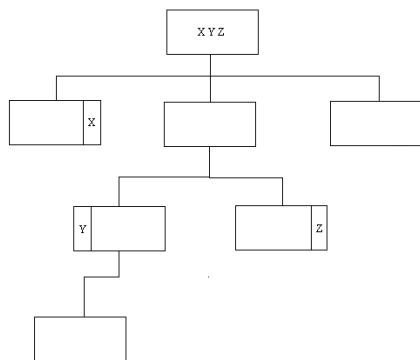


Figure 2: Method lifting

In the Figure 2, for example, the methods X, Y and Z all reside in distinct classes. Nevertheless, they can be

aggregated in the composite class at the head of the hierarchy to form an interface. Once we have formed the composite class and gotten the interface we have a basis to draw the UML 2.1 Sequence Diagram. Each composite class is represented by a rectangle at the head of the sequence diagram. It represents a distinct Agent. Each interface of the composite class is represented by a port and lifeline down the sequence diagram. It represents a distinct role played by the Agent. That is, a single rectangle (Agent) could have many ports (Roles) without changing the semantics of the UML 2.1 Sequence Diagram. That meets our goal of modeling Agents using UML 2.1 through the use of method lifting.

A. Modeling the Sociable Nature of GHDO-based MAS Agents

A Sequence Diagram is generally defined across the page by a series of rectangles, each of which represents a class. Each of these rectangles has a dotted line running vertically down the page. These dotted lines are known as lifelines. As you go down the page, time passes as messages flow between objects. UML 2.1 allows for a particular class to have more than one lifeline. Namely, a particular class may have many ports, each one with its own lifeline. The agent may be represented by a rectangle, and have many ports, each with its own lifeline.

We will use a Sequence Diagram where Composite Classes have more than one port and represent different roles of the same agent. Hanish & Dillon [17] have previously used a similar and related approach to represent an Agent/Class playing different roles. This will enable us to model GHDO-based MAS and represent agents which play more than one role concurrently. For example, SmartAgent plays 2 different roles: collection of information from various InformationAgents and sending SHDO to the InterfaceAgent. Each port has its own lifeline. If there are two ports, this signifies two roles that are played by the agent from which the ports come. We use Composite Class as a rectangle at the head of lifelines in a Sequence Diagram, and each port to represent a role played by the Composite Class, rather than repeating rectangles for each class. In the examples shown in Figure 3, a number of agents play multiple roles which is represented by multiple ports. Depending on which role the agent is acting in when it sends/receives messages, the sequence diagram shows arrows to/from a particular lifeline for the agent.

In the example shown in Figure 3, we illustrate a system that has four agents. The first agent is called InterfaceAgent. The user initially specifies the disease that they want information about by sending a 'Specify Disease' message to the InterfaceAgent. The agent confirms that this message has been received. The next step is that the user specifies their specific interest in the disease. That is, do they want information about disease types, causes, symptoms or treatments? Again, the InterfaceAgent confirms that this message has been received. Once the InterfaceAgent builds a SHDO template (called a Specific Human Disease Ontology Template) using Generic Human Disease Ontology, there is a

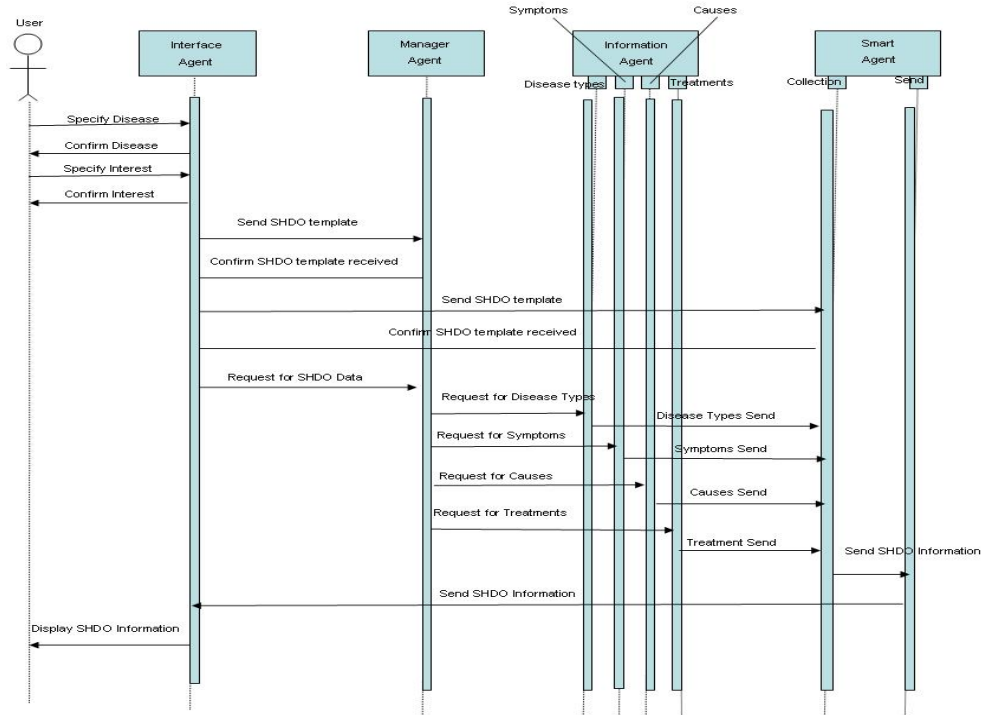


Figure 3. Sequence diagram representing sequence of processes within GHDO-based MAS

structure established which can later be filled out with data. For example, the user may specify that they want information about Diabetes, and only on causes and symptoms, not treatments.

The SHDO template is immediately sent to the ManagerAgent, whose responsibility is to direct requests to InformationAgents who actually retrieve the data required to fill in the template. Thus, the ManagerAgent needs to have the empty SHDO template, in order to know which InformationAgents to activate. The SHDO template is also sent to the SmartAgent. The responsibility of the SmartAgent is to collect (from InformationAgents), analyze, select relevant information and fill out the SHDO template with data returned.

Once the SHDO template has been sent by the InterfaceAgent, this agent makes a request to the ManagerAgent for Disease Data. The ManagerAgent then successively sends messages to an InformationAgent about Disease Type, Symptoms, Causes and Treatments. Each different type of information will require that the InformationAgent play a different role in order to gather it. Hence the messages go to one of four distinct ports, each of which models a distinct role for the InformationAgent.

Each time the InformationAgent gathers information it sends it to the SmartAgent, who is responsible for collecting the information in order to fill out the SHDO template. Once the SmartAgent has all the SHDO data collected and the

template has been filled out it plays a different role to send the information back to the InterfaceAgent. The InterfaceAgent then returns the data to the user.

There are three points worth noting in our sequence diagram:

- (1) the lifelines of agents are solid throughout since agents tend to be persistent
- (2) each rectangle represents a Composite Class which implements an agent type, and
- (3) each distinct role played by an agent is represented by a distinct port on the rectangle with its own lifeline.

In this section, we have illustrated sociable nature of GHDO-based MAS agents by representing inter-agent interactions. In the following section, we will model structure of the individual agents with respect to their goals.

B. Modeling Goal-driven Nature of GHDO-based MAS Agents

We can model the goal-driven aspect of the agent by a Composite Structure Diagram with Parts, and Ports. Each part represents a distinct area of processing within the agent. Each port represents a different role played by the agent.

The <<Agent>> stereotype based on the Composite Structure Diagram [18] can be used to model the SmartAgent. The <<Agent>> stereotype must have a name, at least a Controller part which controls the efforts of the Agent to achieve a goal, and at least one port, which relates to it's playing a role.

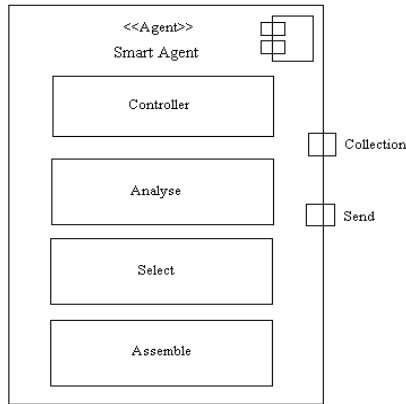


Figure 4. Composite Structure Diagram representing Goal-driven characteristic of the SmartAgent

We use a Composite Structure Diagram to represent the goal-driven nature of an agent. In the case of the SmartAgent shown in Figure 4, we have two ports which correspond to two different roles of this agent, and four parts which show distinct areas of information processing within the agent. It is important to note that the parts in the composite structure diagram in figure 5 can represent either objects or composite structure objects (created by method lifting as discussed earlier in this paper).

Note that the same two ports (Collection and Send) that were present in the sequence diagram are also present here. Each of the ports is a construct which enables the Agent to interact with other Agents, namely InformationAgent and InterfaceAgent. Next to the Controller object, the SmartAgent also has Analyze, Select and Assembly objects. The SmartAgent collect the retrieved information through the Collection port. This information is analyzed, the relevant information is selected and assembled into the SHDO template resulting in the SHDO. The SHDO is sent to the InterfaceAgent via Send port.

V. CONCLUSION

This paper uses UML2.1 to describe an *ontology-based multi-agent system* used for *intelligent retrieval of information about human diseases*. We have used the UML 2.1 Sequence Diagram to capture and represent the *sociable nature of agents*. In this, we have not changed the semantics of the Sequence Diagram which is a critical point that makes our use of UML 2.1 valid. The different *roles* of an agent have been represented by the different *ports*. These ports correspond with the ports of the UML 2.1 Composite Structure Diagram. The UML 2.1 Composite Structure Diagram uses also parts to capture and represent internal processes. The combination of ports which represent agent's roles, and *parts* which capture agent's *internal processes* enable effective representation of the *goal-driven nature of agents*.

REFERENCES

- [1] M. Hadzic and E. Chang, "Ontology-based Multi-agent Systems Support Human Disease Study and Control", Edited by Czap, H., Unland, R., Branki, C., Tianfield, H., in *Frontiers in Artificial Intelligence and Applications (special issues on Self-organization and Autonomic Informatics)*, vol. 135, IOS Press, 2005, pp. 129-141.
- [2] T. Dillon, E. Chang, M. Hadzic and P. Wongthongtham, "Differentiating Conceptual Modelling from Data Modelling, Knowledge Modelling and Ontology Modelling and a Notation for Ontology Modelling", Invited Keynote at the *Fifth Asia-Pacific Conference on Conceptual Modelling (APCCM 2008)*, Australia, 2008.
- [3] V. T. da Silva, R. C. Noya, C. J. P. de Lucena, "Using the UML 2.0 Activity Diagram to Model Agent Plans and Actions", *Proceedings of the 4th International Conference on Autonomous Agents and Multiagent Systems*, Netherlands, 2005.
- [4] OMG Group, "UML version 2.1.2", <http://www.omg.org/spec/UML/2.1.2/>.
- [5] A. Moreno, D. Isern, "A first step towards providing health-care agent-based services to mobile users", *Proceedings of the First International Joint Conference on Autonomous Agents and Multi-agent Systems*, pp. 589-590, 2002.
- [6] J. Huang, N.R. Jennings, J. Fox, "An Agent-based Approach to Health Care Management", *International Journal of Applied Artificial Intelligence*, vol. 9, no. 4, pp. 401-420, 1995.
- [7] J.M. Fonseca, A.D. Mora, A.C. Marques, "MAMIS – A Multi-Agent Medical Information System", *Proceedings of IASTED International Conference on Biomedical Engineering (BioMED 2005)*, 2005.
- [8] E. Merelli, R. Culmone, L. Mariani, "BioAgent-A mobile agent system for bioscientists", *Proceedings of the Network Tools and Applications in Biology Workshop Agents in Bioinformatics*, 2002.
- [9] M. Ulieru, "Internet-enabled soft computing holarchies for e-Health applications", *New Directions in Enhancing the Power of the Internet*, pp. 131-166, 2003.
- [10] P. Srinivasan, J. Mitchell, O. Bodenreider, G. Pant, F. Menczer, "Web Crawling Agents for Retrieving Biomedical Information", *Proceedings of International Workshop on Agents in Bioinformatics (NETTAB-02)*, 2002.
- [11] M. Hadzic and E. Chang, "Medical Ontologies to Support Human Disease Research and Control", *International Journal of Web and Grid Service*, vol.1, no.2, pp. 139-150, 2005.
- [12] K. Kavi, D. C. Kung, H. Bhamhani, G. Pancholi, M. Kanikarla, "Extending UML for Modeling and Design of MultiAgent Systems", *Proceedings of the 2nd International Workshop on Software Engineering for Large-Scale Multi-Agent Systems*, USA, 2003.
- [13] V. T. da Silva, R. C. Noya, C. J. P. de Lucena, "Using the MAS-ML to model a multi-agent system", *Software engineering for multi-agent systems II : research issues and practical applications*, Lecture Notes in Computer Sciences vol. 2940, 2004, pp. 129-148.
- [14] B. A. de Maria, V. T. da Silva, R. C. Noya, C. J. P. de Lucena, "VisualAgent: A Software Development Environment for Multi-Agent Systems", *Proceedings of the 20th Brazilian Symposium on Databases and 19th Brazilian Symposium on Software Engineering*, Brazil, 2005.
- [15] J. Odell, H. P. Van Dyke and B. Bauer, "Representing Agent Interaction Protocols in UML", in P. Ciancarini, M. Wooldridge (eds), *Agent-Oriented Software Engineering*, Springer-Verlag, 2001, pp. 121-140.
- [16] M. Fernández, A. Gómez-Pérez, and N. Juristo, "METHONTOLOGY: From Ontological Art Towards Ontological Engineering", *Proceedings of AAAI'97 Spring Symposium Series, Workshop on Ontological Engineering*, pp. 33-40, 1997.
- [17] A. A. Hanish and T.S. Dillon, "Object-oriented behavior modeling for real-time design", *Proceedings of the 3rd International Workshop on Object-Oriented Real-Time Dependable Systems (WORDS '97)*, USA, 1997.
- [18] D.S. Dillon, T.S. Dillon, and E. Chang, "Using UML 2.1 to model Multi-Agent Systems", *Proceedings of the 6th IFIP Workshop on Software Technologies for Future Embedded and Ubiquitous Systems*, Italy, 2008.
- [19] X. Lu, T.S. Dillon, E. Chang, A. Talevski, and W. Gardner, "Automatic Method Lifting in Composite Object", *International Journal Of Engineering Intelligent Systems*, vol. 10, no. 3, pp. 131-139, 2002.