Implementation of Intelligent Agents with Mobility in Educational Robotics Settings

Ilkka Jormanainen [1], Chiara Moroni [2], Yuejun Zhang [3], Kinshuk [3], Erkki Sutinen [1]
[1] Department of Computer Science, University of Joensuu, Finland
[2] Dipartimento di Ingegneria Informatica, Gestionale e dell'Automazione, Università Politecnica delle Marche (Polytechnic University of Marche), Italy
[3] Department of Information Systems, Massey University, New Zealand
ilkka.jormanainen@cs.joensuu.fi

Abstract

Teachers working in robotics classes face a major problem: how to keep track of individual students' or even small groups' progress in a class of 30-40 students. A multi-agent environment to help teachers with this problem is based on having pedagogical agents to monitor students' interaction, robots' movements, and the construction and programming process of robots. Mobile interaction agents move in network delivering agents' observations to teacher's visualization agent. Results of a linguistic analysis show the possible problems in educational robotics settings where the proposed agent-based system could help.

1. Introduction

In this article, we describe a prototype implementation of educational robotics environment which aims to support the teacher to focus his/her attention on potential problems in the educational robotics classes. The implementation is based on the concept described by Jormanainen et. al [3]. First, we have implemented intelligent agents which inhabit the IPPE programming environment and Lego robots' RCX units. Second, we have built a screen-based model of a classroom setting with the tools provided by an Empirical Modelling environment. The model works as a visualization agent in our system. Finally, we have implemented a mobile interaction agent, which has the ability to move from one computer to another, for example to report a learner problem observed by the other agents.

This article is organized as follows. Chapter 2 describes the basic concept of the environment as well as existing work in the field. Chapter 3 discusses the actual implementation of the system. Chapter 4 describes an observation of discursive interactions in an educational robotics setting, aimed at analysing what kinds of problems the participants had to handle and where the proposed agent-based system could help. Chapter 5 concludes the paper and draws some directions for the future work.

2. Background and related work

Jormanainen et. al [3] introduce an agent-based architecture for the educational robotics environment that helps the teacher to intervene into small groups' work. Unlike most current pedagogical agents which are usually applied to computerized learning environments or simulated virtual realities, the agency technologies described in [3] and in this paper are used in a real physical traditional classroom, in which students are working on a Lego project. The idea of using agency technology to monitor learning progress and providing learner help in a robotics classroom setting is not new. Previous work can be found in [1], where a multi-agent framework for distance support via the Internet in educational robotics is described. Advantages offered by the agency approach in educational robotics are summarized by George and Despres [1] as follows: the system and its functionalities can easily be distributed between tutor's workstation and learners' workstations. The modular aspect of multi-agent system ensures easy modification of an isolated agent's behavior and a multi-agent system provides an open architecture which allows for the integration of new agents if required. We follow this argument, believing that an agency approach is the right choice for monitoring pedagogical activities in the robotics classroom. Based on the previous work done by other researchers, we further develop an agency system by not only focusing on physical manipulations of the robot, but also controlling learners' programming activities and using mobile agents for asynchronous interactions as well as Empirical Modeling for visualization of observations.

3. Implementation

The implementation of the system contains two major tracks: pedagogical agents to observe the learning environment and a visualization agent for the teacher to observe the processes in the classroom. The system can contain several agents which all report their observations to one visualization agent. The teacher can interact with the visualization agent to observe the state of the learning setting and control the behavior of the pedagogical agents according to the needs rising from the current learning situation. In the next section, we will describe the implementation of the system more deeply.

In general, the system described in this paper is a multi-agent architecture including four agent modules. The student agent inhabits the IPPE programming environment [2] to monitor students' programming activities by reading data from the IPPE APIs on key strikes, mouse clicks, program errors and program types. The robot agent is seated in the Lego Mindstorms RCX unit to detect students' manipulations on the robot body by sensing signals from motors, sensors and buttons. The teacher agent makes decisions for the teacher based on the information reported from both the student agent and the robot agent, and implements visualization in which decisions made by the agent are presented to the teacher. The interaction agent works on the local network, facilitating the interaction among the agents.

The student agent collects data about student programming activities in the IPPE environment. The main functionalities of the student agent include testing programming idle time, detecting programming errors and monitoring programming types. The mission task described in certain specification language is received from the Empirical Modeling process. A new instance of the student agent will be created once a new mission is generated.

The robot agent monitors students' physical manipulation of the robot by collecting data from motors, sensors, and buttons. Its main functionalities include monitoring the idle time for manipulation of the robot, detecting motor movements, and checking sensor signals. This agent resides in the robot control unit and sends data via an IR receiver to an interaction agent running on the student computer. Ultimately, the interaction agent delivers the agent data to the teacher agent and the Empirical Modeling process. The execution of the agent is started when the user starts running the program. The agent runs in the thread independently, gathering data from the RCX unit and building reactions based on it. The implementation of the agency is transparent. This means that the user does not notice the agent running in the robotics system.

The teacher agent in the teacher's computer presents agent data to the teacher. In this way, the system assists the teacher to achieve a better understanding about the progress of the students. The agent implements a visualization engine which maintains the classroom model built with the Empirical Modelling (EM) [4] tools. The main reason to select EM as an implementation approach for teacher's agent rises from the nature of EM. The EM approach well supports a cyclic process and user's own observations about the phenomena. This means that teacher can update the model according his/her observations and, on the other hand, the teacher gets real-time feedback from the agents running in the environment. Due to the fact that EM tools have limited capabilities to communicate with the other applications, we will implement also a communication module as a part of the teacher agent. This part will be implemented as a Java application in order to achieve a full advantage with techniques such as RMI which takes care of the communication with interaction agents traveling in the local area network (see next section for more details). This communication module and the EM model will communicate through a local file system. The communication module passes agents' observations to the model as well teacher's definitions to the agents.

The interaction agent is responsible for coordination among agents in general, and data transfer in particular. It is an independent Java program, which receives data from *the sender agent*, then moves towards the node where *the receiver agent* resides and delivers the data locally to the receiver agent. When student activities are reported to the teacher, the sender agent is the student agent on a student computer, while the receiver agent is the teacher agent on the teacher computer. When a new mission task is generated and sent to a student computer, the situation is opposite. Each student computer as well as the teacher computer has an instance of the interaction agent running on it. We use Java RMI to implement the interaction agent. The Java RMI model allows an object running in one Java

Virtual Machine (VM) to invoke methods on an object running in another Java VM.

4. Evaluation

To examine the feasibility of the proposed agent-based solution, we have carried out an observation in a real teaching setting where educational robotics was used. The aim of the observation was to examine the possible problems in educational robotics settings where the proposed agent-based system could help. The analysis was therefore finalized to detect the critical moments occurred during the sessions and to identify the strategies used by participants to handle them. The observation was carried out during meetings of a technology club organised by Advanced Technology Learning Research Centre at Massey University, New Zealand. The club sessions were organized twice a week during the school semester at the premises of Massey University. There were 10 participants in the club sessions, aged between 10 to 40 years. The participants were divided into three groups, and the groups solved given technology-oriented tasks which contained planning, building, programming, and evaluating of a project with Lego Mindstorms robotics set. The participants came from diverse backgrounds having different amount of prior knowledge about building and programming.

The analysis allowed us to identify three main types of problems occurring in this kind of educational robotics setting: lack of interaction, lack of comprehension and conflict between different points of view. Furthermore, the observation shows how cooperation and negotiation strategies are essential to come through critical moments and to improve the problem solving process. Moreover, the observation of the discursive interaction in these educational environments enabled us to detect the problems that the proposed educational agents could be able to support properly. The results of this study indicate that, beside the functionalities described in this paper, the agents should be also able to monitor the interaction amongst the group. The agent implementation described in this paper does not answer to these needs. The proposed implementation is meant to observe students' actions with the programming environment and robots. However, the framework used to implement the system described in this paper, can be extended with such technology which allows monitoring for example dialogue between members of the groups. From that data, student agents could identify the problems in interaction amongst group members and build decisions suggesting collaborative and negotiation strategies.

5. Conclusions

In this paper, we have presented implementation principles of an agent-based educational robotics environment which is based on the description presented by Jormanainen et. al [3]. Proposed multi-agent architecture includes altogether four separate agent modules. We use the Lego Mindstorms educational robotics set including a small-scale computer (RCX unit) and the IPPE programming environment as host platforms for the agents. For implementation of agents, we use Empirical Modelling environment and Java programming language. We have conducted a linguistic analysis to examine the possible problems in educational robotics settings where the proposed agent-based system could help. Based on the results, we argue that the physical locations and activities of all objects in a learning environment, especially interactions among group members, are complicated and hard to model. Automating all the agency processes is difficult, if not impossible. However, proposed agent implementation is meant to observe students' actions with the programming environment and robots and it can be extended with technology which allows monitoring, for example dialogue between members of the groups. From that data, student agents could identify the problems in interaction amongst group members and build decisions in similar ways as described in this paper.

6. References

[1] George, S., Despres, C. (1999). A multi-agent system for distance support in educational robotics. *Proceedings of the International Conference on Telecommunication for Education and Training*, 344-353.

[2] Jormanainen I., Kannusmäki O., Sutinen E. (2002). IPPE - How to visualize programming with robots. *Proceedings of the Second Program Visualization Workshop*, University of Aarhus, Department of Computer Science, DAIMI PB - 567, 69-73.

[3] Jormanainen, I., Zhang, Y., Sutinen, E., & Kinshuk. (2006). Agency Architecture for Teacher Intervention in Robotics Classes. *Proceedings of the 6th IEEE International Conference on Advanced Learning Technologies (ICALT 2006)*. Los Alamitos, CA: IEEE Computer Society, 142-143.

[4] Roe, C. (2003). Computers for Learning: An Empirical Modelling perspective. Doctoral dissertation, Department of Computer Science, University of Warwick, UK.