Role Blending in a Learning Environment Supports Facilitation in a Robotics Class

Ilkka Jormanainen * and Erkki Sutinen
School of Computing, University of Eastern Finland, P.O. Box 111, FI-80101 Joensuu, Finland // ilkka.jormanainen@uef.fi // erkki.sutinen@uef.fi

*Corresponding author

(Submitted June 14, 2012; Revised December 20, 2012; Accepted March 7, 2013)

ABSTRACT

The open monitoring environment (OME) uses a novel data-mining approach to enhance teachers’ pedagogical interventions in a robotics class. According to earlier studies, decision trees resulting from an open and semi-automatic data-mining process are practically useful when classifying subsequently data arising from a robotics class. The current study shows that data-mining features of the OME can be used to predict and, hence, support learning processes also with real-time data. Results show that the data-mining features of the OME are affected by the nature and amount of data when working with a small number of students. Furthermore, the results show that the robotics-class instructors are able to modify the learning environment to match to the current context in a way that goes beyond normal teacher activities in a classroom. This role blending between a teacher and a software developer in a learning environment provides a novel way to build a personalized and contextualized support environment for a robotics class.

Keywords
Robotics, Data mining, Learning environment, Teacher support

Introduction

Effective use of educational robotics requires a learning environment, which supplies open and flexible support mechanisms for the members of a learning community. Teachers working in a robotics class often face a problem in following how student groups proceed in their projects, which are usually based on problem-based or inventive learning and emphasize independent working of groups or individual students. Typically, the students work with their tasks in a cyclic process that involves planning, building, programming, and testing their robots. Groups proceed differently, perhaps being in different phases in the cycle. During ten years of running robotics activities in various technology club settings, we have seen that these characteristics of a robotics environment present difficulties for teachers planning an appropriate intervention strategy for the particular context.

Traditionally, intelligent tutoring systems (ITS) have been used to predict students’ progress in learning environments. Student modelling is usually based on theoretical assumptions about learning and a predefined sequence of actions in the learning process. However, the unpredictable nature of robotics classes establishes a need for novel approaches that allow teachers to follow up on their students’ diverse learning paths and strategies. Categorizing diverse learning strategies allows the teachers to model their students’ progress. A reasonable model is a basis to understand and, hence, to intervene on possible hindrances on students’ learning paths.

We have introduced a novel open monitoring environment (OME) to help the teacher to monitor activities and explore learning processes in a robotics class. The OME monitors students’ actions in a robotics class by automatically collecting data from the learning process, i.e., students’ interactions with educational robotics environment. The OME renders the collected data to visualizations that represent the current classroom activities. The classification rules for the collected data can be created manually based on data available in the OME database, or semi-automatically, for example, by using data mining. The OME uses an open data-mining approach (in particular, decision-tree classifiers) for modelling students’ progress (Jormanainen & Sutinen, 2012). The OME aims not only to help teachers to facilitate the students’ progress, but can also help them to account for what has been happening in the learning process. Facilitation in the context of this research means that the OME produces interpretable visualizations and explanations about the processes in the classroom that are not necessarily visible for the teacher. In this way, the OME supports the teacher in facilitating learning at the right time. This is aligned with a modern teacher’s new professional expectation as a mentor rather than as a traditional teacher.
A key element of the facilitation in the OME is to predict the possible problems that the students are facing in the learning process so that the teacher can intervene accordingly. The OME does not, however, provide information about how to intervene, but in the ideal case, it can detect the series of events that have led to the current situation. The teacher using the OME gets the support for facilitation by exploring these series of events and building his or her intervention strategies, based on the information provided by the OME.

The study presented in this paper is a part of a development research project, which aims to build tools and approaches to support teachers working in unpredictable learning environments, especially robotics classes. During the research project, we have shown that the OME helps the teachers to intervene when the students have problems in their robotics exercises (Jormanainen et al., 2012). Furthermore, we have shown (Jormanainen & Sutinen, 2012) that the open data-mining process and a proof-of-concept implementation for the data-mining module in the OME produces useful and interpretable information about the students’ progress with relatively small datasets.

In this study, the teachers used the OME in a robotics class to analyze learning process with real-time data that was generated while the students were working with the robotics environment. In fact, the learning process was understood as consecutive snapshots of equal time intervals. Each snapshot contains recorded student activities and is an opportunity for the student to progress or get stuck, thus requiring a teacher’s intervention. In our previous experiments, we analyzed the student progress data retrospectively. The main research question of this article is as follows:

How can the OME support facilitation in a robotics class by providing the instructors learning process classifiers that are based on real-time data?

The results from the study are two-fold. First, we identified a situation where the OME failed to produce an expressive classifier, in contradiction to our previous findings in Jormanainen and Sutinen (2012) with the retrospective analysis. In this way, the OME did not provide the needed support for the instructors’ intervention strategies. Secondly, the instructors were able to adapt the OME to meet the context-dependent requirements, and adjust the OME eventually to fulfil the monitoring needs, from the ultimate viewpoint of understanding the students’ learning difficulties. The latter result supports our findings presented previously in Jormanainen and Sutinen (2012) and Jormanainen et al. (2012), where we concluded that the OME supports a novel role-blending approach between a user and a developer.

This article is organized as follows. In the next section, we describe related work and how our approach differs from existing educational data-mining systems. Then we describe the key concepts and tools of our approach: empirical modelling (EM) and open monitoring environment (OME). After the introduction of the research design, we present the main findings from the study. Finally, we conclude the article and suggest directions for future work.

Background

The open monitoring environment adopts many features that can be found in existing intelligent tutoring systems, including a distributed agent architecture for collecting data and the use of data-mining techniques to predict students’ progress. The main difference between the OME and traditional ITSs is that, whereas many traditional ITSs use a theory-based approach for building the learning model, the OME starts from the empirical observations arising from the current learning situation. There are ITSs that apply an empirical approach for building the learner model, for example Wayang Outpost, a multimedia ITS for geometry by Cooper et al. (2009). However, the learning models in these systems are still at least partially predicted based on theoretical assumptions (for example, a given set of features for classifying the user’s emotional self-concept in Cooper et al. [2009]).

Data-mining tools have a recognized status as a part of ITSs and other learning environments. Usually, data mining is used to extract knowledge from e-learning systems through the analysis of the data that the users have generated (Castro et al., 2007). Most of the work in data mining in educational systems contributes to students’ assessment (e.g., Elenbogen & Seliya, 2008), learning material and course evaluation, and course adaptation based on students’ learning behaviour (Kristofic & Bielikova, 2005). Clustering and classification are the data-mining techniques most
frequently used in learning environments (Castro et al., 2007) because of the nature of the problems that typically appear in this context.

However, two aspects limit the usability of the current data-mining approaches. First, data mining is used for inductive or deductive reasoning, i.e., inductively deriving the clusters from the available data or deductively applying the derived rules to classify data. Secondly, in most cases data mining is a black box for the learning community and its results are only implicitly visible to the users. For example, an adaptive learning environment uses learning process data to infer rules to classify learners based on their speed of progress or learning style. These rules are then used with learners or teachers, who are aware only of their existence, not of their contents. Complementary to the current mainstream uses of data mining for learning environments, we apply data mining in the OME for abductive reasoning (Ross, 2010) with transparent explanatory hypotheses. Teachers can elaborate the data-mining-generated rules to effectively facilitate the students. The rules help teachers to look backwards to the reasons behind the outcomes of the learning process, not only to efficiently support the process going forward.

The aim of an abductive problem-solving process is to find the best possible explanation for the current situation, based on data currently available. In the OME, the teacher starts the modelling process from a set of observations and generates the best possible account of the current learning process based on the observations. An abductive approach for building learning environments has been previously used by other researchers, for example by Qiu and Riesbeck (2004). However, the incremental development in Qiu and Riesbeck (2004) focuses more on contributing material during the learning process rather than on building the environment to monitor learning as the OME does.

In contrast to traditional educational data-mining applications, the mining process in the OME is transparent and results are explicitly visible to the teacher. Furthermore, the teacher is involved in the whole process, including training the data-mining algorithm, which is usually a domain expert’s or software developer’s task. Training set creation is an essential part of the process in the OME because a well-defined training set allows the data-mining algorithm to produce a well-working classifier. In the traditional data-mining application-development process, a domain expert or a software developer creates classifiers for an application with predefined data sets during the software development and deployment process. In many cases, data for training sets is processed manually, which is a very time-consuming task.

However, a teacher working in a robotics class has the best domain knowledge in his or her classroom context, and we believe that the teacher is the best expert to model the learning process in the classroom. The OME aims to provide suitable tools so that the teacher can handle the data flow originating from the robotics class and use data efficiently to produce classifiers that are aligned as well as possible with the current learning setting and pedagogical challenges.

To implement learning environments such as the OME, there is a need for tools that support abductive reasoning. Empirical modelling (EM) provides appropriate tools and methods for this. Before describing the OME environment in detail, we shall first introduce the EM tools through illustrative examples.

**Empirical modelling**

We implemented the OME by using the empirical modelling (EM) toolset (Empirical Modelling, 2012), because it is naturally well aligned with the goals of the development process. 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 modeller works with the model, the *tkeden* environment, by defining observables and dependencies using several different notations (Empirical Modelling, 2012).

An observable in the EM environment is an entity (such as line, string, number, window, or list of scalar values) whose current status can be inspected by the modeller. A dependency is a specific kind of relationship between two or more observables. The observables and dependencies in the EM environment are intended to serve as direct counterparts of observables and dependencies of the phenomena being modelled. The use of empirical modelling in the monitoring environment was illustrated in our previous article (Jormaainen et al, 2009). All the aspects of the current state of the model are expressed through definitions. The teacher can, in principle, manage all the details of
the model. In this way, the EM approach guides and encourages the teacher to adopt the software developer’s tasks in order to provide the right support for the students just in time.

**Open monitoring environment**

Open monitoring environment (OME) is an interactive tool that helps a teacher to facilitate students’ learning in a robotics class by providing support for the teacher about when and how to intervene in students’ work. An educational robotic setting usually produces so much data that the OME needs to provide the teacher with semi-automated tools for interacting with data. Data mining helps the teachers to synthesize the massive amount of low-level data so that it would be possible to identify more complex patterns of actions that potentially are of interest when monitoring the learning activity. It is important to note that the OME does not judge students’ learning but provides the teacher with an additional way to predict the possible moments of problems and find explanations behind them.

The OME allows the teacher to model and explore the current learning process based on data that is automatically collected from students’ interactions in the robotics environment (Figure 1). The atomic data rising from the learning process is saved to the OME database and summarised further into time slots. The teacher working with the OME synthesizes training sets for the data-mining algorithm from this data by exploring the real-time information available through the system and combines the findings with his or her own observations from the classroom. The created training sets are used to build decision-tree classifiers that are used to visualize students’ predicted workflows and situations where the particular patterns of actions may lead. Technical details of the data-mining procedure in the OME are discussed later in this article.

![Figure 1. Overall architecture of the OME](image)

We have shown previously (Jormanainen & Sutinen, 2012) that data mining, and decision trees in particular, are efficient for classifying and modelling students’ progress based on data originating from a robotics class. Based on these results, we have built a data-mining module for the OME to enable an interactive process in which the teacher is involved in building decision trees for classifying students’ progress (Jormanainen & Sutinen, 2012). In the OME, the rules are represented as decision-tree classifiers and they are explicitly opened for the teacher’s revision. The teacher can disagree with and change the automatically created rules at any time. Furthermore, the OME, in principle, allows the teacher to control and redefine all the details of the working environment. Any changes will be applied immediately in the environment, without closing it.
Data collection in the OME is based on a distributed agent architecture where software agents observe students’ actions with the robotics environment and deliver data entries with timestamps to the OME database for further processing (Jormanainen et al., 2007). Students’ learning environment consists of Lego Mindstorms RCX educational robotics sets and the IPPE graphical programming environment (Figure 2) for the Lego robots (Jormanainen et al., 2002).

![Figure 2. The IPPE programming environment. Numbers 1–4 refer to Table 2](image)

In the experimental environment, data is collected from the four major functions within the IPPE programming environment. These functions define the key steps in the robotics programming process in this context: 1. Creating individual commands to the sandbox, 2. Constructing a program from these commands, 3. Removing program lines if necessary, and 4. Uploading the program to the robot. The number of event sources was intentionally small, as our aim has been to provide a simple version of the OME for the teachers in this phase of the research project, which still involves a remarkable number of technical aspects.

The OME produces different kinds of visualizations for the current learning process. The visual representations are needed because the amount of data that a robotics environment produces can be too overwhelming for the teacher to handle as a flow of raw data (Jormanainen et al., 2009), despite the small number of attributes. Hence, the OME provides tools for enriching data to pedagogically meaningful collections. The teacher creates and manipulates rules that define how the raw data is treated and visualized. The progress (or lack thereof) can be expressed, for example, with colours or sizes of graphical elements representing the student groups.

![Figure 3. Classroom view module of the OME](image)
As the empirical modelling environment allows the user to redefine all details of a model, the teacher can, in principle, create any kinds of rules and visualizations based on them. However, complete control over an EM model requires a remarkable amount of technical skills not available in a regular classroom. In order to integrate information originating from the classroom into the teaching process, the technical environment for collecting and analyzing data needs to be simple enough, so that the lack of technical skills does not prevent the full use of the environment. At the same time, the environment needs to preserve its flexibility and provide room for the teacher to use pedagogical expertise efficiently in student progress modelling, without having predefined rules and visualizations to restrict the pedagogical process. In the experimental OME version, the students’ progress is visualized with a 2D map of the classroom (Figure 3), with the group markers showing the current situation with one of the four colours indicating the group’s progress as resolved by the current rules (Table 1).

<table>
<thead>
<tr>
<th>Class</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>White</td>
<td>Students are not progressing, but they do not seem to have any particular problems (neutral situation).</td>
</tr>
<tr>
<td>Green</td>
<td>Students are progressing without any noticeable problems.</td>
</tr>
<tr>
<td>Yellow</td>
<td>Students might be facing problems (intervention may be required soon).</td>
</tr>
<tr>
<td>Red</td>
<td>Students have experienced problems that require intervention.</td>
</tr>
</tbody>
</table>

Beside the classroom view, the OME supplies the teacher with a graphical view for the number of events in a specific time window (Figure 4). This view shows an overall progress in the class during the time window and allows the teacher to classify the events into one of the four categories according to the anticipated progress. This is a key element of data-mining process and relies on teacher’s pedagogical expertise and face-to-face experiences with the students in the classroom.

During the classification process, the teacher creates a training set for the data-mining module of the OME. The module uses the Weka 3 data-mining environment (Hall et al., 2009) to create a J48 decision-tree classifier for predicting the students’ progress. J48 is an open-source implementation of the popular C4.5 decision-tree algorithm (Quinlan, 1993). Table 2 presents the data sources in the programming environment and corresponding attributes in training set data. Classification of a time slot produces one line for the training set.

<table>
<thead>
<tr>
<th>Event source</th>
<th>Attribute</th>
</tr>
</thead>
<tbody>
<tr>
<td>Button 1: Add statement</td>
<td>Act_addstatement</td>
</tr>
<tr>
<td>Button 2: Add command to code</td>
<td>Act_addcommandtocode</td>
</tr>
<tr>
<td>Button 3: Remove line</td>
<td>Act_removeline</td>
</tr>
<tr>
<td>Button 4: Upload program to robot</td>
<td>Act_senдоровrobot</td>
</tr>
<tr>
<td>Compiler</td>
<td>Err_comp</td>
</tr>
<tr>
<td>(no specific source)</td>
<td>Sum</td>
</tr>
</tbody>
</table>

Figure 5 shows the definitions for the training data as well as an example of the data in Weka’s arff-format. Attributes in Figure 5 are described in Table 2. In the example entry, the agents have identified one “Add command to code” event and one “Upload program to robot” event, i.e., two events within time window of one minute.
Therefore, the progress within this time window was evaluated green (students were progressing without noticeable problems).

```plaintext
@attribute act_addstatement numeric
@attribute act_addcommandtocode numeric
@attribute act_removeline numeric
@attribute act_sendtorobot numeric
@attribute err_comp numeric
@attribute sum numeric
@attribute progress_class {white, green, yellow, red}
data
0,1,0,1,0,2,green
```

*Figure 5. Training data definition and an example*

Once a sufficient set of data has been classified, the teacher can create a new decision-tree classifier by pressing the “Generate tree” button in the progress classifier module (Figure 4). The OME launches a Weka data-mining application as a background process and passes the training set based on teacher’s choices to the J48 decision-tree algorithm. Without interrupting or closing the environment, Weka generates the formal description of the new classifier, based on the supplied training set. The OME transforms the description into a tree visualization (Figure 6) that is identical to the executable rules as empirical modelling definitions (Figure 7).

*Figure 6. Tree visualization module of the OME*

If the teacher is not confident with editing the classifier rules, he or she can always create a new classifier by continuing the classification process as students progress in their task and more data is available for a new training set. The teacher can also always browse the previous time windows and re-classify them, and try to construct a more suitable classifier in this way. This is a good approach especially at the beginning of a teaching session when training sets are too small for the J48 algorithm to build a classifier that is expressive enough. All visualizations are automatically updated through empirical modelling dependencies whenever new data is inserted to the OME database, when the teacher creates a new decision-tree classifier or when the user modifies the other EM definitions.

The IPPE, and robotics environment in general, possess a vast amount of relevant information that can be used in the future development. This information includes, for example, quantities and qualities of the students’ program code for the robots, robots’ physical construction, and even roles of the students in the groups and interaction between them. It is clear, however, that not all of these aspects can be monitored automatically. For example, observing robots’ physical construction or students’ handwritten notes is difficult as a background process in a natural learning situation where additional monitoring devices (cameras, microphones, etc.) easily interfere with the learning process.
Research design

Research methodology and research question

The experiment presented in this paper is a part of a development research project where the emphasis is on an iterative development process of the environment. We have reported the design process and initial experiments with the OME in Jormanainen et al. (2012) and in Jormanainen and Sutinen (2012). In Jormanainen and Sutinen (2012), we used the data-mining module of the OME to make a retrospective analysis of the robotics class activities. We concluded that data mining, and decision trees in particular, are effective for classifying students’ progress in the educational robotics setting, and that the open data-mining process produces useful and interpretable information about the students’ progress with relatively small datasets.

In contrast to Jormanainen and Sutinen (2012), we report in this article a study that was conducted in order to see how robotics class instructors use data-mining features to build student progress classifiers that are based on currently available, real-time empirical data. The study described in this paper focuses on the following research question:

How can the OME support facilitation in a robotics class by providing the instructors learning process classifiers that are based on real-time data?

Research setting

The experiment described in this paper took place in Kids’ Club of University of Eastern Finland. Kids’ Club is a combined after-school robotics and technology club and an educational technology living laboratory as a platform for designing novel educational technologies together with their intended users, particularly students and teachers. Altogether 13 club participants (between 10 and 13 years) were involved in the study. They were divided into two groups, and these groups were further divided into tree project groups of two or three students. The study was conducted individually for both groups of six or seven students. Each group spent about 30 minutes with their project.

The student groups were provided with a pre-constructed Lego Mindstorms RCX robot. They were asked to program the robot (1) to drive forward and backward for five seconds, and (2) to drive a square. The students were familiar with the basic concepts of Lego robots, as they had used Lego Mindstorms NXT educational robotics sets during the earlier club activities. However, they had not used the older Lego Mindstorms RCX or the IPPE programming environment.

Two club instructors, a 25-year-old man and a 30-year-old woman, participated to the study. Neither instructor was familiar with the OME, the IPPE programming environment, or the empirical modelling toolset before the study. The instructors had been involved in Kids’ Club activities for a school semester before the study, but they did not have a
Data collection

During the experiment, the instructors were using the OME to observe students’ activities in the classroom and to build decision-tree classifiers on the fly in order to see how they predict students’ progress and possible problems. The students’ interactions with the IPPE programming environment were recorded from the main functions of the programming environment (Figure 2). Altogether, 145 events were delivered over a local area network to the OME database, which was implemented as an EDDI (EDEN Database Definition Interpreter) database in the empirical modelling environment (Figure 9). Each observation contained timestamp, as well as information from which group and specific component of the IPPE programming environment the event originates from and the type of current event (user or error event).

EDDI database table structure:

observation (timestamp int, hostname int, event_type int, sender int, message int);

Examples in EDDI notation:

observation << [49462484,1,2,2,7];
obervation << [49492937,2,2,1,5];
obervation << [49523578,2,2,2,7];
obervation << [49519937,1,2,4,11];

Figure 8. Structure and example of automatically collected event data

The summary of the events was collected and presented to the instructors in the OME environment in time windows of one minute (Figure 4). The presentation reflected a real-time progress in the classroom, and new interaction data was added to the summary whenever it became available. Based on this data, the teachers created during the session 13 different versions of decision-tree classifiers by following the workflow described above. Each decision-tree version was saved to a computer’s hard disk for further analysis. Examples of different iterations are presented in the next chapter in Figures 10 and 11.

Beside the data that the students and teachers contributed during the process, the researcher observed how the instructors used the OME, with supplementary questions during the teaching session. The questions were mainly related to how the instructors interpreted the currently available decision-tree models. Furthermore, the instructors were given minor technical guidance with the empirical modelling environment during the teaching session. After the teaching sessions, the instructors were interviewed about their opinions about the OME and how it potentially could benefit their work in robotics classes in the future.

Analysis methods and tools

The collected research material (decision-tree classifiers and supplementary material, such as field notes) was analyzed qualitatively. We analyzed the decision-tree models (n = 13) after the teaching sessions by interpreting them in the light of domain expertise that we had gathered during running robotics activities in various technology club settings for ten years. The focus of the analysis was to see how the decision-tree models have evolved during the teaching session when the amount of data in the OME database has been growing gradually while the students have been working in the robotics environment. Furthermore, instructors’ working process was analyzed in order to see how they used the OME and data-mining visualizations to support the facilitation in the learning process. The analysis was supported with extensive field notes collected during the teaching sessions.
Results

The main result of the study shows that the OME supports the teacher’s facilitation process in a robotics class by allowing the teacher to create decision-tree classifiers that can be used together with classroom visualizations (Figure 3) to interpret and explore the current progress in the classroom and to predict the possible points of problems.

The conclusion was reached by analyzing how the decision trees evolved during the teaching session while the amount of student interaction data increased gradually. The last-created decision-tree classifier (Figure 10) and the visualizations attached to it described the progress so that the instructors were able to use them to predict the problems and intervene in the students’ work before students got stuck.

Furthermore, the deeper analysis was supplemented by observations of how the instructors used the OME while helping the students and how the instructors interpreted information provided by the OME and decision-tree visualizations in particular. The main result as stated above can be further divided into two parts. At first, the OME (as prepared for this experiment) did not produce very expressive or meaningful classifiers, despite several iterations of refining training data and constructing the decision tree (Figure 9). The first version of the tree (left screenshot in Figure 9) classifies progress meaningfully, but it takes into account only a small part of the process. For example, it does not recognize actions when students are uploading the program code to robots nor possible problems occurring in this phase. This may be because the classifier was created in an early phase of the robotics class and such data has not yet been available. On the other hand, the final version of the decision tree (right screenshot in Figure 9) classifies students’ good progress based on frequencies of the “Remove line” action. In the context of the experiment (and in robotics classes in general), this hardly makes sense.

Because the instructors were not satisfied with the resulting classifiers presented above, they started to think after a while how they could reach better results with the data available. When reclassification of the existing data did not produce the desired result, they started to modify the rules produced by the data-mining algorithm in the rule window of the OME (Figure 7). After carefully examining the decision tree available at that time, the instructors disagreed with some choices made by the J48 algorithm. For example, they changed variable sum to variable add_statement in one of the conditions (tree nodes) in order to achieve better results. When even these changes did not improve the result enough, the instructors started to explore more deeply how the data-mining process works in the OME.

The researcher had explained earlier the basic principles of the classification and training set creation in the OME. By using this knowledge, the instructors were able to detect a possible bottleneck in the process. When the OME creates a training set, it divides sums of individual events with the number of the groups within a time window. For example, if three student groups had produced 12 events in total within a time window, the training set was supplied with an average of four events. The instructors concluded that this might prevent a proper classification due to the size or other characteristics of data sets available in the current robotics class, that is, the sums of data items were so small that taking averages from these sums led to training sets with inappropriate values for the J48 algorithm.

Guided by the researcher, the instructors explored the functionality and definitions of the OME through the empirical modelling environment. As part of the modelling process, the EM environment allows a user to query and modify the state of the current observables, functions, and procedures with an interactive tkeden interpreter.
After exploring the OME definitions for a while, the instructors identified a procedure that is used to create the training set based on the user’s choices. Furthermore, the instructors identified a block in the procedure (three lines of the EM definitions) that calculates the averages. The instructors opened an appropriate code file in a text editor and removed the unwanted block. After that, they applied the modified procedure to the OME without interrupting the environment. It is notable that the instructors were not previously familiar with the EM environment or the syntax of the EM definitions. Hence, the researcher gave technical advice to them to complete the task. As a result, the data-mining module in the OME started to produce significantly better results (for clarity, the tree in Figure 10 has been redrawn based on the original OME visualization).

Figure 10. The decision tree after modifications of the OME

At the first sight, the classifier in Figure 10 is more expressive and it describes the learning process better than the final result with the original OME environment (Figure 9). For example, progress is classified as the green category (students are progressing without any noticeable problems) in N6 when there are no errors and the students have removed only few lines from the code they are working with. This indicates that they have created commands, constructed robot code from these commands, and finally uploaded the program to the robot successfully (error events occur when there are problems with compiling or uploading the code).

A thorough interpretation of the decision tree reveals that the classifier exposes patterns of actions that would be otherwise hard to predict. For example, the classifier indicates possible problems arising (yellow) in the node N8 when there are no errors (N4) and the students have removed more than two lines from the code (N5), but they have applied “Add command to code” functionality just once or not at all (N7). This is not a part of a typical workflow with the IPPE programming environment, but the situation is, indeed, possible to reach as the IPPE allows editing code straight to the code editor. In the current research setting, the students had, however, reached the situation by removing the initial program code lines for the body of the robot’s program that appeared in the editor during the start-up. This is clearly an undesired action, because it makes compiling the program code impossible and teacher’s intervention needed.

After the teaching session, the instructors analyzed the final classifier based on their experiences from the classroom. They were able to trace the tree from the root node and justify the choices made by the J48 algorithm. Furthermore, they agreed that the tree gives a deeper view for the individual group’s progress in this particular robotics environment than the trees that the original OME version produced (Figure 9). Finally, the instructors concluded that the tree model could serve as a suitable starting point when preparing the OME for future robotics classes.
Discussion

The findings from this study contradict, to some extent, our previous results (Jormanainen & Sutinen, 2012), when we tested the data-mining capabilities of the OME with small data sets. It is notable that in the current experiment, an average size of a data set in a time window was 20% smaller than that presented in Jormanainen and Sutinen (2012). This may have had a negative effect for the functionality the J48 decision-tree algorithm in the first place. On the other hand, this strengthens our assumption that data-mining process in the OME is very sensitive to the nature of data collected from the learning process, as well as to user’s personal preferences and pedagogical expertise. This makes the classifiers highly context-dependent, and there is a need to provide the teacher with the tools that allow him or her to deeply control the details of the student modelling process.

The interpretability and pedagogically meaningful constructions of decision-tree classifiers are even more important aspects in the OME than the traditional measurements, such as the accuracy and precision ratio that are usually used to evaluate the data-mining applications. It is evident that small data sets in the experiment interfered at first with the data-mining process so that the resulting classifiers were not as expressive or interpretable as those presented in our previous study (Jormanainen & Sutinen, 2012). However, more interesting and unexpected results rose from the instructors’ working process with the OME. The most interesting finding was that the instructors were independently able to identify the problem in the functionality of the OME and explore the construction of it with the supplied tools. More importantly, they were able to apply the changes to the core OME definitions so that the environment started to produce better classification results in the current learning setting. This process goes beyond a teacher’s traditional role and it is well aligned with the principles of the role-blending approach as described in Jormanainen et al. (2012).

If the modifications that the instructors made during the experiment are considered strictly in the light of correctness of the program, it is notable that the modifications made the OME to work wrong. As a result, the training set based on cumulative sums of events is used to create a classifier to predict the progress of individual groups, and we can say that in a general software development project this is not appropriate. However, in the context of this particular robotics class, the “wrong” behaviour of the OME was desired at least when only small data sets that were available during the experiment. This shows that a learning environment can be constructed by using unconventional solutions if it helps to support the unpredictable nature of the learning setting, as long as the changes are manageable by the end-user (in this case, the teacher).

Conclusion

We have presented a novel monitoring environment that facilitates teachers’ intervention in educational robotics classes. The environment uses data mining to classify events rising from the classroom and to predict students’ progress. Unlike traditional educational data-mining applications, the open monitoring environment allows the teacher to create training sets for the decision-tree algorithm based on real-time data. The created classifier is visualized for the teacher and forthcoming events are classified based on this decision tree. If not satisfied, the teacher can iterate the process with more data or manipulate the current classification rules to match them as well as possible to the current learning scenario.

We demonstrated previously that decision trees are efficient when classifying data originating from the robotics classroom (Jormanainen & Sutinen, 2012). In this paper, we have presented a study, which shows that a) the selected J48 decision-tree algorithm is sensitive for the nature and amount of data when working in small robotics settings, and b) instructors in a robotics class are able to modify on-the-fly even the core definitions of the monitoring environment to contextualize the environment to suit the current conditions (i.e., a small number of groups and not enough data). This process with role blending between the teacher and the software developer differs radically from the traditional approaches with static roles and predefined working environments. Effects of group size and amount of data must be studied further in a full-scale robotics class (20–30 students working in parallel). Another aspect for the further studies is to allow the teacher to use arbitrary time windows in the classification instead of using a fixed time window of one minute. Letting the teacher decide suitable lengths for the time windows could enable deeper interpretation of the learning process when the specific events in the classroom can be connected to each other over the longer periods of time.
Acknowledgements

We would like to thank Dr. Meurig Beynon (University of Warwick, UK) for his comments for the manuscript.

References


