2009 Ninth IEEE International Conference on Advanced Learning Technologies

Supporting Teacher Intervention in Unpredictable Learning Environments

Ilkka Jormanainen, Antony Harfield, Erkki Sutinen
Department of Computer Science and Statistics, University of Joensuu, Finland
{ilkka.jormanainen,antony.harfield,erkki.sutinen}@cs.joensuu.fi

Abstract

Modern teaching tools, such as educational robotics, require new learning environments. The teacher especially needs to be supported in novel ways. Conflative learning environment is based on the agents collecting data from the learning process. The teacher can build a support environment based on his or her empirical observations from the classroom. An implementation of the conflative learning environment is described and two case studies about the use of system are reported. The results show that agents are useful and efficient in data collection, and that the Empirical Modelling environment can be used to construct the working classroom models.

1. Introduction

Educational expectations from technology have shifted from management and usability of learning materials towards motivational issues, like how to engage an individual student with a topic. Recently, the low-cost and highly accessible educational robot kits have gained popularity in hands-on learning environments [6]. Technical subjects, such as engineering, electronics or programming, have been taught with educational robotics. However, efficient use of educational robotics requires new kinds of learning environments and teachers have to change their teaching methods according to the needs of the new environment. Otherwise, the use of educational robotics may lead to negative results, as reported in [2].

The open-ended nature of robot building can lead to students taking completely different approaches to an activity, and teacher's needs for the information about the learning process are difficult, if not impossible, to predict. Thus, the teacher might not be able to notice the need for intervention. Traditionally, intelligent tutoring systems (ITS) and expert systems have been used to guide students to achieve the learning goals, and it is possible to build a system to observe students' actions in the robotics classroom as well ([3]). However, these systems often provide only a pre-defined set of predictions about the problems that the student might face. Thus, in most cases a teacher needs to have advanced technical skills in order to make efficient use of the support system. In fact, he or she needs the skills and patience of a technical developer, and this is not frequently available in today's classrooms.

To support the teacher's working process in the learning environment where unpredictable learning activities often take place, we have proposed a concept of *conflative* learning environment [5]. The conflative learning environment gives full freedom for the teacher to modify the environment and support system to match the current situation. The learning environment conflates the roles of the teacher and the developer. This means that the teacher can adopt the developer's tasks into his or her work and build the environment gradually by modelling the empirical observations rising from the current classroom setting.

In this paper, we present two different studies that were conducted to examine the use of the conflative learning environment in the context of educational robotics. The aim of the studies was to analyse a) if the agent-based approach is suitable for data collection, and b) how the teacher can use the conflative learning environment to see if the students have understood the task-related rules that need to be learned in order to complete the task successfully.

We will first present the essentials of the conflative learning environment. The key elements of the CLE - rule-based learning, agents, and Empirical Modelling as a modelling tool - are described. Chapter 3 presents two case studies that were conducted to reveal the potential of the CLE. We will conclude the paper in Chapter 4.

2. Conflative learning environment and rule-based learning

Much of the learning in science and engineering disciplines is related to understanding a set of rules that

978-0-7695-3711-5/09 $25.00 © 2009 IEEE
DOI 10.1109/ICALT.2009.160

determine and regulate a certain physical phenomenon. However, these rules are usually difficult to expose during the learning process. In the case of educational robotics classes, for example, the following phenomena and sets of rules arise from the learning process:

- A learner needs to understand which components he or she can attach together to create a meaningful construction.
- A robot programmer needs to understand the syntax and semantic rules of a programming language.
- A teacher needs to understand the relationships between his or her learners within the working process.

In order to provide the teacher with a better understanding of the observations rising from the learning process, and to help the teacher to build better intervention strategies for his or her teaching, we have built a system to expose the data related to the rules [5]. The *conflative learning environment* (CLE) is based on agents observing students' actions with different components of the learning environment (Figure 1). The agents work in the environment based on a rule that the teacher has defined for them. The rule is related to one item in the learning environment, such as a button or another user interface element in the programming environment, or the program code, which the students are working with. The agent has a state, which can be *true* or *false*. When the agent changes its state according to the rule, it can execute a predefined action, e.g. sending a message or a command to the teacher's environment.

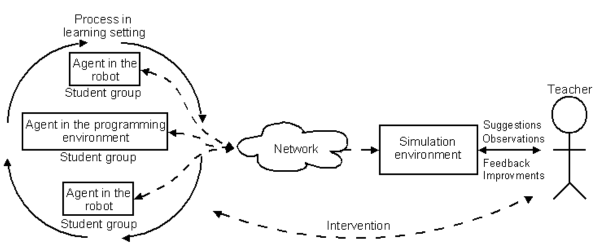

Figure 1. General architecture for conflative learning environment in the context.

An essential part of the conflative learning environment is *role conflation*. The CLE takes the teacher out from his or her traditional role of a technology user and mixes teaching activities with the activities of a technology developer. This means that the teacher can adopt developer's tasks to his or her work and build the environment gradually by modelling the empirical observations arising from the current classroom setting. The environment does not limit the ways in which the teacher can combine the observations, or how the observations are presented. Modelling is an essential part of the ongoing teaching process, and it can be done without interrupting teaching, as would happen within the traditional educational technology development process and when using traditional tools. The conflative learning environment allows the teacher to build a controlling and simulation environment which works in that particular situation, and which helps the teacher to provide the right support just in time and just in case, which is one of the essentials when making learning environments more supportive [4].

2.1 Empirical Modelling in the CLE

The conflative learning environment builds on the use of Empirical Modelling (EM), as the role conflation is one of the key elements of the EM approach [1]. EM is a collection of principles and tools that have been developed by Beynon, Russ and their students at the University of Warwick[1]. The construction of computer-based models using EM is based on the idea that the model is a collection of *observables* (that is, computational items that can be observed). The model is built by defining dependencies between the observables. Interaction between the user and model takes place by observing and manipulating these observables and dependencies.

The current Empirical Modelling tool uses several different notations, such as EDEN (Engine for DEfinitive Notations), and DoNaLD (Definitive Notation for Line Drawings). In the following example, the first three definitions create two EM observables and set the dependencies between them and observable `R3`. The next definitions after the `%donald` marker create and bring up the graphical interface with a circle (Figure 2). The last definition makes the colour of the circle dependent on the state of observable `R3`. Thus, the colour of the circle would change from red to green when both `R1` and `R2` are true.

```
%eden
R1 = true; R2 = false;
R3 is (R1 and R2);
%donald
circle groupA
label groupAName
groupA = circle({200,200},30)
groupAName = label("Group A", {280,180})
%eden
A_groupA is R3 ?
   "color=green,fill=solid" :
   "color=red,fill=solid";
```

[1] http://www.empiricalmodelling.com

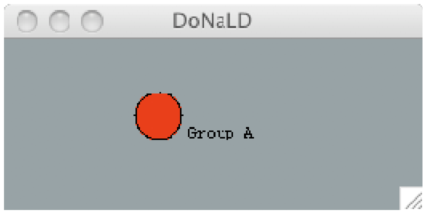

Figure 2. A simple EM model.

To explore the possibilities with the Empirical Modelling approach, a case study in a real classroom setting was conducted. In the study, we extended the primitive example model presented above toward a more complex model of the real classroom setting. The study is reported more deeply in the next section.

3. Case studies in CLE

To examine the use of CLE, two different case studies were conducted. First, we studied if the agent-based approach is suitable for data collection, and if it would be possible to follow students' activities and expose the possible moments of problems based on this data. Second, we wanted to see how the teacher could use the EM approach in the conflative learning environment to observe if the students have understood the task-related rules that need to be learned in order to complete the task.

3.1. Research setting and data collection

Both studies were carried out with a group of primary and secondary school children, aged between 10 – 15 years. The first study had eight participants (four groups), and the second study had four participants (two groups). The children were participating in the *Kids' Club* of the University of Joensuu. The Kids' Club is an after school technology club where students work together in various technology projects, such as designing, building, and programming robots for the RoboCupJunior tournament. The first study was conducted in May 2008, and the second study in January 2009. At the beginning of the studies, the following task was given to the children:

Build a simple wheeled robot with the Lego Mindstorms kit and equip it so that it can react to the collision. Then program the robot by using the IPPE programming environment so that it stops when it collides to an obstacle.

To solve the task, children must use a touch sensor in the robot and program the robot to react to the signals coming from the sensor. The children were working alone or in groups of two. Each group or individual child worked with a laptop computer and the IPPE programming environment (Figure 3). Manuals on how to build a simple wheeled robot and an unlimited number of Lego blocks for the robots construction were available.

3.1 First study: Agents as data collectors

During the first study (conducted in May 2008), four student groups worked in a robotics project as described above. Four agent instances were set to observe the use of the following buttons in the programming environment (Figure 3):

1. Add statement
2. Add command to code
3. Remove line
4. Send program to robot

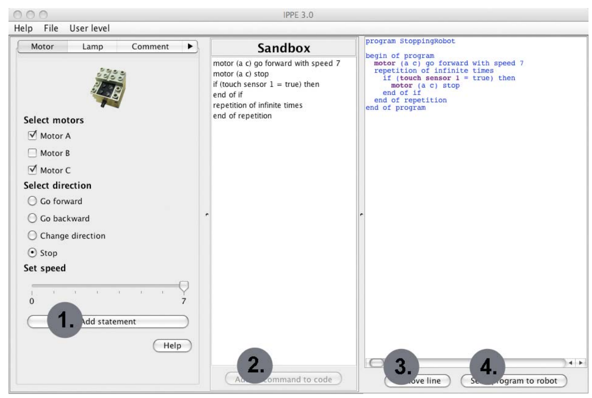

Figure 3. The IPPE programming environment.

Each agent was defined to send a message when a specified delay with the use of that particular button was exceeded. The delay was specific to each agent in a group and all groups had similar agents observing the activities. The delay for the agents was set between 30 seconds and 2 minutes depending on the element that the agent was observing. The delays were also adjusted during the session based on the teacher's judgement about the students' current progress. The data was stored in files and analysed afterwards with standard UNIX command line programs.

3.2 Results and analysis of the first study

According to the log files created by the agents, the student groups worked with the project for 37.32 minutes on average. During this time, each agent instance sent on average 34 messages / minute (Table 1). This was the first clear indication that the amount of data can be too large to handle manually, and there is a need to cluster and filter data according to the current activities in the classroom.

Table 1. Overview for the evaluation

Group	Time (minutes)	Messages	Messages / minute
Group A	19.74	686	35
Group B	36.55	815	22
Group C	55.67	2059	37
Group D	37.25	1535	41
Average	*37.32*	*1274*	*34*

To get a better view of the students' activities, a deeper analysis of the agent data was needed. Thus, we grouped the data according to the number of messages sent by the different agents (Figure 4). The graph in Figure 4 shows that the student groups had similar use patterns with the programming environment.

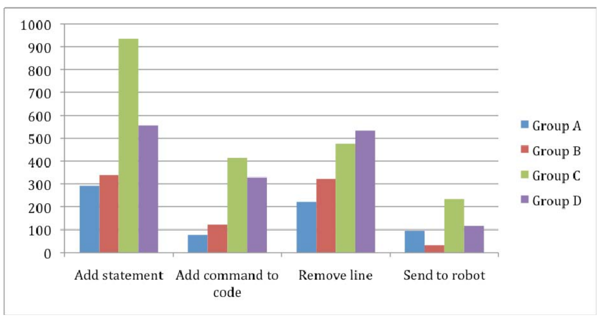

Figure 4. Number of the messages

The user events reported by the agents (Figure 5) confirm this phenomenon. The agent stopped sending messages after the corresponding button was pressed. Thus, the number of agent messages indicates how often students used that particular button in the programming environment. With this information, it is possible to expose patterns of use and students' typical workflows with the environment.

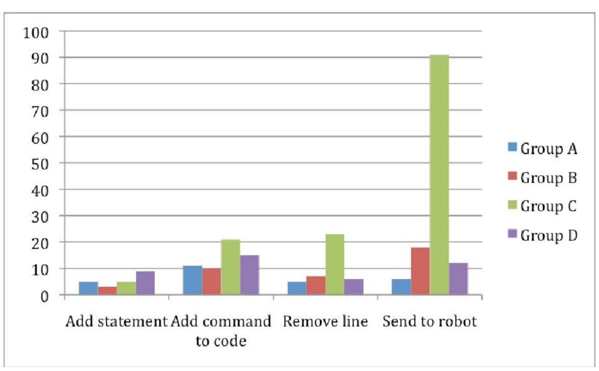

Figure 5. Number of the user events

Besides the overall workflow analysis, the agents can be used to expose unpredictable events in the classroom. Figure 5 shows that the agent observing Group C's behaviour with the "Send program to robot" button has reported a remarkably high number of user events. This most likely indicates a technical problem with the uploading process. However, presenting the data in this way *after* the classes does not provide very much for the "just in time" support. Thus, it would be important to see how the agents' observations could be utilised for a real-time support system. The second case study took a step toward this by using the Empirical Modelling tool as a platform for real-time observations.

3.3 Study 2: Empirical Modelling in the CLE

The aim of the second study was to explore the possibilities that the Empirical Modelling approach allows for building the teacher's controlling environment. Keeping in mind the example presented in Section 2.2, we built a model that shows when the students had completed the task with the robot. The model building process started by asking from the students how they have planned to solve the task. The groups agreed with each other that they would need a robot with a bumper and touch sensor, and a program for observing the sensor. Based on this, the first agent to observe the learning environment (existence of the touch sensor code) was modelled. After a few trials and errors, the students came to the conclusion that they would also need an event loop in order to observe the touch sensor constantly. This was well aligned with the assumptions about what the students would be required to solve the task. The event loop agent was also modelled, and these observables were combined together.

3.4 Results and analysis of the second study

As a result of this study, an EM model was developed (Figure 6). The model shows the progress of each student group as a circle (green, yellow, or red, depending on the state of that group). Two agents were defined to observe the essential parts of the code (existence of the touch sensor and the event loop). In Figure 6, "Group A" has got these essential elements ready, whereas "Group B" does not have either of these two elements included in their code. The agents delivered the observations to the teacher's computer and the EM model over the local area network. The observations made by the agents were totally transparent for the students, and agents didn't interfere with their work. The modelling was done on-the-fly while the children were working with their projects. The final result contains 30 lines of the EM definitions[2], and the following definitions formed the core of the model. The modelling process confirmed that the EM tools are usable when developing the conflative learning environments.

```
/* These will be updated by the agents */
R1_A; R2_A; R1_B; R2_B;

agent(100, 800, "Agent_A", "Group A");
agent(150, 700, "Agent_B", "Group B");

A_Agent_A is (R1_A and R2_A) ?
 "color=green,fill=solid" : R1_A ?
 "color=yellow,fill=solid" : R2_A ?
 "color=yellow,fill=solid" :
 "color=red,fill=solid";

A_Agent_B is (R1_B and R2_B) ?
 "color=green,fill=solid" : R1_B ?
 "color=yellow,fill=solid" : R2_B ?
 "color=yellow,fill=solid" :
 "color=red,fill=solid";
```

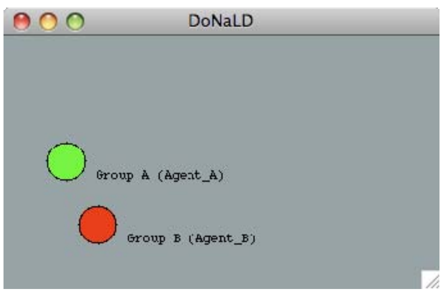

Figure 6. The model from the second study.

4. Discussion

We have presented the concept of a conflative learning environment, which allows the teacher to step out of his or her traditional role and adopt a software developer's tasks as a part of the teaching process. We have presented two case studies, which aimed to explore a) the usefulness of the agent approach in data collection, and b) how the teacher can use the conflative learning environment to see if the students have understood the task-related rules that need to be learned in order to complete the task successfully. The results of the studies show that agents are useful and efficient in data collection, but the amount of data is too massive for manual processing. Also, the analysis, which takes place after the classroom, does not serve well the need for just-in-time support. The second study indicates that the Empirical Modelling environment can be used to construct the working classroom models with a relatively small amount of code, and the results are available immediately without long software development process.

The examples and results presented in this paper can be continued gradually towards a more complex presentation of the classroom. It is possible, for example, to combine the representations of the results discussed in this paper to one, more comprehensive classroom model that shows students' progress in many ways. The teacher can add a graph to follow the students' actions in a way similar to the data analysis of the first study (the observations would be, obviously, real-time whereas in the study the analysis was done afterwards). Furthermore, an element to list the agent messages could be added, as the screenshot of a preliminary model in Figure 7 shows.

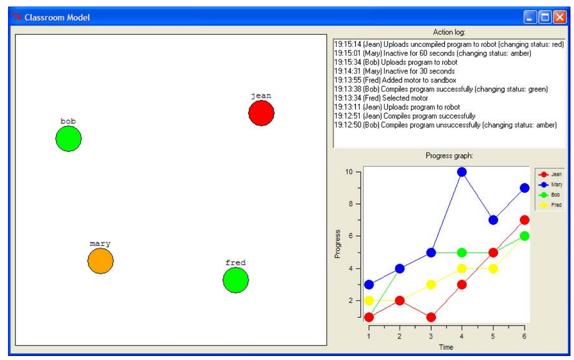

Figure 7. A comprehensives classroom model.

5. References

[1] W. M. Beynon, and A. Harfield, "Empirical Modelling in Support of Constructionist Learning: a Case Study from Relational Database Theory." The 5th IEEE International Conference on Advanced Learning Technologies, 2005, pp. 396–398.

[2] S.B. Fagin, and L. Merkle, Quantitative analysis of the effects of robots on introductory Computer Science education. *Journal on Educational Resources in Computing 2 (4),* 2002.

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

[4] J. Greer, "Learner Support: Convergence of Technology andDivergence of Thinking", ITS 2006 (LNCS vol. 4053), 2006, pp. 815–816.

[5] I. Jormanainen, and A. Harfield, "Supporting the teacher in educational robotics classes: work in progress", The 16th International Conference on Computers in Education, 2008, pp. 931–934.

[6] E. Sklar, S. Parsons, and P. Stone, "Using RoboCup in University-Level Computer Science Education", *Journal on Educational Resources in Computing 4 (2),* 2004, pp. 1–21.

[2] The complete Empirical Modelling script available in the author's web page: http://cs.joensuu.fi/~ijorma/ICALT2009/emscript.txt