Work in Progress - A Distributed Approach to a Learning Management System using Multi-Agent Technology

Rodrigo Filev Maia, Marcio Lobo Netto
Escola Politécnica, University of São Paulo, Laboratory of Integrated Systems,
Av. Prof. Luciano Gualberto, 158 trav. 3 - São Paulo, {rodrigo.maia, marcio.netto}@poli.usp.br

Abstract - The agent technology has in the autonomy of a single agent one of its most important features. The collaboration between agents, inference and learning capabilities are other interesting features. All these issues seem to be useful to represent students in an electronic system, because they permit the creation of virtual tutors that could follow the steps and the evolution of the students. It is a way to improve the interaction between students, teachers and professors. The agent technology could also be applied to create distributed Learning Management Systems (LMS). With some artificial intelligence methods it would be possible to build a distributed Intelligent Tutoring System (ITS). This paper presents two possible architectures of distributed LMS: one using a LMS to store the knowledge (such as a domain module of an ITS) and agents representing the tutor and student modules of an ITS. In the second approach, there is no LMS and all ITS functions are distributed between agents. This paper discusses the benefits of these approaches and how agents can provide help for the students in these architectures.

Index Terms – Agent Technology, Distance Education, Multi-Agent System (MAS), Student Support, Learning Management System (LMS), Intelligent Tutoring System (ITS).

INTRODUCTION

The Distance Education is being considered for many as a possible solution for several problems in education such as a way to deliver education to distance places or a way to reduce costs in corporative education [1]. Although different types of technologies are being used to create e-learning environments, few of them consider the particular aspects of each student. A bigger set of information about all the student learning activities is important for a correct understanding of the grades in exams, and a way to detect the changes in the students’ behavior and development [2].

It is also possible that the students and professionals are not under only one ITS or LMS during their studies. One of the disadvantages of this scenario is having more than one student model for the same person. As consequence the information about the student behavior is incomplete in all models compared with one model that could contain all these data. An integration of the related information of those systems could promote a more accurate student model and provide a better understanding of his behavior, which could be useful for student’s diagnosis.

DISTRIBUTED ITS OR LMS USING A MULTI-AGENT SYSTEM

For this paper, the most important features of a software agent are: autonomy, cooperation and learning capabilities. These characteristics permit agents to execute similar functions provided by a traditional ITS, such as to create a student model, to show the materials of a course or class in a correct order, or even to implement appropriate educational strategies to provide information for the students. [3],[4],[5]. As the agent technologies have proper features for implementing educational software, there are at least two possible approaches considering the MAS acting as educational learning environments. The first approach uses agents with a traditional LMS (or ITS). The second one considers all functions of an ITS or LMS completely distributed between agents. There are two common agents for both architectures: one represents the student (it is his tutor) and the other one represents the professor. The tutor agent must have some method to accomplish and diagnosis the student’s performance such as Bayesian networks or other appropriate heuristic methods. The same is valid to the professor’s agent that is responsible to analyze information that came from all tutor agents and provides reports that contains the diagnosis for the class (or for each student). It aims to identify common students’ difficulties or points of the course that is not adequate to the class.

I. First Architecture: Agents and LMS (or ITS) systems

In an educational environment provided by a LMS, there is a central repository of information, which stores all data about courses, and materials to be accessed by students. It corresponds to the domain module of an ITS1. The LMS could also store the strategies of learning as part of a tutor module. The first agent of this scenario is called Student Agent (SA), which is responsible for student and interface modules of an ITS. Its tasks are: capture and store all information about

---

1 The authors of this paper consider the classical model of an ITS: domain, tutor, student (representing the student model) and interface modules.
student and provide resources for the interaction between the student and all other components of the environment. The desired effect of this agent is the improvement of the student’s motivation and performance during his studies.

The Professor Agent (PA) is responsible for the tutor module in order to select the educational strategies for a particular student. PA also provides an interface for the professor (or teacher).

These two agents are enough to students and professors develop their academic works; but it is necessary other agents responsible for providing an operational environment for all agents, such as an interface to the LMS. For this task, other two agents were designed: Manager Agent (MA) and Database Agent (DA). The MA provides a standard interface for any LMS and controls the initial communication between the other agents of the system. The DA is specialized in accessing and manipulating databases controlled by LMS or ITS as in independent storage systems.

II. Second Architecture: LMS composed exclusively by agents

This approach eliminates both LMS and ITS. The immediate consequence is the complete distribution of content of a course and an improvement in the complexity of communication between agents. In this scenario, the SA comprises the student, interface and part of tutor modules. The SA is responsible for selecting educational strategies to be applied for a specific student while he studies any course content. The PA has the domain, interface and part of tutor modules. For this agent, the tutor module registers and informs to the SA the desired educational strategies for each content. It represents the professor’s believes about how SA should conduct the course.

Although these agents represent all characteristics of an ITS or LMS, they are not enough for a functional multi-agent society. [6],[7],[8]. For this second approach it is necessary a third agent called School Manager (SM), responsible to coordinate all efforts of agents involved in educational activities, such as to provide an yellow page service, identifying all active agents in the MAS and their functions. SM must control the agents life cycles as well, in order to prevent deadlocks or lack of resources.

III Comparing the Both Approaches

The second approach has an evident disadvantage compared to first model due to the complexity of taking data about the course (or the students). It is a consequence of the complete distribution of the content between several PA, and due to more sophisticated searching mechanism when compared both to a traditional ITS and the first approach of the association between MAS and LMS.

However, the second approach eliminates the single point of failure represented by ITS or LMS. The most critical situation of failures in this architecture occurs when SM turns unreachable. In this case new communication channels could not be started. However all established communication between SA and PA will continue operating without interference. An advantage of both architectures is the location of student module near from learner, in order to collect information such as how many times the student accessed the material (even when he is offline), or the interval of time he does that and how much time he studies during each access.

CONCLUDING REMARKS AND FURTHER WORK

Both architectures presented in this paper consider a new way to provide access and assistance for students when they are far from their professors. The characteristics of the agents could create more precise student models, since more information would be available. This extra information could be used to support sophisticated heuristic methods in order to give better and personalized assistance for students.

Some tests are under execution and according to the preliminary results obtained by the simulation of the first proposed architecture, the communication between agents, works as expected, as well as the other issues of the agents. The heuristic model used by SA demonstrated to be a useful tool to identify which type of assistance is appropriate for each student. However, it is quite important to use the architecture proposed with real learners during a regular course, in order to identify how the students will react to an assistance provided by a virtual tutor. Actually a new prototype is under construction to be used by students in a distance learning course. In this new experiment, it is expected to identify if this accomplishment provided by software can really promote the increase of number of approved students and if can improve the quality of the education. Another desired result expected in this new experiment is increasing the number of students that finish the course, because a critical difficulty of distance education courses is the great number of students that give-up their studies before the end. One reason for that is the difficult of use the environments or the lack of assistance, and these questions could be solved using an agent as described in this paper.

REFERENCES


