Agent-oriented Modeling for Collaborative Learning Environments: A Peer-to-Peer Helpdesk Case Study

Renata S. S. Guizzardi[1], Gerd Wagner[2] and Lora Aroyo[1]

[1]Computer Science Department – University of Twente (UT)

P.O. Box 217 NL – 7500 AE – Enschede – The Netherlands

{souza, aroyo}@cs.utwente.nl

[2]Faculty of Technology Management, I&T – Eindhoven University of Technology (TU-E)

P.O. Box 513 – 5600 MB – Eindhoven – The Netherlands

G.Wagner@tm.tue.nl

Abstract. In this paper, we present the analysis and modelling of Help&Learn, an agent-based peer-to-peer helpdesk system to support extra-class interactions among students and teachers. Help&Learn expands the student's possibility of solving problems, getting involved in a cooperative learning experience that transcends the limits of classrooms. To model Help&Learn, we have used Agent-Object-Relationship Modeling Language (AORML), an UML extension for agent-oriented information systems modeling. The aim of this research is two-fold. On one hand, we aim at modeling the variety of roles and the complexity of their interactions and activities within the Help&Learn system. On the other hand, we aim at showing the expressive power and the modeling strengths of AORML.

Keywords: collaborative learning environments, peer-to-peer technology, agent-oriented modeling

1. Introduction

The WWW has significant impact on education, by functioning as a gateway for information available worldwide and educational interaction with global dimensions. These changes have direct impact on both class-related activities and extra-class interactions. Various research exemplifies the benefits of the Internet to support students in extra-class discussions [1, 2, 3, 4]. These initiatives explore mainly a non-hierarchical relationship between teachers and students, in an environment where everyone can teach and learn.

From an implementation perspective, this requires a flexible approach to support the organization and coordination of people and resources within such distributed environments. Currently, the application of software agents as personal assistants and as monitors and coordinators within peer-to-peer (P2P) knowledge sharing spaces, appears to be a suitable solution. The fact that in P2P there is no central server, i.e. each node of the network can be, at the same time, client and server [5], reflects in the non-existence of a central power or any kind of authority controlling the peers interactions and exchanges. This way, teachers, students, experts and organizations can be seen as providers of information and owners of knowledge, gathered in self-organized communities, where agents play a crucial role in the coordination of the processes.

From a software engineering perspective, the analysis and design of the distributed processes become increasingly sophisticated and require an agent-specific view provided by an appropriate methodology for agent-oriented analysis and design.

In this paper, we gather these two perspectives and propose an agent-based P2P helpdesk system, called Help&Learn, to support extra-class discussions between students and teachers. Help&Learn expands the student's possibility of solving their doubts, getting involved in a cooperative learning experience that transcends the limits of classrooms. By collaborating with other peers, the students learn with the doubts of others, besides developing cognitive abilities, such as to state clearly their doubts and thoughts; to interpret questions; to mediate discussions; and to solve problems. We use the metaphor of a helpdesk, where somebody asks for help (*the helpee*) and somebody provides the needed help (*the helper*). The interesting point here is that the helpee can stay anonymous and it also does not matter where the help comes from, as long as his problem is solved.

As in a typical peer-to-peer application, the key issue in Help&Learn is finding the best peer to satisfy a certain help request. A helper is selected if he can fulfill a help request, by providing a *what-is* or *how-to-do* explanation, a bibliographic or a WWW references on the subject, etc. Besides expertise, the time and availability of the peer are also considered when searching for the most appropriate one. A teacher may know the answer to a student's question but he may have less time than an advanced student to spend on it. In this process, agents are used to create and maintain the user's profiles, to collaborate in the search for the best peer, and to organize the available material.

To model Help&Learn, we have used AORML, Agent-Object-Relationship Modeling Language [6], an UML extension for agent-oriented information systems modeling. The Agent-Object-Relationship (AOR) modeling approach [6] defines the fundamental components of semi or fully automated social interaction processes to explicitly help achieving coherent behavior. The main concepts here are agent, event, action, claim, commitment, and object. Agent and object form, respectively, the active and passive entities, while actions and events are the dynamic entities of the system model. Commitments and claims establish a special type of relationship between agents. An agent can be human, artificial or institutional. Institutional agents are usually composed of a number of human, artificial, or other institutional agents that act on its behalf. Organizations, such as companies, government institutions and universities are modeled as institutional agents allowing to model the rights and duties of their internal agents.

Thus, on one hand, we aim at modeling the variety of roles and the complexity of their interactions and activities within such collaborative environments. On the other hand, we aim at showing the expressive power and the modeling strengths of AORML.

Further in Section 2 we present the modeling process of the Help&Learn system and we conclude in Section 3 with some directions for future work.

2. Help&Learn Modeling

AORML has been used through the whole development process of Help&Learn, starting with domain modeling (supporting requirements specification) and proceeding with analysis and design phases. The analysis starts with defining AOR external models, which adopt the perspective of an external observer of the (prototypical) agents and their interactions in the P2P helpdesk domain. Figure 1 depicts the agent diagram, which includes all human, artificial and institutional agents

(distinguished by UML stereotypes) involved in the helpdesk and their relationships. Note that this diagram is very similar to the UML class diagram, showing the system's classes and relationships between them. For clarity purposes, the attributes of agents and objects are omitted in this diagram. They can be however expressed following the traditional UML syntax.

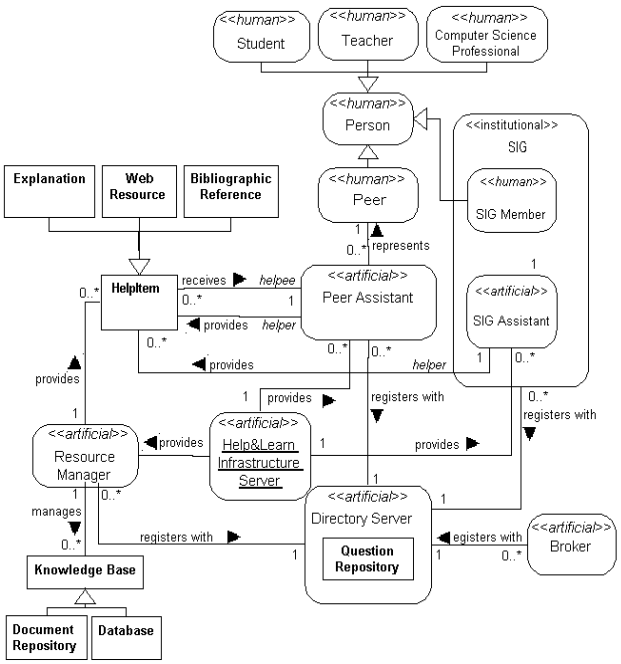

Figure 1 – Helpdesk System Agent Diagram

In order to start participating on discussions in the system, a Person downloads the Peer Assistant (PA) from the Help&Learn Infrastructure Server. This way, this Person becomes one of the system Peers, being able to act both as a *helpee* and as a *helper* for other Peers. Every time the PA goes online, it registers with the Directory Server (DS), becoming available to answer help requests.

When registering with the DS, the PA will provide a minimal Peer profile, indicating what topics can be answered by him. On the other hand, the Broker creates his own Peer profile by contacting the PAs and also by applying data mining techniques on the DS profiles, in order to make rankings and classifications. The Broker ranks the Peers based on expertise, availability and reliability and it classifies them based on interests. This way, when queried by the PAs, it can provide information on the most appropriate Peers to answer a certain help request.

Special Interest Groups (SIGs) are also allowed to participate in the system. These SIGs usually pre-exist the system, but can also be created by suggestion of the Broker. It is not necessary that all the members of a SIG are Peers, only one member is enough. The Broker has a representation of the SIGs and can also suggest a PA contacts one of the SIG Assistants in order to ask the SIG for help. The SIG Assistant broadcasts the message to all Peer members of the SIG. Then, the answers are sent back to the PA. Today, there are many SIGs advertised in the Web, specialized in several different areas. By introducing them to the helpdesk system, we hope to broaden their interaction scope, at the same time that we give the opportunity for other Peers to have their help request answered by an expert on the topic. The Resource Manager brings, to the system, existing knowledge bases, which can be databases, document repositories etc. This way, help items that are not owned by any of the system Peers can also be considered and consulted by the PAs.

The next step after defining the agents in the system, is to model their interactions using Interaction Sequence Diagrams (ISDs) for concrete examples as shown in Figure 2. Here, Anna, a system Peer, issues a request for help to her PA, asking "what is p2p?". First, the PA tries to find out if this question has been asked before, by querying the DS that maintains a Question Repository (see Fig.1). Since this question is asked for the first time, the PA cannot provide a direct answer and asks the Broker to find the best Peer to respond to the question. The Broker returns a ranked list of possible Peers for the PA to select. Then, Anna's PA contacts the first on the list (Mark's PA) with the request for help. Mark's PA will, then, forward Mark the request for help and Mark provides the following answer: "p2p is a distributed technology…", which is forwarded by his PA to Anna's PA and finally gets to Anna.

Note, that this ISD shows just one of many possible interactions. The system developer makes a number of ISDs in order to capture all interaction perspectives. This way he can afterwards generalize the interaction between two agents in Interaction Frame Diagrams (IFDs). Further, the interactions can be detailed in Interaction Patterns Diagrams. These diagrams depict general interaction patterns expressed by means of a set of reaction rules defining an interaction process type. Reaction rules are the chosen component by AOR to show the agent's reactive behavior and they can be represented both graphically and textually.

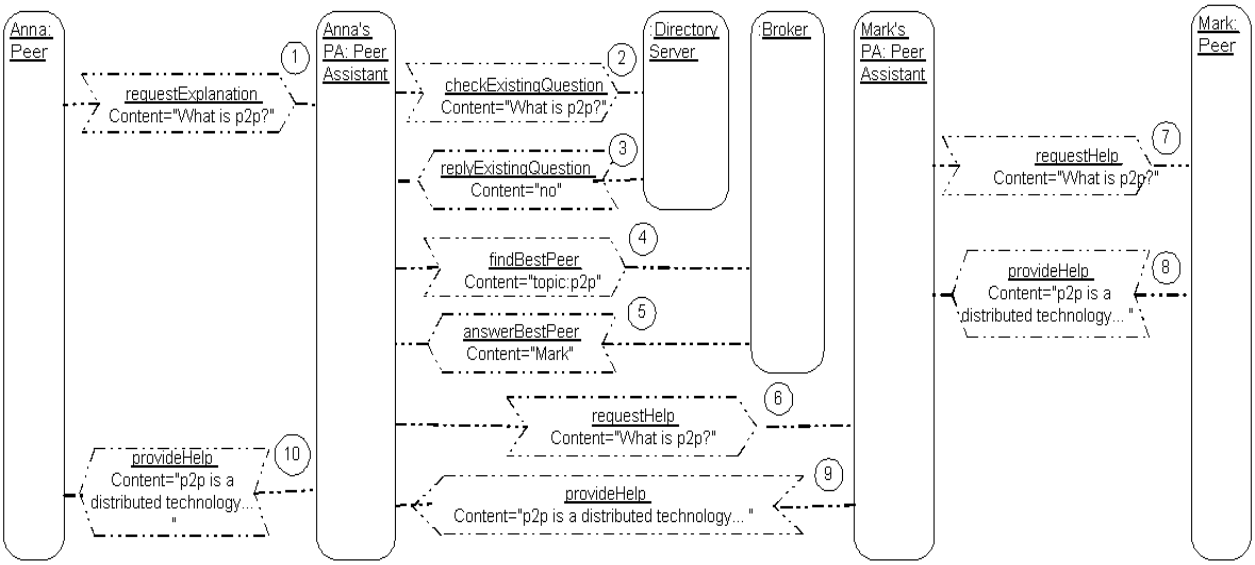

Figure 2 – Interaction Sequence Diagram

After all these steps, the modeling can proceed to the design stage, in which, for each type of agent system to be designed, the external model is *internalized* according to the perspective of the respective agent, and subsequently further refined. For instance, an *action event*, if created by the agent to be designed, is turned into an *action*, while it is turned into an *event* if it is perceived by it. Using such an internal perspective and the corresponding indexical terms (such as *actions* and *outgoing messages* versus *events* and *incoming messages*), leads to a natural terminology for designing and implementing agents.

3. Conclusions and Future Work

Help&Learn provides students with a rich collaborative environment for extra-class learning interactions. It consists of a self-organizing peer-to-peer network augmented with two special help provider nodes: knowledge resource managers and special interest group assistants. We take an agent-based perspective on system architecture, where agents play a crucial role in supporting the effectiveness, flexibility and personalization of the whole process. Following, we apply an agent-oriented modeling approach illustrated with the means of AORML, which proved to be very efficient. The fact that it is an extension of UML makes it quite accessible. Besides, the mentalistic concepts of AORML, such as beliefs and commitments, are powerful abstractions for agent-based modeling. The experimentation with AORML has also provided us with some feedback on how this language can be extended, adding new constructs to facilitate agent-based modeling. The further plan is to implement and test the proposed system design.

Acknowledgement: This paper has been written in the context of Agent Academy: A Data Mining Framework for Training Intelligent Agents, a project funded by the European Union 5th Framework IST Programme.

References

[1] Funaro, G. M. (1999). Pedagogical Roles and Implementation Guidelines for Online Communication Tools. ALN Magazine, 3 (dez).

[2] Evard, M. (1994). Articulation of Design Issues: Learning through Exchanging Questions and Answers. In: National Educational Computing Conference, Boston-MA, USA.

[3] Souza R., & Menezes, C. (1998). Usando Situações Reais de Comunicação na Aprendizagem de Língua Estrangeira: Proposta de um Ambiente na Internet. In proc. of SBIE'98 Fortaleza/CE, Brazil.

[4] Vassileva, J. et al. (1999) A Multi-agent Approach to the Design of Peer Help Environments. In proc. of AIED'99, Le Mans, France.

[5] Olson, L. (2002). NET P2P: Writing Peer-to-Peer Networked Apps with the Microsoft .NET Framework. Available at: http://msdn.microsoft.com/msdnmag/issues/01/02/netpeers/netpeers.asp

[6] Wagner, G. (2002) The Agent-Object-Relationship Meta-Model: Towards a Unified Conceptual

View of State and Behavior. *Information Systems*, 2002/2003. (to appear). Available at: http://AOR.rezearch.info/