Understanding open learning processes in a robotics class

Ilkka Jormanainen
University of Joensuu
Departmenf of Computer Science
P.O. Box 111
FI-80101 Joensuu, FINLAND
ijorma@cs.joensuu.fi

Meurig Beynon
University of Warwick
Department of Computer Science
Coventry CV4 7AL, UK
wmb@dcs.warwick.ac.uk

Erkki Sutinen
University of Joensuu
Departmenf of Computer Science
P.O. Box 111
FI-80101 Joensuu, FINLAND
sutinen@cs.joensuu.fi

ABSTRACT
Robotics is a functional approach for learning basic concepts of computing. In this role, it has been successfully used in introductory classes of Computer Science and Information Technology from primary to higher education. In a typical scenario, learners design and program robots in small groups that comprise between 2 and 4 students. Although such activity is stimulating for learners, the teacher of the class may find it hard to follow each individual student's learning processes. Empirical Modelling gives the teacher a platform to monitor individual group processes by collecting data from the construction and programming of the robots and allowing the teacher to model the empirically observed process. Unlike most adaptive learning systems, the model and the modelling process are transparent and open to the teacher, and even the students are able to assess their own learning based on the derived models.

Categories and Subject Descriptors
K.3.3 [**Computers and education**]: Computer uses in education – *Computer-assisted instruction (CAI).*

General Terms
Design, Human Factors

Keywords
Educational robotics, student modelling, Empirical Modelling, intervention, agency.

1. INTRODUCTION
Educational robotics has become a recognised tool for teaching at different school levels from kindergarten to university. The diversity of disciplines to which educational robotics has been applied is wide, and educational robotics is also a recognised part of computer science curricula [6]. The usual work process with educational robotics is based on group oriented working methods and open-ended problem solving. This readily leads students to take different paths to solving their problems, and groups may progress differently within a cycle of planning, building, programming, and testing. A robotics classroom might have 30 to 40 students divided into groups of 2 to 4 students. The unpredictable problem solving strategies and multiple student groups quite often cause the teacher to face difficulties in identifying the appropriate points for intervention. We are addressing this problem by utilising a system based on a multi-agent architecture [4] to support a teacher's observation process in the classroom. The agents can observe, for example, the students' construction and programming processes, as well as the teamwork and dynamics within and between the groups.

In this paper, we present an application for supporting a teacher in an educational robotics class. Based on the concept of conflative learning environment [4], we have built an environment for modelling the learning processes. The modelling is done with Empirical Modelling (EM) tools that allow open and transparent modelling of the learning process. The EM environment encourages role conflation, where a teacher can adopt a software developer's tasks in his or her own work and build in this way a support environment to match the current learning situation. The application allows the teacher and learners to build a model of the learning and group processes in a gradual way, based on the empirical data collected from the empirical observations arising from the current classroom setting.

Compared with traditional intelligent tutoring systems (ITS), the conflative learning environment framework provides a novel approach for the teacher to adapt the rules which form a base for modelling the students and learning processes. Instead of having predefined and static sets of rules, the teacher can construct the required rules from scratch by making use of logical operations to combine the atomic observations produced by the agents. Furthermore, the teacher can define what data should be collected, and how the data should be reflected to the model.

This paper is organised as follows. We first compare our approach to the previous work in the fields of ITS and adaptive systems. We then briefly describe Empirical Modelling. In section 4 we describe the conflative learning environment and a prototype application that we have built for deployment in educational robotics classes. Finally, we conclude the paper and sketch directions for future work in section 5.

2. BACKGROUND
Educational robotics has become a recognised tool in many disciplines and school levels, including computer science

education. In CS curricula, educational robotics has been used to teach both the basics of robotics as such and other computing concepts. Examples of the integration of robotics into the CS curriculum include for example teaching the Java byte code with the Lego robotics [3] and teaching systems-level programming topics by using the Lego robotics as a target platform [5].

Monitoring student groups' activities in the educational robotics classroom is difficult. Traditional intelligent tutoring systems have been applied also in this context [2]. However, as these systems are traditional programs with predefined specifications, they only offer the teacher a set of predefined options for interaction. The application based on the conflative learning environment framework provides support for open-ended exploration while having as its starting point the empirical observations of student activity. It is possible to support this kind of teaching process with traditional programming techniques and languages. However, where there are unpredictable scenarios, a single initial specification of a program is not enough to cater for all needs; the teacher also needs to have some degree of control over the development of the learning environment.

The traditional division of the roles in ITS and educational technology development processes usually strictly separates the roles of developer, teacher, and learner from each other. Moreover, the tasks undertaken by these process participants usually follow each other in a cycle with predefined steps. Beynon and Roe [1] argue that constructionist computer-assisted learning approaches can be seen as unifying the roles of the student, the teacher, and the developer. Following this line of argument, and invoking the concept of conflative learning environment described in [4], we can compare how the student and learning modelling process within the conflative learning environment framework differs from that associated with traditional ITS tools (Table 1). The main difference is that, whereas traditional ITS applications use a theory-based approach for building the learning model, the conflative approach starts from the empirical observations arising from the current learning situation. Another important aspect is that the EM based approach allows role conflation, and the tools are easier to adapt to different contexts and application areas.

Table 1. Comparison between the conflative and traditional tutoring approaches

	Conflative (EM-based) approach	**Traditional ITS approach**
Modelling approach	Empirical	Theory-based
Learning model	Constructed	Given
Adaptation	Transparent	Black box
Roles in the learning community	Flexible	Fixed
Direction of modelling	Bottom-up	Top-down
Modifications to the tools	On demand in the actual learning situation	Through the software development process

In the robotics classroom, the open-ended nature of robot building and programming typically leads to students taking completely different approaches to the activity. Accordingly, the teacher might not be satisfied with the existing sets of agents and rules for them, so that the system needs to be modified. In the traditional educational technology development process, the software developer does this. The developer can also make major modifications to the environment – for example, adding new data representations to the environment to create alternative views to record the students' progress. However, traditional software development methods are not flexible enough to support the teaching process within modern learning environments, where students explore solutions to problems independently.

3. ABOUT EMPIRICAL MODELLING

Empirical Modelling (EM) is a collection of principles and tools developed by Beynon, Russ and their students at the University of Warwick, UK. EM can be used to construct computer-based models that are based on the modeller's empirical observations about the phenomenon that is the subject of the modelling process. The modelling is done in the *tkeden* environment with several different notations.

The EM model is constructed by defining *observables* and *dependencies* with the notations mentioned above. An observable is a "computational" entity (such as a line, window, string or list of scalar values) that represents an element of the modelling subject. A dependency is a relationship between two or more observables. A key feature of the EM approach is that, after an initial definition, the EM environment automatically keeps the model updated according to the dependencies. This is similar to spreadsheet applications where values of the cells are updated automatically according to formulas that might contain references to other cells. In the next section we present through an example how the EM can be applied in the conflative learning environment.

4. A CONFLATIVE LEARNING ENVIRONMENT

To support the teacher's working process in a learning environment, such as an educational robotics class, where unpredictable learning activities often take place, we have proposed a concept of *conflative learning environment* (CLE) [4]. By exploiting the EM principles described earlier, the CLE gives full freedom for the teacher to modify the environment and support system to match the current situation.

The CLE framework consists of two parts. First, a number of *agents* work in the background of the learning process collecting observations about students' activity. Second, the teacher has a *model* constructed with the EM tools that reflects the current situation in the classroom, and the model-building is an ongoing process that accompanies the learning activities themselves. Each agent in the system has a dedicated task to which it has been appointed during the modelling process. For example, an agent might observe the use of a button in the graphical user interface of the robots' programming environment. This agent sends a message to the teacher's modelling environment over the network. The message can take the form of an EM definition so that the message redefines parts of the model. Alternatively, the message can be a natural language string that will be presented to the teacher as text. All types of messages contain a timestamp, and the messages are recorded in a database for later use.

The general idea is that agents do not process data by themselves, but collect data and deliver the data to the teacher's model and database for further observation. Even so, two different levels of "intelligence" can be distinguished within the agent population. The simplest form of agent works as a data collector.

For example, an agent can observe a button in the robot's programming environment and send a message to the teacher's classroom model when students press that particular button. A more sophisticated agent possesses limited computing capabilities that enable it to do simple reasoning. For example, an agent can observe the existence of keywords or certain structures in the students' program code.

The working process in the educational robotic class usually takes a cyclic form. It is thus crucial that the teacher's tools also support cyclic working methods where the teacher can redefine the tool as needed when unpredictable events occur in the classroom. According to Empirical Modelling principles, the model is built up gradually by making redefinitions. The current state of the model is at all times captured by the set of definitions that have been introduced to date. Redefinitions can originate from both human participants and the automated agents, and these definitions then affect the model according to the current dependencies. The CLE periodically includes new definitions produced by the agents, and in this way the agents can automatically update the model according to the current situation.

It is obvious that there are technical challenges in using the EM tools to construct a learning environment. However, the teacher does not have to have expertise comparable to that of a technical developer. The most important thing is that the teacher utilises his or her expertise in the learning domain, and that the teacher has a clear understanding of the observables that mediate the learning activity. To make the EM-based conflative learning environment more accessible for the teacher, we propose that the modelling of the learning process should be divided into two parts. The first part, *technical modelling*, consists of setting up the basic modules of the environment. This part of the modelling process can take place before and even between the robotics classes, when the model can be redefined to meet the new requirements. The second part, *pedagogical modelling*, is the process that takes place during the classes. In this part of the modelling process, the teacher defines contextually meaningful observables and visualisations for the data that the agents collect. It is possible that these observables are usable in context-specific settings, for instance, for a particular class, or dependent on the phase where the students are in their project (building, programming, or testing).

4.1 A prototype application

By following the principles of constructing the conflative learning environment described in the previous sections, we have built a prototype environment to support teachers' intervention in the robotics classroom. The environment has been built gradually by following the cyclic process of EM model-building. As a starting point for the model building process, we conducted two experiments in which we collected data and analysed students' activities with a simple EM model as described in [4]. Based on the results and technical lessons learned from these experiments, we have constructed a model that can be used as a starting point for building a contextualised observation environment for different kinds of robotics classroom settings (Figure 1).

It is crucial to note that the application is an example, and most likely does not fulfil all the requirements of a teacher working in an educational robotics class. This is due to the fact that each teacher may want to observe different issues from the classroom and the learning process. We have built the application in such a way as to give a good overall impression about the use of the conflative learning environment framework and the potential of the EM tools in this kind of model building process.

With the application, a teacher can observe the progress of the student groups through various modules with graphical user interfaces. The modules are updated automatically as the agents make new observations and deliver them to the EM modelling environment. Furthermore, the teacher can simulate the students' progress subsequently based on the data that the agents have automatically collected and stored in a database. This post-processing can be also done with rules different from those used in modelling in the real-time situation. In this way, the teacher can potentially learn new things about students' actions and progress.

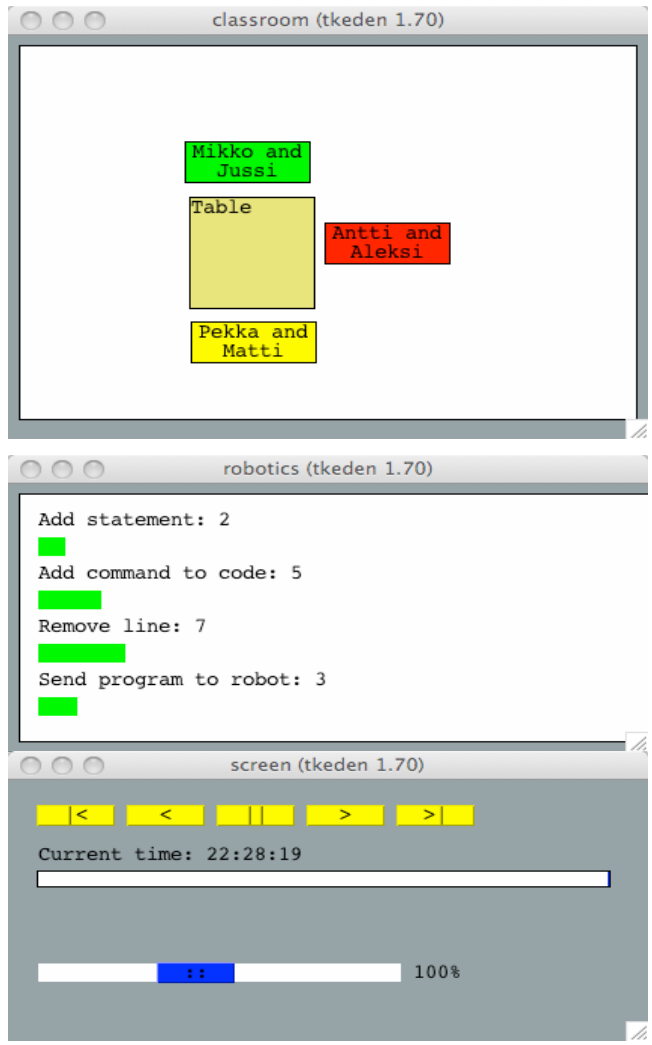

Figure 6. The prototype of the observation environment.

The current model consists of three modules. The first module (Figure 1, topmost window) shows a simplified map view for the classroom. The teacher can use this module to observe the overall progress of the student groups. In this prototype implementation, the student groups are shown as rectangles with the name of the group in it. These group markers can be moved around on the screen to reflect the disposition of groups in the classroom. In the screenshot, one table has been modelled in the view and the student groups have been placed at the corresponding places around the table. The colour or size of the group marker can be bound by dependency to observables of interest – for example to the length of the program code that the student group has constructed so far. The model building is automated so that, besides automatically reflecting the agents' observations in the model, the groups are also appended or removed automatically from the model when they start or close the programming environment in the classroom. In this way, the model can readily be maintained to be consistent with the current situation in the learning setting. The second module (window in the middle, Figure 1) visualises the overall progress of the student groups as measured by a cumulative sum of clicks for the four most important buttons in the programming environment. The third module (lowermost window in Figure 1) implements replaying functionalities for the observation environment. By using the controls in the graphical user interface, the teacher can for example return to a certain moment in the learning process. All other modules are bound to this control module so that they will be updated to show the situation in the learning process in that particular moment of time. This module can also be used to process the data that has been automatically collected by the agents after the activity has finished, as opposed to in real time, and new rules and dependencies may be added for this purpose to give alternative views to the learning process. Reconstructing the live states of students' interactions so that the teacher can in principle experiment within these states in a fresh way is definitely one advantage of using the Empirical Modelling tools, and we argue that reconstructing the learning process like this is more difficult with traditional ITS tools.

This prototype application of three modules can be extended toward a more complete presentation of the classroom setting. As mentioned earlier, all visual elements can be redefined, and completely new views can be built to support the teacher as required in the current classroom situation.

4.2 Extending the application

An important aspect of the Empirical Modelling approach is the process of constant refinement of the model and the re-use of existing models. The EM repository[1] provides a catalogue of pre-existing models which can be modified to suit the new contexts. The adaptation of the existing models obviously requires a certain amount of work, and a technically oriented person should do this as part of the technical modelling process.

The conflative learning environment framework and the applications built on it can be also applied in other contexts. The data collection methods and learning process reconstruction tools are especially well-suited for deployment in other application areas. While building our robotics application, we applied the replaying module to an HIV/AIDS educational game. The new module allowed the teacher to replay students' actions in the game and analyse their thinking during the learning process. The adaptation of the existing module to a new context required very few changes to the original definitions, and the experience confirmed our view that Empirical Modelling can be used as an effective approach for constructing conflative learning environments.

5. DISCUSSION

Recently, low-cost and highly accessible educational robot kits have gained popularity in hands-on learning environments, especially in technical fields, including Computer Science. However, the effective use of educational robotics in the classroom requires new kinds of classroom settings and teachers have to change their teaching methods according to the needs of the new environment. The open-ended nature of robot building can lead to students taking completely different approaches to an activity, and the teacher's needs for information about the learning process are difficult, if not impossible, to predict.

In this paper, we have presented a learning environment that allows the teacher to get and process information about the learning process through the empirical observations arising from the process itself. The application utilise the Empirical Modelling environment and model-building process to allow the teacher to modify the environment to meet the requirements of a particular learning process. Unlike most adaptive learning systems, the model and the modelling process are transparent and open to the teacher. The prototype application for monitoring robotics classes in its current form has been built through a cyclic process which took as its starting point empirical data collected from real classroom settings [4]. As the Empirical Modelling process characteristically involves a gradual open-ended development of the environment, we shall also develop the model further to provide the teacher better support in the classroom. In addition, we shall bring the modelling environment to students' screens, so that even the students are able to assess their own learning based on the derived models. This is a step towards a fully open and equal tutoring system where all participants in the learning community can participate in the modelling of the learning process by bringing to the model their own view of the activities.

6. REFERENCES

[1] Beynon, W. M. and Roe, C.P. 2004. Computer Support for Constructionism in Context. In Proceeding of the Fourth IEEE International Conference on Advanced Learning Technologies, 216-220.

[2] George, S., and Despres, C. 1999. A multi-agent system for distance support in educational robotic. In the proceedings of the International Conference on Telecommunication for Education and Training, 344–353.

[3] Jipping, M. J., Calka, C., O'Neill, B., and Padilla, C. R. 2007. Teaching Students Java Bytecode Using Lego Mindstorms Robots. In Proceedings of the 38th SIGCSE Technical Symposium on Computer Science Education (Covington, Kentucky, USA, March 07 - 11, 2007). SIGCSE '07. ACM, New York, NY, 170-174.

[4] Jormanainen, I., Harfield, A., and Sutinen, E. 2009. Supporting Teacher Intervention in Unpredictable Learning Environments. In Proceedings of the 9th IEEE International Conference on Advanced Learning Technologies (ICALT 2009), 584 – 588

[5] Klassner, F. and Continanza, C. 2007. Mindstorms Without Robotics: an Alternative to Simulations in Systems Courses. In Proceedings of the 38th SIGCSE Technical Symposium on Computer Science Education (Covington, Kentucky, USA, March 07 - 11, 2007). SIGCSE '07. ACM, New York, NY, 175-179.

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

[1] http://www.dcs.warwick.ac.uk/modelling