DISSERTATIONS IN

FORESTRY AND NATURAL SCIENCES

Ilkka Jormanainen

Supporting Teachers in Unpredictable Robotics Learning Environments

Publications of the University of Eastern Finland
Dissertations in Forestry and Natural Sciences No 121

UNIVERSITY OF EASTERN FINLAND

ILKKA JORMANAINEN

Supporting Teachers in Unpredictable Robotics Learning Environments

Publications of the University of Eastern Finland
Dissertations in Forestry and Natural Sciences
No 121

Academic Dissertation
To be presented, by permission of the Faculty of Science and Forestry, for public examination in the Auditorium AU100 in Aurora Building at the University of Eastern Finland, Joensuu, on September 20, 2013, at 12 o'clock.

School of Computing

Kopijyvä
Joensuu, 2013
Editors: Prof. Pertti Pasanen, Prof. Pekka Kilpeläinen, Prof. Kai Peiponen, Prof. Matti Vornanen

Distribution:
University of Eastern Finland Library / Sales of publications
P.O. Box 107, FI-80101 Joensuu, Finland
tel. +358-50-3058396
http://www.uef.fi/kirjasto

ISBN: 978-952-61-1230-5 (printed)
ISSNL: 1798-5668
ISSN: 1798-5668
ISBN: 978-952-61-1231-2 (pdf)
ISSNL: 1798-5668
ISSN: 1798-5668

Author's address:	University of Eastern Finland School of Computing P.O.Box 111 80101 JOENSUU FINLAND email: ilkka.jormanainen@uef.fi
Supervisors:	Professor Erkki Sutinen, Ph.D. University of Eastern Finland School of Computing P.O.Box 111 80101 JOENSUU FINLAND email: erkki.sutinen@uef.fi
	Reader Emeritus Meurig Beynon, Ph.D. University of Warwick Department of Computer Science Coventry UK email: wmb@dcs.warwick.ac.uk
	Professor Kinshuk, Ph.D. Athabasca University School of Computing and Information Systems Edmonton CANADA email: kinshuk@ieee.org
Reviewers:	Associate Professor Ryan Shaun Joazeiro de Baker, Ph.D. Teachers College Columbia University Department of Human Development Box 118 New York, NY 10027 US email: baker2@exchange.tc.columbia.edu
	Associate Professor Piret Luik, Ph.D. University of Tartu Institute of Education Salme 1a Tartu 50103 ESTONIA email: piret.luik@ut.ee
Opponent:	Professor Henrik Hautop Lund, Ph.D. Technical University of Denmark Department of Electrical Engineering Building 349 DK-2800 Kgs. Lyngby DENMARK email: hhl@elektro.dtu.dk

ABSTRACT

Understanding and supporting school student groups' progress in educational robotics classes is demanding because the groups usually take unpredictable paths in problem solving. To improve teachers' possibilities for intervention at the right time, we have introduced Open Monitoring Environment (OME) that allows the teacher to monitor, model and, thus, understand, the learning process based on the real data rising from the current learning setting. In parallel with developing the OME, we have defined a Conflative Learning Environment (CLE) approach that guides construction process of environments for unpredictable learning settings. The environments built by following the CLE approach allow teachers to blend design and development activities of a learning environment with their conventional work in the classroom. The aim of the role blending process is to make the learning environment to support facilitation better also in unpredictable learning situations.

The OME uses a novel approach where rules derived from the learning process are open for revision by the teacher. The rules are used to extract pedagogically and contextually meaningful patterns of actions from the data flow that various agents produce in the robotics environment. For example, the agents may observe the type and frequency of students' interactions with the learning environment and deliver this data for further processing to the teacher's monitoring environment. By using his or her pedagogical expertise the teacher modifies the rules to describe the current learning process. Deriving the pedagogically meaningful rules from an unpredictable learning setting where agents produce excessive amount of data is usually too difficult for the teacher. The OME features a data mining approach, which helps the teacher to create potentially meaningful rules. In line with the design principles of the OME and the CLE approach, the whole data mining process is accessible and transparent for the teacher. Similar student modelling and data mining processes in the learning environments work typically in a black box and only results are visible for the users.

The work presented is a result of an experimental development research with initial implementations of the OME. The OME was tested in authentic educational robotics settings in a Finnish after-school technology club and in a South African primary school. The results show that the teachers can use their pedagogical expertise effectively to analyse their students' progress, as informed by the OME. Furthermore, the results show that open data mining techniques, and decision trees in particular, can be used to support teachers' intervention in robotics classes.

Universal Decimal Classification: 004.42, 004.6, 37.091.26, 37.091.321, 37.091.33

Library of Congress Subject Headings: Educational technology; Teaching; Teachers; Learning; Open learning; Robots; Robotics; Educational evaluation; Design; Data mining; Decision trees

Yleinen suomalainen asiasanasto: opetusteknologia; opetus; opettajat; oppiminen; oppimisympäristö; suunnittelu; kehitys; oppimistulokset; robotit; robotiikka; tiedonlouhinta

Preface

Good ideas may take some time to get ready, but one has to push the limits sometimes to reach the goal. I warmly thank Professor Erkki Sutinen who enforced me to believe in that. I wouldn't be writing this without Erkki's patient supervision and the endless idea generator. Especially I'd like to thank for the encouraging discussions we have had in the luxurious heat of Mekrijärvi's smoke sauna and in the dark African night in the bank of Great Ruaha River.

Dr. Meurig Beynon guided me to the fascinating world of Empirical Modelling, and provided constructive and critical enough feedback for the article and thesis manuscripts. Professor Kinshuk had a crucial role at the beginning of the research by familiarising me with agent technologies. He also kindly hosted my visit at the ALTRC of Massey University in *Aotearoa*, "Land of the long white cloud". Besides getting a good start for the research, it was easy for us to fell in love with New Zealand.

I highly appreciate the detailed and constructive feedback and guidelines for the future research that I have received from the pre-examiners, Associate Professor Piret Luik and Associate Professor Ryan S. Baker. I'm very happy to have Professor Henrik Hautop Lund for an opponent for my dissertation.

No research can be done alone. Dr. Antony Harfield was a great help when I was struggling with concepts and implementation details of Empirical Modelling. We also co-authored articles for the thesis and spent memorable moments all over the world. I truly hope that we shall meet before long and continue the enjoyable collaboration! I also thank Mr. Yuejun Zhang for co-authoring the papers with me.

edTech$^{\Delta}$ research group of UEF has been a great base for the research. I would like to thank Jarkko Suhonen for managing the chaos and guiding with the practical matters. Jussi Nuutinen provided me great peer-support before escaping to the real work after his graduation (which, once upon time, was supposed to be on the same time with mine...). I have had many great conversations with Roman and Carolina. Especially I'd like to thank Marcus "Dewey" Duveskog and Andrés "Louie" Moreno for sharing *the experience* and great, sometimes even research-oriented, adventures in Africa.

Kids' Club technology club and its instructors and tutors have provided a very functional platform for carrying out the research. I'd like to thank Marjo Voutilainen for our joint efforts on developing technology and robotics education. Mikko Laamanen has taken a great initiative on developing the club activities further. He has been also working intensively with the web-enabled version of the OME. I'd like to thank Päivi Sutinen, Brenda Windram, Thato Foko, Seugnet Blignaut, staff members of Hands of Hope School in South Africa, and Tuija Jetsu for helping to organise the case study and to collect and analyse empirical research material.

This research has been carried out at the School of Computing of University of Eastern Finland (previously Department of Computer Science, University of Joensuu). I'm thankful for the funding and practical help that the staff members of the department have provided in various occasions. I appreciate support from the North Karelia Regional Fund of the Finnish Cultural Foundation. Joensuu Science Society has been a great place to work during the studies. It has been a privilege to work with Kirsi, Pauliina, and Heikki on developing SciFest and other science and technology education initiatives in a very concrete level.

At last, I would like to thank my parents Erkki and Arja and my brother Jouni for believing in what I have been doing and providing all kinds or support whenever needed.

My beloved ones, Sari and Aleksi! You have bravely carried your share of my studies. I'm sorry for the long nights and me being absent too often. After all, the best thing I can imagine is to see Aleksi learning how to ride a bicycle and experiencing it with you!

Lehmo August 27, 2013 *Ilkka Jormanainen*

LIST OF PUBLICATIONS

This thesis consists of the present review of the author's work in the field of educational robotics and the following selection of the author's publications:

I I. Jormanainen, Y. Zhang, Kinshuk and E. Sutinen. "Pedagogical agents for teacher intervention in educational robotics classes: Implementation issues". In *The First IEEE International Workshop on Digital Game and Intelligent Toy Enhanced Learning (DIGITEL 2007)* (Los Alamitos, CA, March 2007), IEEE Computer Society, pp. 49–56.

II I. Jormanainen and A. Harfield. "Supporting the teacher in educational robotics classes: work in progress". In *The 16th International Conference on Computers in Education (ICCE 2008)* (October 2008), Asia-Pacific Society for Computers in Education, pp. 931–934.

III I. Jormanainen, A. Harfield and E. Sutinen. "Supporting teacher intervention in unpredictable learning environments". In *The 9th IEEE International Conference on Advanced Learning Technologies (ICALT)* (2009), IEEE Computer Society, pp. 584–588.

IV I. Jormanainen, M. Beynon and E. Sutinen. "Understanding open learning processes in a robotics class". In *The 9th Koli Calling International Conference on Computing Education Research* (2010), Uppsala University, Sweden, pp. 51–54.

V I. Jormanainen, M. Beynon, and E. Sutinen. "An abductive environment enables teachers intervention in a robotics class". In *The Proceedings of 17th International Symposium on Artificial Life and Robotics (AROB 2012)* (2012), pp. 1075–1078.

VI I. Jormanainen and E. Sutinen. "Using data mining to support teacher's intervention in a robotics class". In *The IEEE Fourth International Conference on Digital Game and Intelligent*

Toy Enhanced Learning (DIGITEL) (2012), IEEE Computer Society, pp. 39–46.

VII I. Jormanainen and E. Sutinen. “Role blending in a learning environment supports facilitation in a robotics class”. *Journal of Educational Technology and Society* (2013). Accepted for publication.

Throughout the overview, these publications will be referred to by Roman numerals. The publications will be linked to the research questions in Chapter 2. The original publications have been included at the end of the printed version of this thesis according to the agreement with the respective copyright holders.

AUTHOR'S CONTRIBUTION

The publications selected in this dissertation are original research papers on educational robotics. The author was the main contributor to the manuscripts. Original ideas and outline for technical implementations presented in Paper I and Paper VI emerged through discussion with Professor Sutinen. The author carried out the development of the agent architecture for collecting data from the students' programming environment for Paper I. The implementation of collecting data from the robots was contributed by Yuejun Zhang. His work has been presented more deeply in [70]. A demonstrative version of the classroom model presented in paper Paper II was contributed by Dr. Antony Harfield. The author was the main developer of the Empirical Modelling artefacts and data mining environment presented in Papers III - VII. Dr. Meurig Beynon worked on deploying the OME modules to an HIV/AIDS educational game presented in Paper III. In all publications, the author carried out qualitative and quantitative analysis as well as interpretation of the results. The author has written the manuscript to all papers; in all papers the co-operation with the co-authors has been significant.

LIST OF TERMS AND ABBREVIATIONS

Android An operating system for mobile devices from Google.

Automatic rule generation A technique applied in this research to define rules that classify learning process. The rules can be generated manually or automatically for example by using data mining.

Abductive reasoning A knowledge building approach aiming to generate the best possible explanation to the phenomenon under observation [59].

CLE Conflative Learning Environment approach - a set of design and implementation principles that proposes how learning environments for unpredictable learning settings could be constructed.

Data Mining An interdisciplinary sub-field of computer science aiming to discover previously unknown knowledge from large and incoherent data sets [68].

Decision tree A data mining scheme to represent decision rules in a tree-like structure. Each node of a decision tree contains a condition, and tree branches lead to leaves of the tree where the final states of the phenomenon can be found [55].

DoNaLD Empirical Modelling notation for 2D line drawings [15].

Educational robotics Small, relatively cheap, and easily accessible construction sets for building and programming autonomous devices [60].

Empirical Modelling (EM) - a modelling approach for building interactive, computer-based models, developed at University of Warwick, UK [15].

EDDI EDEN Database Definition Interpreter, a database notation in the Empirical Modelling environment [15].

EDEN The core notation of the Empirical Modelling tools with C-like syntax [15].

EUD End User Development - a software development approach where end-users of software are involved in the development process [44].

Facilitation Understood in the context of this research as teachers' efforts to support their students in a learning environment. Project-based and student-centred learning environments change teachers' role, and facilitation in these environments is more mentoring than traditional teaching.

FIPA Foundation for Intelligent Physical Agents.

HTML5 A recent version of the markup language for building web pages and applications.

iOS An operating system by Apple Inc for mobile devices.

IPPE Instructive Portable Programming Environment [33] is a dedicated programming environment for Lego Mindstorms RCX robots. The IPPE was used as a primary programming platform for the experiments in this research.

Java An object-oriented programming language developed by Sun Microsystems.

JavaScript A scripting language for building interactive web-applications that run in users' web browser.

JS-EDEN A web-based variant of the Empirical Modelling environment [28].

K-12, K12 Often used term for modern information technology education initiatives aimed for school kids and students from kindergarten to age of 18 - 19 years ("12" refers to the 12th grade in US school system).

Kids' Club A collaborative research laboratory where school children work together with university students and researchers of computer science and education [16]. Organised for more than 10 years by University of Eastern Finland (formerly known as University of Joensuu).

Lego Mindstorms An educational robotics set by LEGO Group [20]. The first generation Mindtorms RCX set was mainly used in this research.

Learning Environment Understood in this research as a combination of physical and virtual environments where formal and informal learning takes place.

OME Open Monitoring Environment - a monitoring environment for following the students' progress in educational robotics settings. A major development contribution of the research.

Scout A notation for describing screen layout in the Empirical Modelling environment [15].

STEM Science, technology, engineering, and mathematics.

tkeden An interactive interpreter for the Empirical Modelling environment [15].

Unpredictable learning environment Understood in the contex of this research as a learning environment where a variety of learning paths exist, and learners proceed differently with their tasks. The students' activities in the learning environment cannot be predicted easily and teachers may find it difficult to follow students' work.

Windows RT An operating system for mobile devices from Microsoft.

LIST OF FIGURES

LIST OF TABLES

Contents

1 Introduction

New learning environments based on modern technology set new kinds of professional expectations for teachers. Besides facilitating students' learning, teachers should be able to use technology effectively throughout curriculum. Also, traditional views on learning, where all students progress synchronously through predefined steps and reach the learning goal approximately in the same way and same time, are changing towards more asynchronous and varied learning paths where emphasis is on students' project oriented learning [24]. Teachers working in these learning settings are, however, often uncomfortable how to facilitate students' learning process or even how to assess learning outcomes [48].

Educational robotics is an example of the modern learning environments that provide attractive possibilities for teaching at various school levels and subjects [12]. However, teachers working in a robotics class often face the fact that following student groups' progress is difficult because the groups usually take different and unpredictable paths when solving the tasks that are usually open-ended and allow room for students' creativity when planning and implementing the project. The teachers are required to manage often unexpected situations and problems in the classroom [12]. Furthermore, different phases of the students' problem solving processes produce useful information that the teacher should be aware of. This leads to a situation where the teacher does not necessarily notice important events that essentially define the progress of the learning processes.

In line with the new expectations for the teachers, the learning environments should provide support for the teachers on facilitating asynchronous and unexpected learning in the classroom. For example, teaching and learning processes in an educational robotics setting are not static. Students do have different preferences for example when it is an appropriate time to start programming. Some students and groups prefer to program the robot as soon as possible, even when the robot's construction is not fully completed, whereas others spend longer time on finalizing all details of the robot's physical construction before the first attempts on programming. Hence, the learning environments supporting these processes need to be adaptable. This means that applications need to be open for a constant refinement that goes beyond the normal use of a learning environment or an application in the classroom, that is, blending the roles between an end-user (a teacher or a student) and a software developer.

Role blending requires a development approach that allows constant changes to the applications in on-the-fly basis. As a result of our research, we devised a *Conflative Learning Environment* (CLE) approach. The CLE defines characteristics, working methods, and implementation principles for a learning environment that allow role blending for example between a teacher and a software developer. According to the CLE approach, the development process of a learning environment takes place through iterative modelling cycles instead of conventional software development processes with strict roles of a developer and a user and a pre-defined product life cycle.

The main result of this research, an *Open Monitoring Environment*, was developed by following the requirements of the CLE approach. The OME aims to improve teachers' possibilities for intervention at the right time in educational robotics classes by allowing the teacher to monitor and model the learning process based on the experimental data rising from the current learning setting. The OME automatically gathers data from the educational robotics classroom and processes this data to collections and visualisations that are relevant for the current learning context. The collections and visualisations are based on the rules that are developed for classifying students' progress and detecting unexpected events. The rules are refined constantly throughout the usage process of the monitoring environment by a technical modeller (developer) and a pedagogical modeller (teacher). The rules are open for the teacher's revision and he or she can interpret the events in the classroom by exploring the learning progress classifications based on the rules, and modify the rules freely to make them meaningful for the current teaching context, taking into account for example the order of actions that the students take during their project.

However, the amount of data that can be extracted from the educational robotics learning process is too large and complex for manual processing. Deriving the rules from the data flow requires semi-automated tools that can be used to initialise the rule creation and refinement process. As a part of the OME, we implemented an open data mining functionality that enables transparent and active rule creation by allowing the teacher to classify a sample set of the events for the data mining algorithm (so called *training set*), and to control the classifier creation process. A decision tree classifier is created upon teacher's manual initiation in an appropriate moment of the learning activity, and the created classification rules are used to bring visual clues for the teacher about the student groups' progress. The semi-automatically created classifier can be used as a starting point for iteratively building a contextually meaningful support environment.

A visual representation of the classifier rules in the form of the decision tree in the OME shows patterns of actions that the students may go through with their robotics project. In this way, the OME can be used not only for predicting the progress, but also for exploring the past event and reasons behind them, hence supporting a teacher's *abductive reasoning* process. Abduction has been seen as a potential approach in informatics when facing vague or even unrecognised problems [59]. Traditional data mining applications in learning environments keep the classification processes in a black box and only the results are visible for the users of the learning environment. In the OME, the teacher uses his or her pedagogical expertise and experiments with data mining tools to build a set of rules and visualisations that helps him or her to interpret the current learning process. By opening up the data mining process for the teacher, the environment supports teacher's abductive reasoning when he or she is exploring explanations behind the current events in the classroom.

In this work, we have used *Empirical Modelling* (EM) tools [52] to build the OME. The EM tools form a unique modelling environment where the end-user is taken out from the traditional role of using an application. The EM is a collection of tools and principles for modelling real-world phenomena based on modeller's empirical observations. The model is captured based on a modeller's experiences about the state of the subject (”state-as-experienced” approach) [52]. An essential feature of the EM approach is that, after an initial definition, the EM environment automatically keeps the model updated according to specific links between different parts of the model. This is similar to spreadsheet applications where values of the cells are updated automatically according to formulas that might contain references to other cells. Use of Empirical Modelling and different concepts connected to it are illustrated in Section 4.3. In EM, the working environments for the developer and the user are essentially the same, and all changes to the EM model are effective immediately, allowing the user to adjust in principle all details of a model. In the OME, this feature of the EM tools is used especially to enable learning process modelling with real-time events and rules derived from them, in contrast to traditional Intelligent Tutoring Systems (ITS) where the learning modelling is usually based on pre-defined, and hence, static rules. The EM tools, such as *tkeden* and various notations of the EM environment, as well as working processes of the EM, are well aligned with the CLE approach. EM environment enables building learning environments that are bound to the current classroom scenario in an open and flexible way. Figure 1.1 illustrates how the key concepts of the research relate to each other.

We have tested the OME in various educational robotics settings in Finland and South Africa. Evaluation shows that the OME provides suitable support for teachers who are novices in an educational robotics classroom. The teachers were able to use the OME to support their pedagogical expertise to identify the prob-

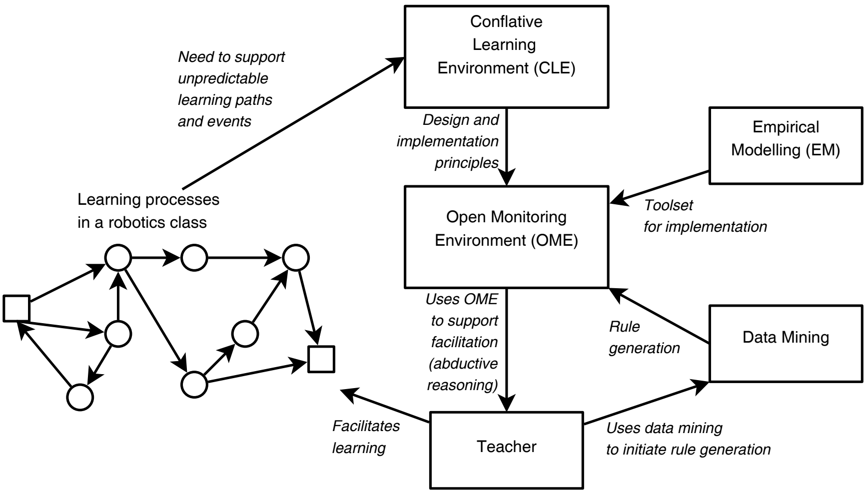


Figure 1.1: Key concepts and their relations

lems otherwise unnoticed and intervene to student groups' work. From a technical point of view, the results show that the implemented OME architecture is suitable for collecting and processing data from a robotics classroom, and data mining methods implemented as a part of the OME are efficient for classifying students' progress.

This research has been conducted by following a development research framework [58]. The analysis (literature reviews, data collection, experiments, interpretation of results, reporting) and development work (need analysis, software development, modelling, testing) have followed each other in iterative cycles, where the result of a completed stage serves as an input for the following stage. Moreover, the CLE approach itself adopts features similar to the development research framework, and especially experimental parts of this research have followed the CLE approach.

Ilkka Jormanainen: Supporting Teachers in Unpredictable Robotics Learning Environments

2 Questions and Methods

2.1 RESEARCH QUESTIONS

The research objective of the thesis is to *develop a concept and appropriate tools to support teachers' facilitation process and intervention strategies in unpredictable learning environments, such as educational robotics classes*. The research problem was divided into four detailed research questions, which were answered by conceptualising the Conflative Learning Environment (CLE) approach, and by developing an Open Monitoring Environment (OME). The OME was tested by experimenting with it in various educational robotics settings in Finland and South Africa. The lessons learned during the experiment, together with research literature, guided the iterative design-based research process. The detailed research questions involved in the process were as follows.

Q1: *What are teachers' expectations for modelling learning progress in educational robotics environments?*

Educational robotics as a learning environment leads often students to take unpredictable paths to problem solving. The robotics projects and problems in them are often open-ended, comparing to the closed problems in a more teacher-driven learning environment. The teachers working in educational robotics environments face problems in following the students' progress especially when they work in small groups. Traditional Intelligent Tutoring System (ITS) approaches and tools do not necessarily fulfil teachers' requirements, as they only offer the teacher a set of predefined options for interaction, and options to modify the environment to support diverse learning settings are limited. Question Q1 is answered in Chapter 3 by identifying the teaching challenges in existing educational robotics settings.

Q2: *How is it possible to conceptualise the modelling process?*

The continuously changing learning setting needs to be *modelled* based on empirical observations rising from the current setting. To realise an environment that supports empirical modelling of a learning progress, there is a need to conceptualise the processes related to building and using the environment. We have formulated a Conflative Learning Environment (CLE) approach, which defines these processes. The CLE suggests that roles, tasks, and working environments of different actors in the learning environment development and usage processes should be blended. It is, however, obvious that different actors cannot, and even should not, adopt others' tasks completely, and there is a need to make a distinction between pedagogical and technical modelling. By answering this question, we give suggestions about the level of the role conflation between the teacher (pedagogical expert) and the developer (technical expert) in the CLE approach. Question Q2 is answered in Chapter 3.

Q3: *What are technical requirements for implementing a learning environment based on the CLE approach?*

The Open Monitoring Environment (OME) was developed and evaluated as a part of this research. The OME is an example of a learning environment whose development and usage follow the CLE approach. The OME is a major development contribution of this research. The OME implementation contains a set of agents for collecting data from the learning process by observing how students interact with the IPPE programming environment and Lego robots. The teacher follows and manipulates data in a modular monitoring environment. However, it is crucial to notice that the OME was developed for monitoring the students in a specific robotics setting, and it is likely that the environment does not fulfil properly the requirements of another educational robotics setting. This issue is discussed in the light of the CLE approach, and question Q3 is answered in Chapters 4 and 5.

Q4: *How does the OME support teachers' work in unpredictable robotics settings?*

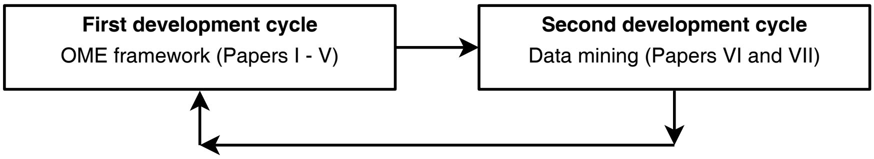
The OME environment was evaluated in various robotics settings. First, the evaluation focused on technical issues, such as how the agents work as data collectors and how data mining techniques could be implemented as a part of the OME. Secondly, we studied how the OME supports the teachers' abductive reasoning and students' facilitation in educational robotics classrooms. Question Q4 is answered in Chapter 5.

2.2 RESEARCH PROCESS

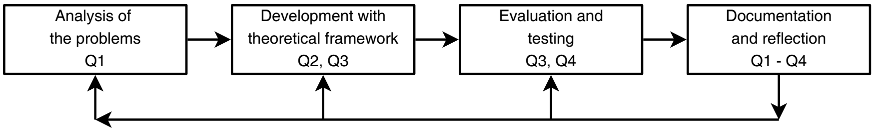
The research questions presented above were answered by following a *development research framework* [58]. This framework emphasises iterations where an application or a concept is developed based on existing theories and results from the continuous evaluations. The research process took two major cycles (Figure 2.1b) within the development research framework (Figure 2.1a). During the the first cycle (covered mostly in Papers I - V), we developed and evaluated the basic functionalities of the OME without having the automated rule generation for classifying students' learning progress. The second cycle consisted of development and evaluation of rule generation with the help of data mining (covered in Paper VI and Paper VII). Within the two major cycles, several minor development cycles existed. Figure 2.1b also illustrates how research questions (Q1 - Q4) were under consideration in specific phases of the research process. Similar phases of the research can be identified from both development cycles, and the description in Figure 2.1b can be generalised for both cycles.

2.3 RESEARCH METHODS

Several research methods were used in order to answer the questions presented in Section 2.1. Table 2.1 shows how different research methods have been used to answer the specific questions. Furthremore, Table 2.1 connects research questions to the original articles and chapters of this thesis. Classification of the research



(a) Two cycles of the development research framework



(b) Structure of a cycle in the development research framework (modified from [58])

Figure 2.1: Development research approach used in this study

methods is based on [22].

Table 2.1: The research methods used in this thesis

Question	Method	Main articles	Chapter
Q1	Literature review	I, II	3
Q2	Conceptual analysis	II, IV, V	3, 4
Q3	Concept implementation	I, III, IV, VI	4, 5
Q4	Case study	III, V, VI, VII	5

Literature review was used to study relevant background knowledge about different educational robotics settings and learning environments. The main sources were the most common literature databases in computer science and educational technology, namely ACM and IEEE digital libraries. The systematic review focused on the following journals and conferences with an appropriate mix of the keywords *educational robotics*, *K-12*, *programming*, *data mining*, *monitoring*, *decision tree*, *learning environment*, *teacher support*, and *learner modelling*.

- Journal on Educational Resources in Computing (JERIC), vol. 1 - 8, years 2001 - 2008

- ACM Transaction on Computing Education (TOCE, formerly known as JERIC), vol. 9 - 12, years 2009 - 2012
- IEEE Transactions on Education, vol. 46 - 56, years 2003 - 2013
- Journal of Education Data Mining, vol. 1 - 5, years 2009 - 2013
- Proceedings of International Conference on Educational Data Mining (EDM), years 2008 - 2013
- Proceedings of IEEE International Conference on Wireless, Mobile, and Ubiquitous Technologies in Education (WMUTE), years 2007 - 2012
- Proceedings of IEEE International Conference on Digital Game and Intelligent Toy Enhanced Learning (DIGITEL), years 2007 - 2012
- International Conference on Intelligent Tutoring Systems (ITS), years 2002 - 2012

Furthermore, the literature analysis was completed with additional articles from international peer-reviewed journals and conference proceedings as well as published books. These items are referenced in the appropriate places in the introduction and original articles.

Question Q1 was answered mainly by research reviews. The research literature was used to analyse factors that have led the reported educational robotics experiments to succeed or fail. Furthermore, the analysis of the problems in the robotics environments was supplemented with the experiences that I have gained during the development and utilisation of the Kids' Club [18] in 2002-2012 in various school and after-school settings.

Question Q2 was answered by analysing the working processes of researchers, software developers, and teachers who were involved in the research. Data was collected and analysed in several case studies that were conducted during the research process (Table 2.1).

Question Q3 was answered by implementing the Open Monitoring Environment based on the CLE approach. Answering this question required various software development and modelling tasks, for example developing agent architecture to collect data from the robotics environment and processing information in the OME. The primary tool for the implementation was Empirical Modelling. Also, Java was used to implement the mechanisms for data collection.

Question Q4 was answered by conducting four case studies (referred as CS1 - CS4) in real educational robotics settings. A variety of appropriate data collection and analysis methods were used in each experiment. The collection methods include observations, video recordings, focus group interviews, questionnaires, and log files of agent messages (students' interactions with the robotics environment). The video transcriptions, interviews, and answers for the questionnaires were analysed qualitatively. Data from log files was used while developing and testing the new features of the OME. In particular, the iterative development of the data mining module of the OME was driven by the results achieved by experimenting with data collected during the previous case studies.

Participants in the case studies were 10 - 15 years old primary and secondary school students (Table 2.2). The participants were individuals, i.e. same student, teacher, or instructor participated only in one study. The volunteer Kids' Club instructors in the studies CS1 and CS4 did not have pedagogical qualifications. The teachers in the studies CS2 and CS3 were professional teachers with varying background on using computers.

Tables 2.2 and 2.3 summarise design of the case studies. The case studies CS1 and CS4 were carried out in the Kids' Club research laboratory of University of Joensuu and University of Eastern Finland (see Section 3.1 for a detailed description of the Kids' Club). The case study CS2 was carried out in a boarding school in South Africa. Data that was collected during the case study CS2 was also used in case study CS3 for development of the OME and for analysing classroom activities retrospectively. The case studies CS1 and CS3 were divided to two phases. In both cases, the first

Table 2.2: Participants of the case studies

Study	Venue	Participants
CS1	Kids' Club of University of Joensuu	Phase 1: 8 students Phase 2: 4 students (10 - 15 years)
CS2	Boarding school, South Africa	4 teachers 12 students (grade 6 - 9, 12 - 15 years)
CS3	No specific venue (retrospective analysis)	Phase 1: no participants Phase 2: one teacher
CS4	Kids Club of University of Eastern Finland	2 instructors 13 students (10 - 13 years)

phase of the study was about developement of a technical feature of the OME, and the second phase involved human participants to test the developed feature. Detailed research designs for the distinct case studies with methodological and technical discussion, including reliability and validity, are presented in Chapter 5 and in corresponding original articles (Table 2.3).

2.4 THE MAIN RESULTS AND CONTRIBUTIONS

The main result of this research is the development and evaluation of the Open Monitoring Environment for educational robotics classes. The OME integrates well-known technologies, such as software agents for data collection and data mining for classifying students, by following the principles of the CLE approach, which was also formulated during the research process. Opening the technical implementation of the learning environment to the teacher means that he or she has in principle unlimited possibilities to modify the details of the environment to match the current context. The OME

Table 2.3: Summary of the research design

Study	Material	Collection	Method	Publication
CS1	Live interaction data	Data logging, observations	Mixed	Paper III
CS2	Live interaction data, video recording, field notes	Data logging, filming, focus group interview, observations	Mixed	Paper V
CS3	Logged interaction data, video tapes	Data logging, notes	Phase 1: Quantitative Phase 2: Qualitative	Paper VI
CS4	Live interaction data and decision trees, field notes	Data logging, observations, interviews	Qualitative	Paper VII

implementation that we used in the case studies opens especially the data mining process for the teacher and let the teacher to adopt the traditional roles of a software developer or a domain expert to his or her own work.

The main results achieved by evaluating the OME show that teachers working in robotics classes benefit from using the environment as they are able to model the learning process so that they notice students' problems that most likely would have left unnoticed without support from the OME.

The research work presented in the thesis is a collection of seven articles published in international peer-reviewed conference proceedings or journals. I have been the main author in all of these articles.

Paper I: I. Jormanainen, Y. Zhang, Kinshuk and E. Sutinen. "Pedagogical agents for teacher intervention in educational robotics classes: Implementation issues". In *the First IEEE International Workshop on Digital Game and Intelligent Toy Enhanced Learning (DIGITEL 2007)* (Los Alamitos, CA, March 2007), IEEE Computer Society, pp. 49–56.

In this article, we presented a fundamental problem for this research: How to keep track of individual students' or even small groups' progress in an educational robotics class of 30-40 students. As a solution, we introduced a multi-agent environment to monitor students' interaction, robots' movements, and the construction and programming process of robots. An initial outline for the Open Monitoring Environment was presented (however, the name 'Open Monitoring Environment' was not yet proposed). Technical contribution created a base for building the agent architecture for later implementations of the OME.

Paper II: I. Jormanainen and A. Harfield. "Supporting the teacher in educational robotics classes: work in progress". In *the 16th International Conference on Computers in Education (ICCE 2008)* (October 2008), Asia-Pacific Society for Computers in Education, pp. 931–934.

The paper presented the first conceptual outlines of the OME. Building strongly on Paper I we explored approaches that the teacher can use to observe and decide when and how to intervene in a classroom situation. As an alternative approach to traditionally developed programs, Empirical Modelling was introduced as offering a working environment in which the teacher can construct an artefact that reflects their personal understanding of the rich situations that can arise in a robotics class.

Paper III: I. Jormanainen, A. Harfield and E. Sutinen. “Supporting teacher intervention in unpredictable learning environments”. In *the 9th IEEE International Conference on Advanced Learning Technologies (ICALT 2009)* (2009), IEEE Computer Society, pp. 584–588.

The contribution of the paper was two-fold. First, we studied the extent to which the agent architecture proposed in Paper I suits for data collection from real educational robotics environments. Secondly, we studied how the teacher can use the *conflative learning environment* built with the Empirical Modelling tools to see if the students have understood the task-related rules that need to be learned in order to complete the task successfully (as proposed in Paper II). Two case studies were conducted. The results confirmed that agents are useful and effective in data collection, and that the Empirical Modelling environment can be used to construct the working classroom models. An initial and simplified version of the OME was presented.

Paper IV: I. Jormanainen, M. Beynon and E. Sutinen. “Understanding open learning processes in a robotics class”. In *9th Koli Calling International Conference on Computing Education Research* (2010), Uppsala University, Sweden, pp. 51–54.

The technical contribution of the paper focused on creating a functional implementation of the OME that was used for evaluations presented in Papers V - VII. The implementation was based on the initial version presented in Paper III. This implementation of the OME was also used to deepen the concept of a Conflative Learning Environment. A distinction between pedagogical modelling and technical modelling was introduced (cf. research questions Q1 and Q2). As another result, we noticed that the OME implementation and especially data collection methods and learning process reconstruction tools of the OME are especially well-suited for deployment in other application areas.

Paper V: I. Jormanainen, M. Beynon, and E. Sutinen. “An abductive environment enables teachers intervention in a robotics class”.

In *Proc. 17th International Symposium on Artificial Life and Robotics (AROB 2012)* (2012) pp. 1075–1078.

The main result of the paper shows that the OME based on the CLE approach allows teachers to blend design and development activities with their work. In consequence, they can use their pedagogical expertise effectively to analyse their students' progress, as informed by the OME. The study reported in this paper was conducted in a South African primary school by using a modified implementation of the OME that was presented in Paper III.

Paper VI: I. Jormanainen and E. Sutinen. "Using data mining to support teacher's intervention in a robotics class". In *4th IEEE International Conference on Digital Game and Intelligent Toy Enhanced Learning (DIGITEL 2012)* (2012), IEEE Computer Society, pp. 39–46.

In this paper we report a case study where various data mining methods were tested with authentic data that was collected from the educational robotics setting in a South African school (Paper V). Results indicated that decision trees are effective for classifying automatically students' progress in the educational robotics environment. Based on this result, we proposed a novel approach where initial rules produced by the classification algorithm are open for revision by the teacher. By using his or her pedagogical expertise, the teacher adjusts the parameters of the data mining process and experiments with different rules to build a set of rules that defines the current learning process. In this way, the OME supports the teacher in abductive reasoning by providing tools to derive the reasons behind the events in the classroom. The results of the paper guided our further work with the OME, leading to an implementation with a more comprehensive set of data mining techniques to cover more complex data available in educational robotics settings.

Paper VII: I. Jormanainen and E. Sutinen. "Role blending in a learning environment supports facilitation in a robotics class". *Journal of Educational Technology and Society* (2013). Accepted for publication.

In the paper, we used the same OME version that was used in Paper VI to analyse and model students' progress in an educational robotics class in real-time, in contrast to the experiment in Paper VI. The results show that the data mining features of the OME can be used to predict and, hence, support learning processes also with real-time data. We also noticed 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. Also, in accordance with the principles of the CLE framework, the instructors were able to modify the learning environment to match the current context in a way that goes beyond normal teacher activities in a classroom.

The results of the paper strengthen the idea of the CLE framework that role blending between a teacher and a software developer in a learning environment provides a novel way for building personalized and contextualized support environments.

2.5 STRUCTURE OF THE THESIS

This thesis is organised as follows. Chapter 3 introduces background relevant to our work. Then, modern educational robotics environments and the Empirical Modelling approach and tools with illustrative examples are introduced. Chapter 3 also describes challenges that teachers face when working in the robotics environments and reviews proposed solutions that meet these problems. To position our work, Chapter 3 presents a scheme for categorising learning environments according to how they utilise learning and learners' modelling.

Chapter 4 presents the CLE approach and the OME in detail. The chapter also shows the steps that we have taken on developing the environment and concepts related to it. Chapter 5 analyses our work through several cases in which the OME has been used. The chapter provides evidence about the feasibility of our approach.

Chapter 6 interprets the results from the viewpoint of the literature and the original research questions. The chapter also discusses the impact of this research for educational robotics classrooms especially in K-12 educational robotics settings. Finally, Chapter 7 summarises the results and concludes the thesis. The chapter also identifies directions for future work.

3 Robotics as a Learning Environment

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 tangible learning environments [61]. Technical subjects, such as electronic engineering [49] or programming, have been taught with educational robotics. Educational robotics has been used also to increase motivation of children in technology related school activities [11] or to promote STEM-based outreach programs [10]. Williams argues that Lego Mindstorms are qualitatively effective as educational technology for computer engineering in general, not only for robotics education [67].

However, successful use of educational robotics requires new kinds of classroom settings 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 [19]. Correll *et al.* [9] propose solutions to the recognised issues related to deployment of large-scale robotics classes, such as scalability of a learning environment, or complexity of robot constructions that may prevent learning of the subject.

In this chapter, we present Kids' Club as an example of robotics learning environments for K-12 education. Kids' Club provides also an important context for our research work as described in Chapter 2.

3.1 KIDS' CLUB AS A LEARNING ENVIRONMENT

Kids' Club is a collaborative research laboratory where school children work together with university students and researchers of computer science and education. Kids' Club was founded in October, 2001, at the University of Joensuu, Finland, (nowadays University of Eastern Finland), with the following aims and motivations [16]:

- Fostering children's interest in creative design and problem solving with novel ICT-flavoured artefacts;

- Developing novel tools and approaches for understanding technology and learning science; and

- Encouraging children, particularly girls, to consider their future careers in ICT and related fields.

Kids' Club was launched to address the need for an educational technology research platform and the need for novel technology education practices in curriculum and after-school contexts. The Kids' Club has been active since 2001, and the original needs are perhaps more topical than ever. National school curricula are undergoing changes in many countries. For example, the Finnish government has published plans for a new national curriculum[1] with a strong emphasis on ICT as an extensive element throughout the whole curriculum. In another initiative, the British Department of Education has announced that Computer Science will be a science option in English Baccalaureate, alongside with mathematics, physics, and chemistry[2]. This type of development sets new expectations for the learning environments and teachers' professional knowledge.

Dissemination of Kids' Club technology education concept has led to an international network of schools and after-school clubs working with various kinds of educational technology projects. Figure 3.1 shows Finnish school kids working with an educational robotics project. In Figure 3.2 school kids design and implement a soccer game with the Scratch programming language [45] during

[1] http://www.oph.fi/english/102/0/ops2016_renewal_of_the_core_curriculum_for_pre-primary_and_basic_education, accessed April 27, 2013

[2] https://www.gov.uk/government/news/computer-science-to-be-included-in-the-ebacc, accessed April 27, 2013

Figure 3.1: Children working at Kids' Club with robotics projects

Kids' Club activities in Nampula, Mozambique. In the Finnish networks and schools alone, more than 500 school kids get involved with robotics activities annually.

For *children*, Kids' Club appears as an after-school technology club with an opportunity to learn interesting skills in a playful, non-school like environment which lets room for creative ideas and alternative approaches for problem solving [18]. For *researchers*, Kids' Club provides a platform for developing novel methods and applications of educational technology. Often the needs and ideas for the research and development arise from the Kids' Club activities.

For example, Kids' Club instructors were in need of formal evaluation methods of the projects and learning outcomes. Several tools, such as web and paper forms were used at the beginning to conduct questionnaires, but these methods did not give the expected information for researchers. Furthermore, the reflection processes of the children were not deep enough for the meaningful

Figure 3.2: Children working at Kids' Club in Mozambique

learning. For this reason, a spin-off research project was started to find a solution for the problem. As a result, the virtual reflection environment Virre [17] was developed first as a computer application and later it was embedded inside a big teddy bear. The Kids' Club has proved to be an efficient research and development platform especially in the field of special education [37].

Research methods in the Kids' Club environment are mainly based on action research. Separate studies are realised according to their own research plans, and the researchers are involved in the club activities while conducting their studies. As examples, the following concrete research questions have been studied at Kids' Club:

- How to visualize the programming with the robotics? [33]
- How does the technology-rich environment affect special education? [37]

- What are new forms of collaboration between children and companies in the field of technology? [56]
- How does the virtual reflection environment Virre help in the learning process? [57]

Tutors in Kids' Club are mainly undergraduate students, who voluntarily take part in club activities by helping children with their tasks. The tutors are often students of computer science or education, and they can also complete training periods, courses, or project work at Kids' Club. It is remarkable, that robotics activities including Kids' Club in school settings, change a teacher's traditional role from teaching towards fostering collaborative learning amongst the students [24].

Kids' Club is based on the concretisation of socio-cultural and constructionist [51] views on learning. Robots and other concretisation tools allow the children to make their thoughts and mental models explicit and easier to manipulate in the physical world. These tools encourage children in inventive learning which, in turn, emphasises creative and open problem solving and the creation of new artefacts through the evaluation and comparison of different solutions. Learning at Kids' Club is based on cyclic problem based learning processes (Figure 3.3), where the creation of physical artefacts as possible solutions is stressed.

The technical environment of Kids' Club contains several software and hardware tools to support activities in different phases of learning. Some of the tools are ready-made commercial or free software or hardware, whereas others have roots in the research projects conducted at Kids' Club. For example the programming environment *IPPE* [33] and the virtual reflection environment *Virre* [17] have been developed in the Kids' Club context. Components of the technical environment can be divided into four categories according to the iterative life cycle of the club activities (Figure 3.3). Computers are usually used in all phases, and the computer environment is based on laptops and wireless network for a maximised flexibility of the setting.

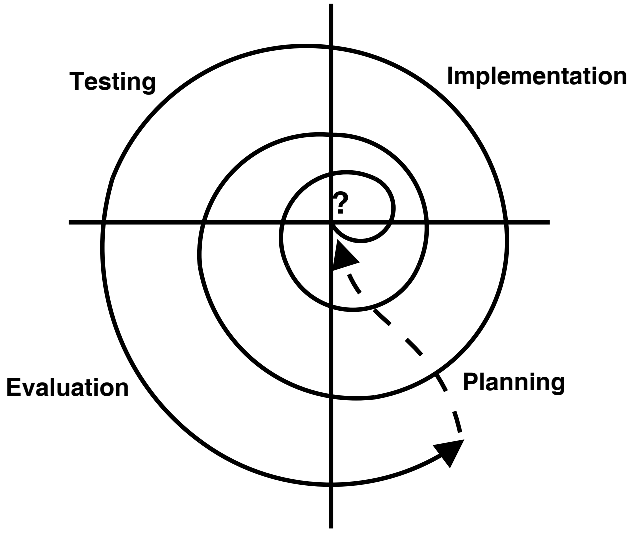


Figure 3.3: The life cycle of a typical Kids' Club project

Usually the first phase of Kids' Club project is *planning*. Several creative problem solving methods and various analogue and digital tools, such as pen and paper, white board, active boards, and mind mapping software, have been used for planning purposes.

The *implementation* phase of a Kids' Club project typically involves building a physical artefact, for example with an educational robotics set. Lego Mindstorms Robotics Invention System [4] (Lego RIS) and its successor Lego Mindstorms NXT [20] are flexible and complete construction kits, which enable children to design and build their artefacts based on instructions or their own imagination. Children are encouraged to combine the robotics kits with other materials in order to give a personal flavour to their robots and models. In addition to the robotics kits, other kinds of hardware have been used in the Kids' Club projects, including an automatic door, a fingerprint reader, different sensors, mobile phones, and touch screens.

An important part of the implementation phase is programming. The programming environment is selected based on the skill levels of the participants and the needs of the current project. For example, Lego RIS, Lego NXT-G [38], and IPPE [33] programming environments for LEGO robotics and the TileDesigner environment for Soccer Robo robots have been used. When the children have learnt basic skills of programming, they can also progress to more advanced programming environments, such as Java [42] or NQC [4].

When the project is ready, it is *tested* and *evaluated*, and documented. For these phases, children typically use digital cameras, video camcorders, presentation software, posters, and websites. In this way, the participants are encouraged to share new knowledge and experiences with each other.

3.2 TEACHING CHALLENGES AT KIDS' CLUB

Learning environments, such as educational robotics, based on project-based pedagogy [46] often lead students to take unpredictable paths for solving open-ended problems. Development and dissemination of Kids' Club in various school and after-school settings has shown us that these kinds of learning settings pose a particular challenge for teachers - how to follow all students' or one individual's activities. The project-based working methods, open-ended tasks, different problem solving strategies, group dynamics, and students' different roles in the groups, and the iterative nature of robotics projects easily lead the students to take different paths and rhythms in their work. This makes it difficult for the teacher to detect the students' problems and the right moments for intervention. Obviously, the problem becomes more complicated when the number of students and the number of small project groups increase.

Cyclic processes in the educational robotics setting generate various asynchronous information streams for the teacher at any particular moment. For example, one group starts the task by designing the robot carefully, whereas another group applies a trial-and-error approach where it experiments with different robot constructions. Some groups even start by addressing the programming aspect of the task, and they build the robot only after the program is ready. All of these different phases and tasks within them produce specific kinds of information that the teacher must process in order to notice possible problems in the classroom. Figure 3.4 presents the phases with examples of possible tasks and information outputs.

When organizing the Kids' Club activities at the university and in schools in Finland and abroad, we have analysed teachers' and instructors' problems and noticed that there is a need for tools to provide support facilitation especially when unpredictable events occur. However, it is clear that not all data can be monitored automatically. For example, data from the planning phase may include hand-written notes and drawings, and these processes are hard to monitor completely automatically. In the learning environments involving educational robotics, most of automatically collected data is extracted from the implementation phase.

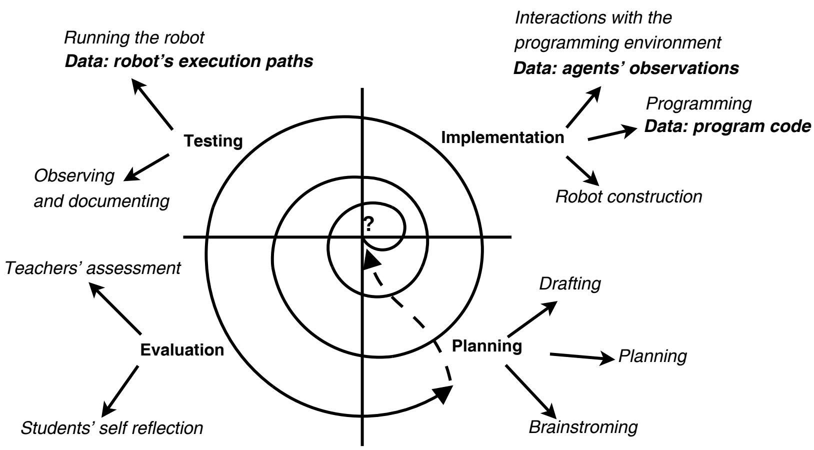


Figure 3.4: Phases of an educational robotics project with examples of tasks and generated information

3.3 EXISTING TOOLS FOR FOLLOWING THE STUDENTS

Several solutions for following the students automatically or semi-automatically can be found in the literature. Intelligent Tutoring Systems (ITS) [50] use widely a distributed agent approach to gather data and inform the teacher about the possible difficulties in a learning setting (for example, see [47] and [63]). These systems give support for observing the overall progress of all students in the classroom, detecting the problems that the students have, and analysing the actions of a particular group or even an individual student. Despres and George's research [13] is probably one of the closest to our approach in the educational robotics domain. They describe an ITS that allows the teacher to follow the students' activities in an educational robotics classroom. In the first instance, their system provides support directly to the students, and if that fails, the system reports the difficulties to the teacher as well. Our monitoring environment applies technical features that can be found in the literature. For example, the OME, like several other existing agent systems, uses a FIPA (Foundation for Intelligent Physical Agents) compliant language to describe the interaction protocols between the agents.

In previous research, the idea of using software agents to observe and monitor students' activities has been applied mostly in a theory-driven way. The environments, based at least loosely on Intelligent Tutoring Systems (ITSs), have offered a selection of choices by which the monitored data is classified and used for modelling the students. The main difference to the OME is that, whereas traditional ITS applications use a theory-based approach for building the learning model, the OME starts from the empirical observations arising from the current learning situation. There do exist ITSs that apply empirical approach for building the learner model (for example *Wayang Outpost*, a multimedia ITS for geometry by Cooper et al. [8]). However, the learning models in these systems are still at least partially predicted based on theoretical assumptions. For example, a given set of features is used for classifying the user's emotional self concept in [8].

Data mining tools have a recognised status as a part of modern learning environments. Data mining techniques have been widely applied in learning environments for modelling learners and estimating learning processes. Usually, data mining is used to extract knowledge from e-learning systems through the analysis of the data that the users have generated [7]. Most of the work in data mining in educational systems contributes to students' assessment (for example [14, 23]) and course adaptation based on students' learning behaviour [40]. Clustering and classification are the most widely used used data mining techniques in learning environments [7] because of the nature of the problems that often appear in learning environments. Also, data distillation for human judgment and discovery with models have been prominent and increasingly popular methods for educational data mining [3]. Only few systems have tried to make logging data directly useful and available to teachers or instructors [66]. Learning environments with an open data mining approach, where data mining processes and results would be explicitly visible to the users are rarely reported in the research literature. Rather than that, the data mining takes place in a black box and the results are only implicitly visible to the users.

Systems presented in the literature are usually based on an approach where a domain expert manually labels data sets and builds models describing the learning activities in advance. This is a time-consuming and error prone process especially in exploratory learning environments with open-ended problems, because prior definitions of relevant behaviours are necessarily not available for labelling data and training the model [1]. Amershi and Conati [1] present data mining solutions to support exploratory learning activities where there is no clear distinction between relevant and irrelevant behaviours of students. Their approach relies on data mining to automatically identify common interaction behaviours and using them to train a user model. Amershi and Conati [1] conclude that a potential problem of their approach is that it requires a substantial amount of data to work. We have shown that empowering teachers working in the robotics classes to direct the data mining process produces valuable insight into the progress of the learning activity even on relatively small datasets [Paper VI].

3.4 CATEGORISATION OF LEARNING ENVIRONMENTS

To set our work in the right context, we have divided learning environments into three categories according to the modelling approach that they use (Figure 3.5).

Model-based environments *deduce* a given learner's characteristics from a set of existing learning models, whereas in modelling-based approaches, a set of models is *induced* from empirical data about learners' progress.

The third category is explanation-oriented learning environments that use *abduction* for the modelling. *Abductive reasoning* has became topical recently also in the computing research community [59]. In an abductive learning environment, the modelling process starts from the current learning situation and the aim is to generate the best possible explanation for the current situation with a set of empirical observations and rules, as described in Figure 3.5. Ross concludes in [59]: *Perhaps abductive reasoning can help us discover what we are not even looking for.*

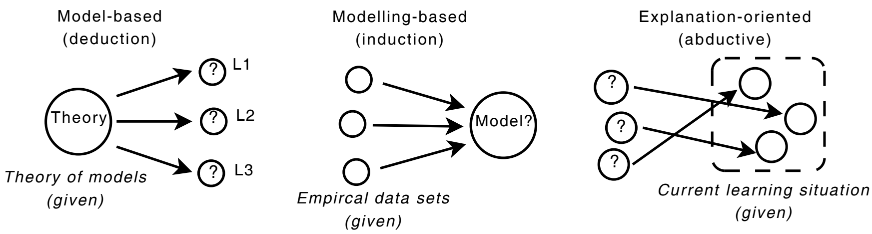


Figure 3.5: Deductive, inductive, and abductive approaches in learning environments

An abductive approach for building learning environments has been previously used by other researchers, for example by Qiu and Riesbeck in [54]. However, the users' incremental development in [54] focuses more on contributing material in the particular learning environment during the learning process rather than on building the actual learning environment as the users of the OME do.

Table 3.1 shows classification of learning environments between

Table 3.1: Comparison between the traditional ITS and conflative approaches

	Model-based	**Modelling-based**
Modelling approach	Theory-based, deductive	Empirical, inductive
Learning model	Given	Constructed
Adaptation	Black box	Transparent
Roles in the learning community	Separated	Blended
Working environments	Separated	Conflated
Direction of modelling	Top-down	Bottom-up
Modifications to the tools	Through the software development process	On demand in the actual learning situation

theoretical extremes of model-based environments and modelling-based environments (leftmost and rightmost columns in Table 3.1). The model-based environments are traditional ITSs that take deduction as a starting point for the learner's modelling. At the another extreme, inductive modelling-based environments take empirical observations arising from the current learning setting as a starting point for modelling. It is important to recognise that these are theoretical extremes, and real systems such as the OME supporting abductive reasoning, usually fall between these extremes.

Abductive reasoning in explanation-oriented learning environments needs to be supported with appropriate tools. As described earlier, learning and teaching processes for example in educational robotics classes are dynamic, and traditional monitoring tools or intelligent tutoring systems do not necessarily fulfil the demand for a flexible and easily modifiable working environment.

4 Roadmap to the Open Monitoring Environment

The development of the Kids' Club learning environment and various collaboration projects with schools on educational robotics have indicated that the learning environment should have features that provide support especially for novice teachers. The major development contribution of this research, *Open Monitoring Environment* (OME) was built by following a development research framework [58], and experimental work has played a key role in the process.

The following steps can be identified from the research project. At first, the problem of following students was indicated and an agent architecture (Figure 4.1) to monitor the students in the learning environment was proposed [Paper I]. The next step was to conceptualise teachers' working environment by using the Empirical Modelling approach and toolset [Paper II]. Several EM models were developed and their suitability to the educational robotics setting was tested [Paper III] with groups of school children in Kids' Clubs. In parallel with defining the functionality of the OME environment, the concept of Conflative Learning Environment was developed [Paper IV, Paper II]. The OME environment was finally tested in real educational robotics settings [Paper V, Paper III, Paper VI, Paper VII].

4.1 CHALLENGE 1: HOW TO COLLECT DATA FROM AN UNPREDICTABLE ROBOTICS ENVIRONMENT?

Software agents have long been applied in educational environments to provide learning support. Agents can monitor progress, give instruction when needed, help organize students' work, and provide feedback for tutors. However, there is no agreement in the literature on a unified definition for the term "agent". The term is used slightly differently according to the context where agents are employed. Agents exhibit properties, such as autonomy, social ability, responsiveness, and proactiveness, upon which a common consensus has been reached [69].

Sklar and Richards [62] categorise agents in human learning environments in three categories. *Pedagogical agents* are personalised assistants that interact directly with the lerner. *Peer learning agents* are interactive partners built into the user interface, but they are typically less engineered than pedagogical agents. *Demonstrating agents* are interactive mediums for learning, such as educational robotics [62]. Agents possessing mobility are called *mobile agents*. Mobile agents are usually software programs, which may be dispatched from one computer and transported to a remote computer for execution. The motivation for using mobile agents stems from a number of potential benefits, such as efficiency and reduction in network traffic, asynchronous autonomous interaction, interaction with real-time entities, local processing of data, and support for heterogeneous environments [39].

Furthermore, many systems employ agent architectures that are not explicitly visible for the user. The underlying system components provide means for adapting the system. These components are dedicated for example for collecting data and perform user modelling in the background [62]. The Open Monitoring Environment employs an agent architecture that belongs in this category. Data collection from a robotics learning environment can be automated by utilising agents to observe students' interactions within the environment (Figure 4.1). The basic idea is that agents do not process data by themselves, but they collect data and deliver the data to the teacher's learning process modelling environment and database for further observation. Two different levels of "intelligence" can nonetheless be defined within the agent population for collecting data from a programming task. At the basic level, an agent works as a data collector. For example, an agent can observe a button in the IPPE programming environment for robots and send

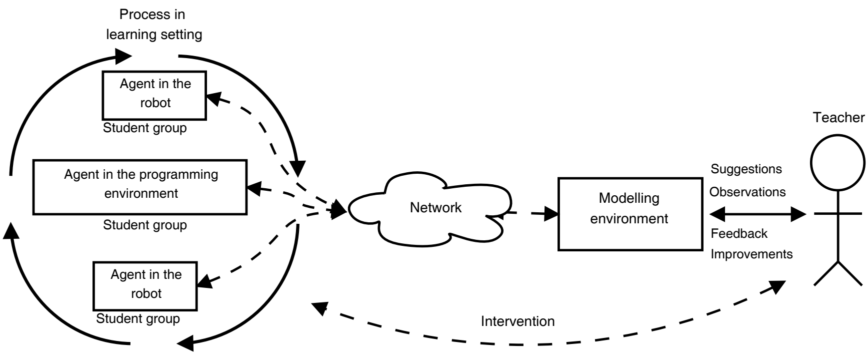


Figure 4.1: Overall agent architecture for the monitoring environment (modified from Paper II)

a message to the teacher's classroom model when students press that particular button. At the upper level, an agent can possess 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. This type of agent can send a message to the teacher's model when its internal rule identifies that a condition has become true, for example when a student's code contains the defined keywords.

4.2 CHALLENGE 2: HOW TO USE THE DATA EFFECTIVELY?

An educational robotics environment where student groups work independently with their projects can produce an overwhelming amount of data [Paper III]. Applications in the learning environment must provide support for manipulation and effective use of data originating from various sources in unpredictable order. The working processes in robotics classes are not static, and traditional Intelligent Tutoring System approaches and applications may be too static to support a rapidly changing and dynamic learning setting.

Much of the learning in science and engineering disciplines is related to understanding a set of rules that determine and regulate a certain phenomenon. 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.
  Example: If the robot has a touch sensor, it can detect collision to an obstacle in the environment (walls, furniture, etc).

- A robot programmer needs to understand the syntax and semantics rules of a programming language.
  Example: If the robot program has an event loop (a loop in which the sensors are monitored), the robot can read the sensors constantly.

- A teacher needs to understand the relationships between his or her learners within the working process.
  Example: If a student uses the computer and programming environment more frequently than other students, he or she might be dominating the programming process of that group.

In order to provide the teacher with better understanding for the observations rising from the learning process, a learning environment needs to expose data related to the rules. Also, the learning environment should enable data processing so that a teacher can define the rules and how to use them when reasoning about different aspects of a learning process in the robotics class. For example, the teacher may be interested in what types of learners there are or what has caused the current problems. As described in Section 3.3, many existing Intelligent Tutoring Systems often provide only a pre-defined set of predictions about the problems that the student might face. In most cases a teacher needs to have very advanced technical skills in order to make effective use of the support system or to modify it, if this is even possible. In fact, he or she needs the skills and patience of a technical developer, and this is not frequently available in today's classrooms. Also, the implementation tools of a learning environment should support development and use processes that differ from those of traditional learning environments or ITSs.

The need to open the rules and learning modelling processes in the learning environment to the end-user essentially drives to role blending between a software developer and a teacher. This means that the teacher needs to adopt software developer's or domain expert's task into his or her working processes. Traditional software development tools do not support this well. For example, a change to rules that define classification of students by their learning styles may require a challenging process of modifying the source code of the learning environment, compiling the code with the appropriate software development tools, and deploying the environment again in the classroom.

To overcome the technical and conceptual challenges connected with the limitations of traditional software development tools, we have used in this research Empirical Modelling to enable processes that are required in an open and highly modifiable learning environment.

4.3 EMPIRICAL MODELLING

Empirical Modelling (EM) [15] 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.

An EM artefact (a model) is a collection of *observables* that the modeller creates and manipulates in order to make the model reflect a real-world phenomenon. As the EM is about computational models, there are no restrictions on what an observable might be [52]. It can be an application window, a drawing canvas in the window, a graphical object (line, circle, square), string, or a list or a numerical value that does not have an explicit presentation that would be visible in the user interface.

Observables are not variables in a sense as variables appear in traditional programming languages. Maybe the closest counterpart to an observable is an instance of a class (an object) in object-oriented programming. However, the concept of the observable is more closely linked to the real world than the concepts of an object or a variable. Observables usually reflect a real-world phenomena in the EM model under construction, and they are interpreted according to what experience they offer to the modeller through interaction [5]. Objects or variables, istead, are used primarily to maintain programs' internal logic and structures.

Two or more observables in an EM model are connected together with *dependencies*. Dependencies are specific kinds of relationships between observables, and they are used to model the connections as real world's artefacts are connected together. An essential 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.

The Empirical Modelling environment consists of an interactive scripting environment, *tkeden*, which is used to create the models with several different notations. EDEN (Engine for DEfinitive Notations) is a core notation of the EM approach, and it has been extended with various notations, such as DoNaLD (Definitive Notation for Line Drawings), SCOUT (notation for SCreen layOUT) for describing the geometry and layout of windows, and EDDI (Eden Definition Database Interpreter) for definitive database operations. An essential feature of the EM environment is that all changes to the model can be applied without interrupting the execution of the model, hence without closing the EM environment.

Empirical Modelling adopts many features from the End-User Development (EUD) [44] software development approach. The core features of EM, such as a mechanism to apply changes without closing the environment, are also essential in EUD. By definition, there is only one role in EM which both a "developer" and a "user" (as understood in traditional programming approaches) belong to, and role blending is an inseparable feature of Empirical Modelling. The unified environment for developing and using the models fits the requirement of open and flexible learning modelling environment for robotics classes, where the end-user (a teacher in a robotics class) can affect the modelling of learning processes. Hence, Empirical Modelling was selected as an implementation platform for the monitoring environment. Next we will discuss more deeply some aspects of the EM through illustrative examples.

4.3.1 EM principles: Book on a table

By way of illustration, in a real world situation, if a book is placed on a table, the book's position in the room is obviously related to the position of the table in the room. Thus, if the table is moved, also the position of the book will be changed accordingly, without touching the book itself. In an EM model presenting the table and the book on it, the table and the book are *observables*. The relationship between the positions of the book and the table can be modelled with *dependencies* by using the EM keyword "`is`" as follows (for simplicity, I consider the positions here only in a 2D coordinate system):

```
book_x_position is table_x_position + (table_width / 2);
book_y_position is table_y_position + (table_length / 2);
```

These definitions relate the current values of the observables that define book's position, to values of the observables that define the table's position. In addition, the book's position is also relative to the table's size, being in the middle of the table. Now, keeping in mind that the EM environment maintains the dependencies automatically, we can see that when the position or the size of the table is changed in the model, the position of the book is also updated automatically to approximately the middle of the table according to the definitions above.

4.3.2 EM in a robotics class: Monitoring students' activity

To illustrate the use of the EM and the concepts of the observable and the dependency more deeply, we next present an example from the robotics classroom.

Let us consider a scenario in a robotics class of elementary level where students have been divided into several groups to solve a basic task in robotics (see Figure 3.1 in Section 3.1). The groups use the Lego Mindstorms robotics kit with the IPPE programming environment. The teacher of the classroom has devised the following task for the students:

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 with an obstacle.

This task can be divided into a number of sub-tasks, such as robot construction, programming, design, and testing. These can be divided further into smaller items, such as selecting the right components for the robot, or defining what program structures are needed (cf. examples of the rules in Section 4.1). Finally, primitive atomic actions, such as pressing the button in the programming environment to upload a code to the robot, can be identified.

Let us assume that the teacher wishes to monitor an activity level of a student group in the classroom. He or she needs first to define how the current activity level is calculated. The decision to observe activity level, or any other feature associated with the learning process, is based on the teacher's pedagogical preferences or intuition. In this example, the current waiting time could be measured as the time in seconds that has passed since the students' last interaction with the learning environment, such as pressing a button in the programming environment as discussed above. After defining the formula for the activity level, the teacher might want to visualise the data in an application by introducing a green bar, whose length corresponds to the current activity level. In an EM

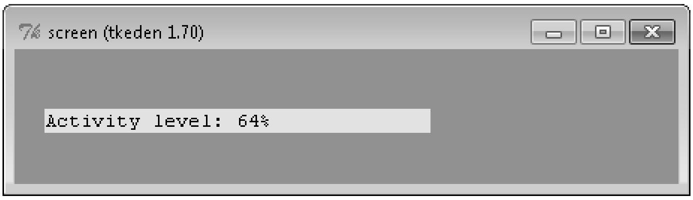


Figure 4.2: EM model for monitoring students' activity level

model presenting this visualisation, the activity level and different measurements defining the entity are observables. The relationship between the observables can be modelled with dependencies as follows in EDEN notation:

```
activity_level is (max_waiting_time - waiting_time);
```

This definition relates the current value of the observable that defines the activity level to the values of the observables that define maximum waiting time and current waiting time. When the `waiting_time` is changed in the model, the `activity_level` is also updated automatically as defined with the dependency. In an educational robotics environment, an agent observing students' activity could take care of updating the `waiting_time` observable. The agent observes the time that has passed since students' last activity with the programming environment. When, for example, the students create a piece of robot code in the programming environment, the observable will be reset to value 0.

The `activity_level` observable can be connected to the length of green bar with a dependency as follows in EDEN notation:

```
activity_bar_length is activity_level;
```

By using these definitions, a simple graphical interface can be built for visualising the data (Figure 4.2). In the visualisation, the length of the green bar and the text referring to the current level are updated automatically whenever the agent updates the observable `waiting_time`.

The simple model can be elaborated and refined gradually to provide richer presentations and alternative views for the data. All aspects of the current state of the model are expressed using the EM definitions, and the teacher can in principle manage all details of the model. It is, however, usually too much to expect that teachers without previous knowledge of programming could modify the EM definitions in a textual form. Even though the EM focuses on capturing the modeller's empirical observations through its "state-as-experienced" approach, the syntax of core EDEN notation (cf. examples above) follows the formal syntax of C programming language on which the EM environment is based [52]. To overcome the problems connected to working with formally defined expressions, a graphical user interface can be provided to make the model building process more accessible to the teacher. This is also in line with EUD technologies, where visual programming, including runtime interface tailoring, is one of the significant approaches [64].

The features of Empirical Modelling informed the core design principles and characteristics of how we developed the learning environment for unpredictable learning settings. The *Conflative Learning Environment* approach was designed and applied when implementing the *Open Monitoring Environment* for educational robotics classes.

4.4 CONFLATIVE LEARNING ENVIRONMENT (CLE)

A conflative learning environment (CLE) is an approach for building an open and flexible learning environments. The CLE approach was devised during the research process. The traditional division of the roles in educational technology development processes usually strictly separates the roles of developer, teacher, and learner from each other, in contrast to Empirical Modelling as discussed in Section 4.3. Moreover, the tasks undertaken by these process participants usually follow each other in a cycle with predefined steps. This process may be costly and time-consuming, and many systems whose design is reported in research literature do not get past the prototyping phase due to various reasons related to developmement and assesment processes [62]. Beynon and Roe [6] argue that constructionist computer-assisted learning approaches, such as EM, can be seen as unifying the roles of the student, the teacher, and the developer. By following this argument, the CLE use principles familiar from Empirical Modelling to take the users of the learning environments beyond their traditional roles of 'technology user' and blend the users' and developers' activities and working environments with each other.

The role blending can take place between the teacher and the developer, the student and the developer, or even between the student and the teacher, but the CLE approach does not define the level of the role conflation. The role blending takes place through cyclic processes where the users contribute to building the learning environment gradually by modelling the empirical observations arising from the current learning setting. Modelling is an essential part of the ongoing processes in the learning environment, and it can be done without interrupting the process in order to develop the features needed to enhance the learning environment, as would happen within a traditional educational technology development process using traditional tools.

The modelling process in the CLE approach (Figure 4.3) is based on individual data streams originating from the current learning process. The data is, for example, information about the users' activities within the learning environment, or automatically collected sensor data, or students' self-reflections about their progress. A learning environment based on the CLE approach should provide appropriate tools for the teachers to process this primitive data so as to obtain pedagogically meaningful collections that can be visualised. As an abstract approach, the CLE does not limit or define how the atomic data streams are combined in the learning environment or what kinds of visualisations the users build.

In a CLE, there is a distinction between development and modelling. When building a learning environment by following the CLE approach, there is, indeed, a need to have "traditional" software developers involved. They prepare the tools and the environment

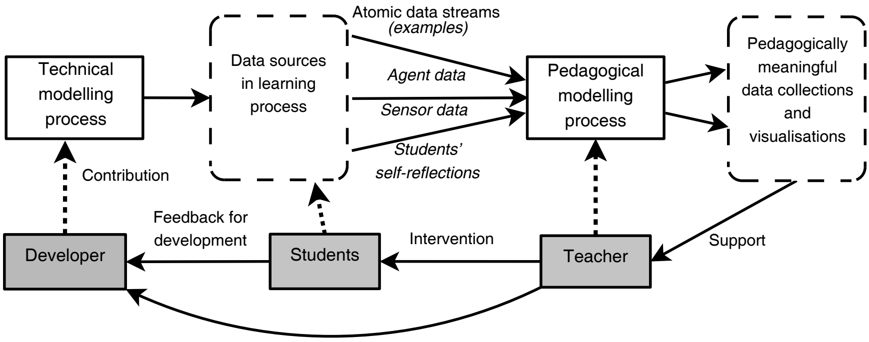


Figure 4.3: Example structure of an application based on the CLE (modified from Paper V)

so that the users can start their contribution (modelling), taking up the role of a developer. Modelling means that the users reflect upon their own surrounding and working environment through their observation of the learning environment.

We have proposed in Paper IV that to make the applications based on CLE approach more accessible for the users, 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 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 the environment by identifying data collections meaningful for the teaching context and visualisations for the data that the agents collect. It may be that the collections and visualisations of a given instance are not usable elsewhere, as they might depend on the phase at which the students are in their project.

4.5 OPEN MONITORING ENVIRONMENT (OME)

We have developed an *Open Monitoring Environment* (OME) by following the CLE approach. The core idea of the OME is to help the teacher to detect the right moments for intervention in a robotics class. In this way, the OME could potentially help the teacher to build his or her intervention strategies. In accordance with CLE approach, OME allows the teacher to combine data streams from the robotics classroom (Figure 3.4 in Section 3.2) with his or her own observations arising from the learning process. In this way, the environment encourages the teacher to model the learning process in the way that best suits his or her preferences and the current learning setting. We have implemented the OME by using the Empirical Modelling toolset, because it supports observing the cyclic processes that appear in the educational robotics learning environments as described earlier.

The OME produces different kinds of visualisations 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 [Paper III]. 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 with the EM observables. The progress, or lack thereof, can be expressed with colours or sizes of graphical elements representing the student groups.

The teacher constructs a set of observables by applying different rules to the available data based on his or her pedagogical expertise. The rules are needed to process and represent data in a meaningful way. For example, data about students' interactions with the learning environment is not necessarily pedagogically meaningful as raw data whereas the length of the program code, in contrast, indicates the students' progress, and there is no need to pre-process this information. The level of resolution and degree of pre-processing required in gathering data, however, cannot be predetermined. The OME allows the teacher to create new rules for classifying data according to the current learning setting, and the rule generation can also be automated with data mining tools [Paper VI]. The overall architecture and processes in the OME are presented in Figure 4.4.

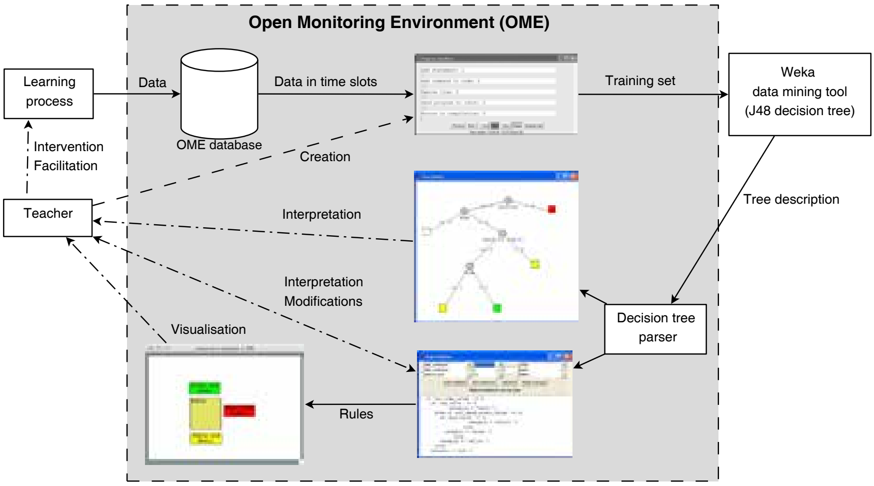


Figure 4.4: Components of the Open Monitoring Environment (modified from Paper VII)

In the current OME implementation, the students work in a robotics class by designing and building robots with Lego Mindstorms sets, and they program the robots by using the visual programming environment IPPE. Students' interactions with the robotics environment produce data that is recorded in the OME database. In the experimental version of the OME, data is generated from the four major functions of the programming environment. These actions essentially define the students' working process with the robots. The teacher can classify the events into one of four categories according to the anticipated progress (Table 4.1). During the classification process, the teacher creates a training set for the data mining algorithm. One classification produces one instance to the training set.

At any point of time, the teacher can launch the automatic rule generation based on current data in the OME database. The OME launches the Weka 3 data mining package [25] and passes the training set to the J48 decision tree algorithm. Weka outputs a description of a decision tree classifier, and the OME transforms the description to a tree visualization and associated rules that are exe-

Table 4.1: Classification of the training data

Class	Explanation
White	Students are not progressing, but they do not seem to have any particular problems (neutral situation)
Green	Students are progressing without any noticeable problems
Yellow	Students might be facing some problems (intervention may be required soon)
Red	Students have experienced problems that require intervention

cutable Empirical Modelling definitions. The rules automatically update the classroom map, and together with the tree visualisation the teacher can monitor and understand problems in the classroom and reasons behind the problems. A more detailed description of the data mining process can be found in Paper VI.

The OME has been built by following the CLE approach, and we can compare and contrast the student and learning modelling processes within the OME with the theoretical model-based and modelling-based extremes presented previously in Table 3.1. We can also use the same classification scheme to compare and contrast the OME with other learning environment. Table 4.2 shows how, for example, the OME environment adopts features from model-based and modelling-based approaches. Also Wayang Outpost [8] uses an empirical approach for data collection, but the learner model is based on theoretical predictions and a predefined model that reflects a more deductive modelling approach.

The OME, as prepared for the studies reported in this thesis, contains four distinct modules (*Progress Classifier*, *Classroom Viewer*, *Tree Visualisation*, and *Rule Builder*) that allow the teachers to monitor the students' activities and explore the learning processes with simple graphical views (Figure 4.5). All views are updated in real

Table 4.2: Comparison between the example systems

	Wayang Outpost	**Open Monitoring Environment**
Modelling approach	Partially empirical, based on theoretical predictions, deductive	Empirical, abductive
Learning model	Partially given	Constructed
Adaptation	Black box	Transparent
Roles in the learning community	Separated	Partially blended (teacher - developer)
Working environments	Separated	Conflated between teacher and developer
Direction of modelling	Top-down	Bottom-up
Modifications to the tools	Through the software development process	Partially on demand

time and are automatically based on the current data, rules, and observables. The user interface and functionality of the OME prepared for the experiments was very simplified. The decision to provide a simple interface was informed by our initial knowledge about the teachers' knowledge of computer-supported learning environments, and of educational robotics in particular.

The **Progress Monitor** module supplies the teacher with a graphical view of the number of student interaction events in a specific time window. The module allows the teacher to classify the events into one of the four categories listed in Table 4.1. This is a key element of the data mining process, and relies on teachers' pedagogical expertise and face-to-face experiences with the students in the classroom. The **Classroom Viewer** module presents the current learning progress model as a two-dimensional map. In this imple-

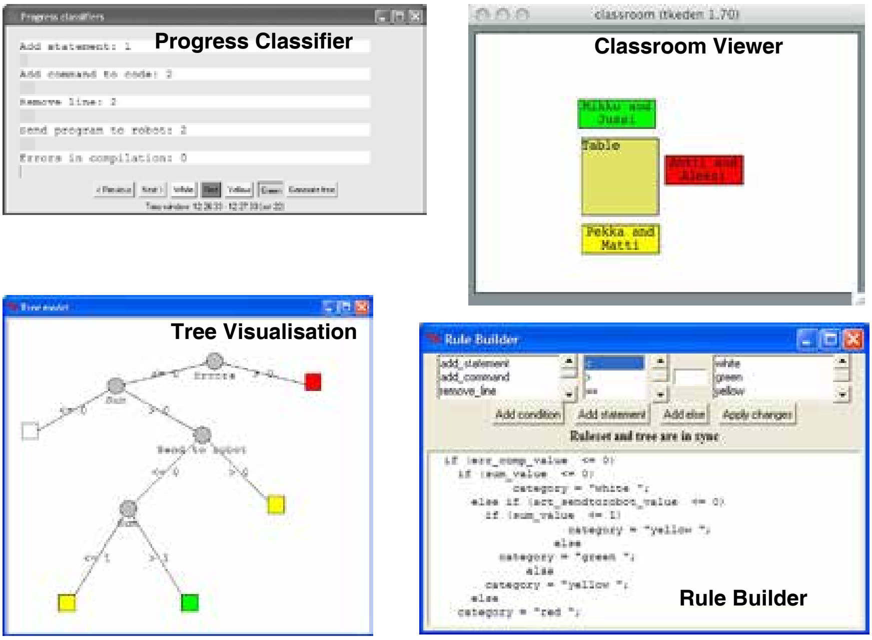


Figure 4.5: Modules of the Open Monitoring Environment

mentation, each student group is visualised as a rectangle with the names of the group members in it. The fill colour of a marker is updated automatically according to the current rule set.

The **Tree Visualisation** module shows the current learning progress classifier (expressed as a pre-ordered binary tree), which is essentially a combination of teacher's pedagogical expertise (training set creation), and a well-known machine learning scheme (the J48 decision tree algorithm). It is typical that a teacher creates several classifier models during a robotics class. The first models are usually not very expressive, and they describe rather general and even obvious patterns of actions. However, during our experiments we noticed that, before long, the classifier is capable of taking into account the aspects in the learning processes that otherwise would remain hidden (see more analysis about this in Chapter 5).

The rules presented in the tree visualisation are available as Empirical Modelling definitions in EDEN notation in the **Rule Builder** module. The teacher can modify the rules in the module either by using the graphical user interface provided or editing the rules directly in the text field. The modifications to the rules are effective as soon as the changes are applied. This results in the automatic update of the visualisations in the Classroom Viewer module.

During the research process, we have developed several versions of the OME. Different visualisations approaches have been tested in the case studies, and all these versions have contributed gradually to the latest OME version presented above. The appropriate parts of the earlier OME versions are presented in conjunction with the corresponding experiments in Chapter 5.

The technical environment of the OME contains laptop or desktop computers, a local area network, and Lego Mindstorms robotics sets. Each student pair has their own computer to work with. The teacher also has his or her own computer, which the teacher uses for observing the students' activities through the OME.

Communication between the data gathering agents and the teachers' modelling enviornment takes place over the network. The communication can exploit any network protocol, but, in the experiments so far, we have used shared network folders as the communication channel. The shared network resources can be located in a dedicated server or in the teacher's computer, as was the case in most of the experiments. The details of the communication over the network are taken care of by the operating systems of the computers. In this way, the communication between agents and the OME is transparent. This also allows easy use of diverse computing facilities with mixed operating systems.

5 Analysis

The research presented in this thesis follows a development research framework [58] where existing theories and results from the continuous evaluations are used as a basis for a concept or an application development. The case studies described in this chapter form a continuous series of experiments and results from individual studies are used as an input for the further development and the next case study. The original research questions (Section 2.1) were answered by utilising the results from the case studies.

The case studies described in this chapter were conducted in real educational robotics settings. In all cases, the students were working in a robotics project with Lego Mindstorms Robotics Invention System sets and the IPPE programming environment. Students' interactions with the programming environment were saved for further analysis. In some cases, additional data collecting methods, such as video recording and field notes, were used. The original research questions (Section 2.1) were answered and two major contributions of this research were developed through the iterative process of software development and modelling, data collection, and analysis as described in the following sections.

5.1 THE MAIN RESEARCH CONTRIBUTIONS AND RESEARCH THEMES

Two main contributions can be identified from this research. First, we devised the **Conflative Learning Environment** approach (described in the previous Section 4.4). The CLE approach adopts its key features, such as role blending, rapid modelling cycles, and live development, from the Empirical Modelling and the end-user development practices. Second, the CLE approach was used to guide the development of **Open Monitoring Environment** for observing and exploring students' learning processes in educational robotics classes. The OME features automatic data collection from a robotics class and open data mining to support teacher's comprehension about the students' progress.

Next in this chapter we present the case studies whose results guided the research process. It is important to notice, however, that an important contribution to the research was made in between the case studies in the form of practical development of the OME. The two major contributions of this research (the CLE approach and the OME) can be divided further into four major themes as follows. These themes are mainly based on research questions Q3 and Q4. The themes can be interpreted also as identifiable phases of the development cycles in the research process. The themes and, as well as research questions, arise from the problems that we face during the research process when working in educational robotics settings. Research questions Q1 and Q2 focus more on literature studies and analysis of background information, as has been indicated earlier in Chapter 2.3. Research questions Q1 and Q2 are answered throughout the whole research process and it is more difficult to identify specific research interventions for them than for questions Q3 and Q4.

- Theme 1: Agent architecture for collecting data from educational robotics settings
- Theme 2: Empirical Modelling as an implementation tool in the CLE
- Theme 3: Teachers using the OME
- Theme 4: Data mining in the OME

5.2 THEME 1: AGENT ARCHITECTURE FOR COLLECTING DATA FROM EDUCATIONAL ROBOTICS SETTINGS

The first case study [Paper III] was conducted to see how the agent architecture proposed in Paper I performs in a real educational robotics setting. The research problem in this case was to study whether the proposed agent-based approach is suitable for data collection. The IPPE programming environment was enhanced with software agents as described in Chapter 4. Each agent in the learning environment was defined to send a message when a specified time in the use of the particular button of the IPPE programming environment was exceeded. The delay was specific to each agent in a student group and all groups had similar agents observing the activities.

The initial delay for the agents was set between 30 seconds and 2 minutes depending on the element that the agent was observing. The initial values were estimates for suitable values in a typical robotics class. The delays were also adjusted during the session based on the teacher's judgement about the students' current progress and the working context. The data was stored in files and analysed afterwards with standard UNIX command line programs.

Data collected in this case study was analysed in two phases. First, the data shows that the student groups worked with the robotics project for 37.3 minutes on average. Analysis of logged data show that during this time, each agent instance sent on average 34 messages / minute [Paper III]. This was the first clear indication that the amount of data originating from the robotics classes can be too large to be handled manually, and there is a need to cluster and filter data according to the current activities in the classroom.

Besides messages sent by the agents, students' interactions with the programming environment were captured. The number of messages of this type indicates the frequency of the use of the particular buttons in the programming environment. The analysis of this information showed us that it is possible to expose patterns of use and students' typical work flows with the proposed agent architecture [Paper III]. These results were used to guide the following research step consisting of design and implementation of the teacher's working environment for the OME with the Empirical Modelling tools.

5.3 THEME 2: EMPIRICAL MODELLING AS AN IMPLEMENTATION TOOL IN THE CLE

The aim of the Conflative Learning Environment approach is that learning environments built upon it also provide contextualised support for unexpected requirements in different learning contexts. Hence, it is not enough just to collect data for example from a robotics environment and show it to the teacher as a stream of raw data or distinct events. The second theme in this research process focuses to study how Empirical Modelling (EM) could be used as an implementation platform when following the CLE approach. Various aspects of this theme were studied throughout the research process from different viewpoints. Results of these studies are presented especially in Paper II, Paper III, Paper IV, and Paper VII. The features of the EM environment and their connections to the key concepts of the CLE have been discussed earlier in Chapter 4. We next present the main findings on this theme from the case studies as reported in the original articles.

The initial Empirical Modelling study reported in Paper III, explores the possibilities that the Empirical Modelling tools allow for building the teacher's controlling environment, and further, the benefits that the use of EM brings in comparison with the traditional software development tools.

The case study was carried out with two groups of primary and secondary school children, aged between 10 - 15 years. The children were participating in the Kids' Club of the University of Joensuu. 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.

During the robotics class we built a simple model that shows when the students had completed the assigned robotics task. The model building was an incremental process, and the model was realised while the students were working on their robotics projects. The model building was supported through questioning the students about how they had 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, and further that they would also need an event loop in order to observe the touch sensor constantly.

As a result of this study, an EM model presented in Figure 5.1 was developed. 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, namely existence of code observing the use of the touch sensor and the event loop. In Figure 5.1, "Group A" has got these essential elements ready, whereas "Group B" does not have either of these two elements included in their code. The final result contains 30 lines of the EM definitions (see Paper III for details). The modelling process indicated that the EM tools are usable when applying the CLE approach in learning environment development.

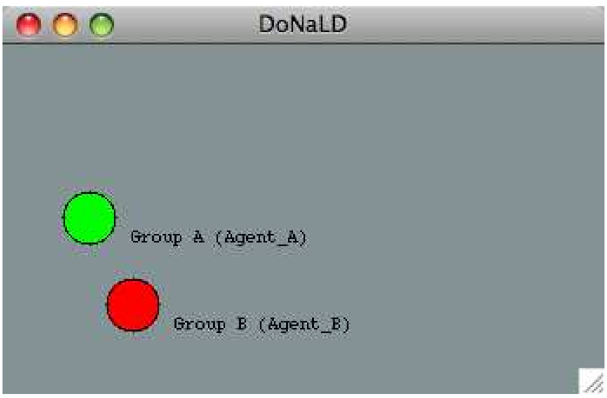


Figure 5.1: The first EM model for monitoring students' progress [Paper III]

The simple OME model presented in Paper III was redefined during the research process into different forms and more comprehensive models. The role of Empirical Modelling in the CLE approach was discussed, for example, in Paper II where we explored various aspects that the EM could bring to the development process. We concluded that not all teachers would be satisfied with a simple classification of student status into three colours, and the teacher can in principle change all aspects of the visualisation while the model is running and without restarting the environment. Potentially, the teacher could extend the model by adding new visual elements to it. Figure 5.2 illustrates some additional visualisation features of the OME that can be used to provide for the teacher a more comprehensive view over the students' progress.

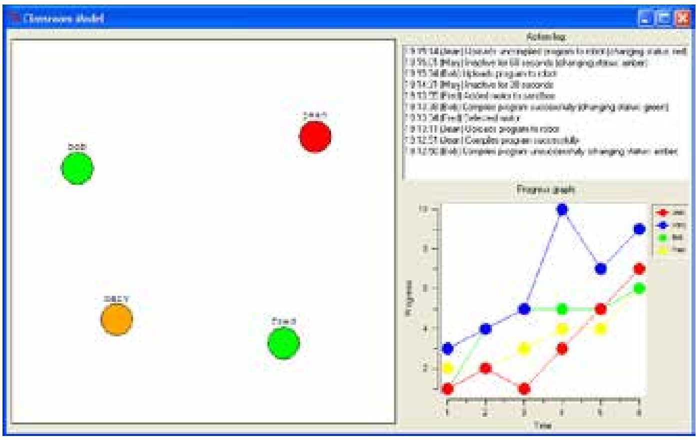


Figure 5.2: An example of a more comprehensive EM model [Paper III]

In Paper IV we report the outcome of the next iteration of the cyclic model-building process for the OME. In this version, we added alternative views for observing data, as well as functionality to simulate students' progress subsequently (Figure 5.3).

We concluded in Paper IV that there are technical challenges when using the EM tools to construct a learning environment. We proposed that to make the learning environment based on the CLE approach and Empirical Modelling more accessible for the teacher, the modelling of the learning process should be divided into two

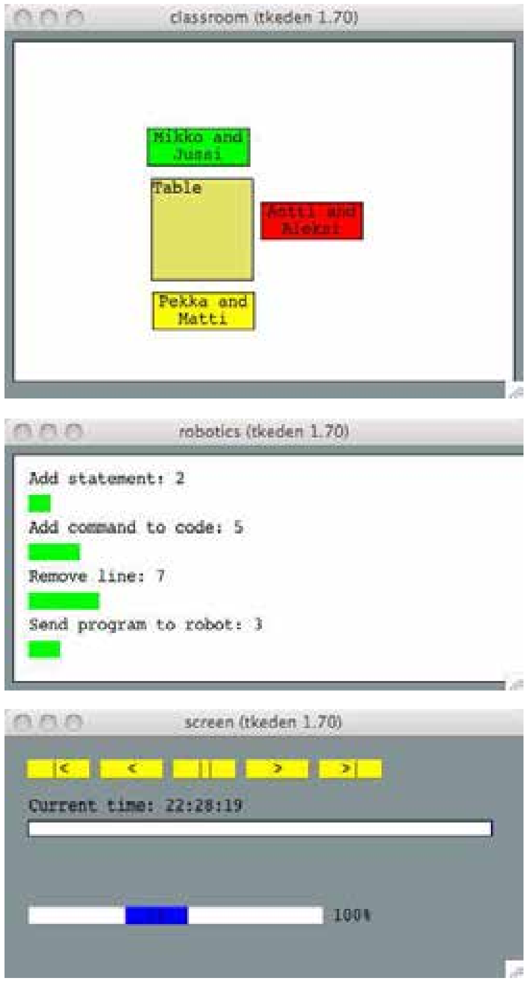


Figure 5.3: An enhanced OME model [Paper IV]

part as described earlier in Section 4.4.

5.4 THEME 3: TEACHERS USING OME

The main outcomes of the following experiment are reported in Paper V. In this experiment, the research focused on the following question closely connected to research question Q4: How does the OME help the teachers who are working in a robotics classroom?

The experiment took place in a secondary school near the city of Johannesburg, South Africa. The boarding school had 40 children from Grade 6 to Grade 9 registered as students. The school students were mostly abandoned children from neighbouring communities, and they were living in the school area.

Four teachers and 12 students participated in the study. The participants were divided into two groups of two teachers and six students. During the study, the students worked in pairs for a given robotics task. Neither teachers nor students had prior familiarity with robotics. The student groups were provided with pre-constructed wheeled robots, and the following task was given to them:

Program the robot so that it runs forward for five seconds. Then, continue the program so that the robot runs backward to the starting position.

The teachers' work during the experiment can be roughly divided into three parts. First, they observed students' actions through the OME environment. When the agents delivered new data to the modelling environment, it was automatically updated to reflect the current situation. Secondly, the teachers used this output to determine when it was appropriate to intervene in students' work (e.g. when there appeared to be a problem). Thirdly, the teachers modified the rules that determined how the agent data was reported in their environment.

Data collection in the study was conducted in three ways. First, teaching sessions were recorded with a video camera. Second, data produced by the agents based on the students' actions was saved to the students' workstations besides delivering the data to teachers' modelling environment. Also, teachers' actions with the OME were saved to log files. Third, the focus group interview was conducted and recorded with a video camera.

All data was collected anonymously. The OME automatically coded data produced by the agents with anonymous identifiers. Videotapes were transcribed and anonymous identifiers were used also for the persons appearing in the video. All data was collected from the computers and videotapes to one location. Before the ac-

Table 5.1: Codebook classification

Code	Category
C1	Background information
C2	Experiences in using the monitoring environment
C3	Approaches for following the students
C4	Improvements for the monitoring environment
C5	Other application areas for the monitoring environment
C6	Robotics in the classroom
C7	Use of technology in the classroom

tual analysis, all data was saved to a DVD disc for a backup. After verifying the backup, the original data from the students' and teachers' computers and videotapes was erased.

The analysis started by transcribing the video and interview material. During the transcription, a codebook (Table 5.1) was built to help the analysis and make the analysis consistent by following a method of open coding [43]. However, before the actual analysis, the reliability of the codebook was tested.

In order to test the reliability of the codebook, two researchers classified transcribed research material independently according to the codebook categories. The researcher classified the material completely, whereas another researcher classified about 30% of the material. To get a representative sample over the whole material, the transcription was first divided into four parts, and a random sample of 30% was selected from each part. Based on the classifications, the inter-rater agreement was calculated by using Cohen's Kappa. The result ($\kappa = .75, p < 0.001$) indicates that there was substantial agreement about the codebook classification between the two raters, based on the interpretation of Cohen's κ-measures given by Landis and Koch [41]. Furthermore, it is very unlikely that this level of agreement was achieved by chance. Based on the substantial inter-rater agreement, the codebook was used to support the interpretation of the qualitative research material.

The focus of the analysis was to identify the opportunities that the OME brings for teachers' work, and to identify the role that EM principles and tools play in implementing an environment based on the CLE approach. Furthermore, the analysis was open for unexpected patterns and ideas that would guide further development. The new ideas and teachers' opinions about the additional information that the environment should provide were appreciated as potentially valuable for improving the OME further. This process is an essential part of the CLE approach. The outcome of the research process served as an input for the next iteration of technical modelling, and it also helped us to identify ways in which the teacher can be enabled to adopt the role of developer in accordance with CLE principles. Transcribed video and interview material was analysed by categorising material according to the codebook with the ATLAS.ti software.

One of the main results of our study was showing that the OME is useful in the classroom especially with teachers who do not have prior knowledge about the educational robotics environments. The OME gave the teachers additional data on which to rely when making decisions regarding intervention, and they were able to use the information provided by the OME to support the decisions.

> *"It's helpful because you don't have to rely on students to tell they are having a problem or they are stuck, because I know that we are having issues with some of students that they don't ask questions when they don't know what they are doing."*

The teachers were able to recognise students' particular problems through the OME system better than when observing the students without such support. This was especially evident in situations where one of the teachers was following the students' progress through the OME system, whereas another teacher was observing the student groups by herself. In these cases, the teacher in the classroom did not notice that the students were having problems, but another teacher was able to see this through the OME and help the group with the problems. Most of these cases were related to technical problems when the students were unable to upload code to the robot due to the unreliability of the infrared link between the computer and the robot. It was typical for these cases that the students did not ask for help but repeatedly tried to upload the code to the robot, despite the error messages generated by the programming environment.

The analysis of the research material (questionnaires, video material, interview) showed that, in general, the teachers benefit from a tool like the OME that helps them to follow the dynamics of groups of learners. This becomes especially important if the teaching setting and tools are new to the teachers. Furthermore, if the number of students grows, the need for this kind of tool becomes more obvious.

> *"It's always useful to understand the dynamics of the group"*

> *"It's useful in the teaching environment to know what is going on in the group. Anything that helps understand that better would probably help to manage in the class."*

5.5 THEME 4: BACK TO THE RULES: USING DATA MINING IN THE OME

The results presented above indicate the need for data mining tools to automate and initiate the rule data processing and rule generation. Data collected from the experiment described in the previous section and in Paper V was used to build a data mining functionality to enable semi-automatic rule generation in the OME [Paper VI, Paper VII]. The OME version used in these case studies is the same as that presented in Figure 4.4. The first research question asked during the case studies related to the data mining features and research question Q3 was [Paper VI]: What are suitable data mining techniques for the Open Monitoring Environment and how should they be implemented in the OME so that they support abductive reasoning?

The development of data mining features was started by identifying suitable data mining techniques by consulting data obtained from the previous study. Captured data was preprocessed into a suitable training set for the Weka 3 data mining environment [25] as described in Paper VI. The training set was used to experiment with several algorithms with Weka, including different decision trees, decision tables, Bayesian networks, and multi-layer perceptrons. To measure the accuracy of the tested algorithms, we used the 10-fold cross-validation method. In this method, 10% of the data is used as a test set and the remaining 90% is used as the training set. This process is repeated 10 times, each time with a disjoint test set. Finally, the accuracy reported is the average accuracy across these 10 trials [68].

The J48 decision tree algorithm, best-first decision tree (BFTree) algorithm, and multi-layer perceptrons (MLP) algorithm outperformed other algorithms. Table 5.2 shows the cross-validation results for the different algorithms. In the light of the weighted average for the accuracy (true positives), J48 performs best (87.1%). Besides accuracy, we used the *Receiver operating characteristic* (ROC) area measurement for validating the classifiers. According to [53], the use of the area under the ROC curve in measuring classifier performance should be preferred over accuracy. There were no statistically significant differences ($p > 0.05$) between the accuracies or ROC areas of the different algorithms. On this basis, any of these three algorithms could be used in the OME.

Decision trees are simple and interpretable; they are also easier to implement in the OME than for example neural networks, which are very hard to trace. Thus, J48 was selected as a starting point for building the data mining features for the OME. J48 is an open-source implementation of the popular C4.5 decision tree algorithm [55].

Based on these results, we designed an architecture and developed and tested a proof of concept implementation for a data

Table 5.2: Results of cross-validation

	J48		**BFTree**		**MLP**	
	Accuracy	*ROC*	*Accuracy*	*ROC*	*Accuracy*	*ROC*
Average	87.1%	0.887	82.3%	0.899	82.3%	0.890
White	100.0%	0.921	100.0%	0.921	100.0%	0.897
Green	82.6%	0.877	82.6%	0.894	82.6%	0.908
Yellow	50.0%	0.718	37.5%	0.763	50.0%	0.733
Red	100.0%	0.971	80.0%	0.970	70.0%	0.963

mining module for the OME as described in depth in Paper VI. An initial experiment was conducted with a primary school teacher who did a retrospective analysis for data collected during the South African case study with the data mining module of the OME. The analysis was supported with video material collected during the experiment. The aims of the experiment were a) to test a technical feasibility of the data mining implementation, and b) to get an initial understanding about what kind of classifiers a professional teacher is able to create with a relatively small set of data, and how he or she interprets the results.

By following the workflow described in Section 4.5, the teacher created the first decision tree after classifying a data set of five minutes (29 events). By iterating the data mining process, the teacher created a well-working classifier after 22 minutes (92 events). The teacher's overall interpretation for the classifiers (Figure 5.4) was that they indicate that the students may have had problems on compiling and sending their programs to the robots. Depending on the classroom setting and number of students, teachers often experience difficulties on tracking and especially comprehending reasons behind this type of problems without having additional tools, such as the OME. Based on the videotapes and field notes collected during the case study, this, indeed, was the case during the particular robotics class. The result indicates that it is possible to create useful and interpretable classifiers with a relatively small amount of data.

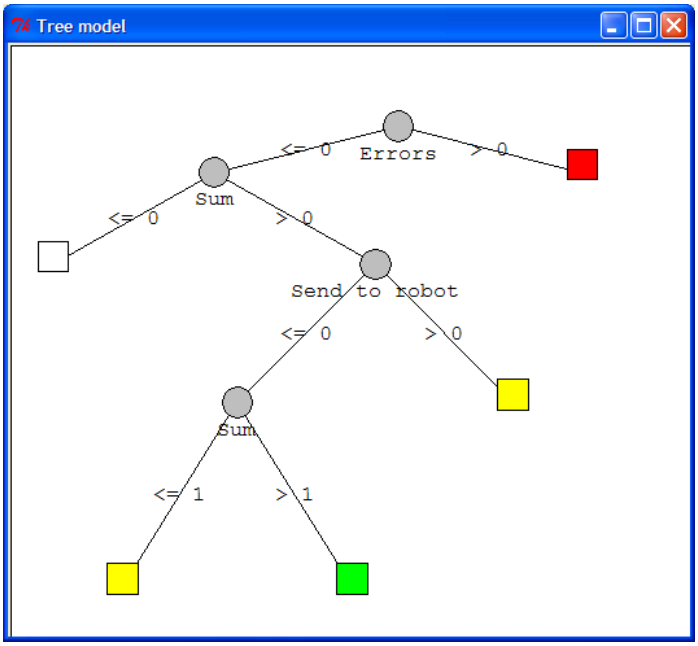


Figure 5.4: The final decision tree generated based on teacher's classifications during the initial data mining experiment

The final data mining case study described in Paper VII took place at the Kids' Club of University of Eastern Finland. 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 three 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 research question for this case was [Paper VII]: How can the OME support facilitation in a robotics class by providing the instructors learning process classifiers that are based on real-time data?

In the experiment, by following the workflow described in Section 4.5, the teachers created during the session 13 different versions of decision tree classifiers. Each decision tree version was saved to a computer's hard disk for further analysis. The collected research material (decision tree classifiers and supplementary material, such as field notes) was analysed qualitatively. The saved decision tree models (n = 13) were analysed after the teaching sessions. The focus of the analysis was to see how the decision tree models evolved during the teaching session when the amount of data in the OME database was growing gradually while the students worked in the robotics environment. Furthermore, the instructors' working process was analysed in order to see how they used the OME and data mining visualisations to support the facilitation in the learning process. The analysis was supported with extensive field notes collected during the teaching sessions.

The results from the study were two-fold. At first, we identified a situation where the OME failed to produce an expressive classifier, in contradiction to our previous findings in Paper VI with the retrospective analysis. In this respect, the OME did not provide the needed support for the instructors' intervention strategies. An average size of a data set in a time window in this study was 20% smaller than that presented in Paper VI. This may have had a negative effect for the functionality the J48 decision tree algorithm. On the other hand, data mining process in the OME is very sensitive to the nature of data collected from the learning process, as well as to users personal preferences and pedagogical expertise. After not being satisfied with the output of the OME, the instructors were able to adapt the environment to meet the context-dependent requirements, and adjust the OME eventually to fulfil the monitoring needs, with the ultimate view of understanding the students' learning difficulties.

The analysis of data mining features and the robotics class instructors' working processes show us that the instructors are able to use the key EM features to modify the learning environment to match to the current context as proposed in the CLE approach. This activity goes beyond normal teacher activities in a classroom, and role blending between a teacher and a software developer in a learning environment provides a novel way for building a personalized and contextualized support environment for a robotics class.

6 Discussion

Teachers working in explanation-oriented learning environments often need support for facilitating students. Finding the explanations behind the current phenomena in the classroom, a process of abduction, can foster creativity in the learning and teaching processes [59], but so far this has been rarely exploited in the learning environments.

Intelligent Tutoring Systems often use data mining to build models for learners and their progress. However, model building and implementation is usually a complex and time consuming process, and it involves domain experts to prepare data and train the models, and software developers to implement the models within the learning environment. Amershi and Conati [1] give an example of an automated data mining solution to support exploratory learning activities. They, however, conclude that their approach requires a substantial amount of data to work.

The Open Monitoring Environment aims to encourage the teachers to think creatively and explore the reasons behind the current situation in the classroom by classifying the patterns of events and visualising the predicted progress based on a semi-automatic and incremental data mining process. One of the main findings from our research is that, by opening the data mining process to the teacher, it is possible to implement the required support for finding explanations and exploring learning processes even with relatively small data sets.

It is evident that the OME and similar monitoring environments are beneficial especially for the teachers who are novices in working with a robotics environment. Use of technology in teaching was not an issue for teachers who participated in our study in South Africa [Paper V], because all of them were familiar, for example, with the computers, and they had used them in teaching. However, neither the teachers nor the students were familiar with the robotics and the possibilities that the robotics environment affords. Despite the lack of experience in working in the robotics environment, or perhaps just because of it, the teachers gained benefit from using the OME and it helped them to see the problems that they would not necessarily have noticed.

End-user development, with its various aspects, provides tools for tailoring learning environments to suit different contexts. However, new tasks and processes that the development activities and role blending bring to the teachers' work may rapidly turn to be too overwhelming or difficult. For example, the requirements for the skills that are needed to modify the Empirical Modelling definitions as presented in the examples of this thesis easily go beyond a regular teacher's level of skills. This is because EM indeed requires a certain amount of programming knowledge and skills for formal thinking, even though working with the EM tools should not be considered as programming in traditional sense.

It would be possible to provide sophisticated user interfaces with the Empirical Modelling or other implementation platform, but providing too finished solutions limit the teachers' possibilities to effect "under-the-hood" functionality of a learning environment. One way to balance between these two extremes is that proposed by the OME, where the user interfaces are essentially fixed, and hence not subject to pedagogical modelling. The low-level access to selected core functionalities of the environment, such as interaction with the data mining process and resulting classifiers, blends the roles of a developer and a teacher and activities associated with them. The role blending, however, does not interfere with the teaching but on the contrary, it naturally supports the decision making in the classroom.

When working in unconventional ways in the software development domain, there is also a need to develop an alternative metrics for evaluating the outcome. Data mining applications are usually evaluated by their statistical accuracy and precision ratio, but other factors such as interoperability and pedagogically meaningful constructions of the classifiers are even more important aspects in the

OME than the traditional data mining measurements. When working with the OME and data mining environment, the teacher incrementally makes a classifier that resembles his or her ways of comprehending students' progress. This evidently makes the classifier perform better in the particular learning setting, but not necessarily in other similar settings or with other teachers whose the pedagogical preferences and teaching style are different.

To make role blending a solid part of the learning environment construction process, as proposed by the CLE approach, the tools should also support such a working paradigm. We have used Empirical Modelling with the current OME implementation. It is of course clear that a monitoring environment with features similar to those of the current OME could be constructed using paradigms other than Empirical Modelling. However, there are fundamental differences between EM tools and traditional programming approaches that need to be addressed in this discussion.

It is important to notice that EM is a holistic approach for building computer-based artefacts, and it cannot simply be compared either to any particular phase of programming, or to an entire programming process [26]. This arises from the fact that "traditional" programming approaches are focused on producing an output based on the specifications, whereas EM is primarily about developing understanding of a situation in the current world (e.g. a robotics classroom, in the context of this thesis). Based on this, I argue that EM suits in the CLE approach better than traditional development tools, such as Java. The working processes in the CLE approach require a flexible and easily accessible development environment, and EM has these features built-in allowing needed role blending in the development. It is clear that the development and teaching processes cannot be blended completely. There is always a need to prepare the environment for each new context in advance by making a distinction between technical and pedagogical modelling.

6.1 APPLYING THE OME IN OTHER CONTEXTS

We have presented the CLE approach and the OME with a strong focus on the educational robotics context. However, the CLE approach and the applications built based on it, such as OME, can also be rather easily applied in a completely different domain. 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 [15] provides a catalogue of pre-existing models which can be modified to suit to 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 of the CLE approach.

While modelling the early prototypes of the OME, we noticed that the data collection methods and learning process reconstruction tools of the OME are especially well-suited for deployment in other application areas. While building our robotics application, we applied the replaying module in an educational game for a research project studying technologies for HIV/AIDS education [2]. 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.

The latest development on using the OME and ideas behind it is described in [28], where we concluded that one of the major challenges in tablet-based learning environments is in providing sufficient support for teachers. Furthermore, we explored what possibilities the OME could provide in such environments. This development is well aligned with our work on bringing open data mining features to a commercial Geddit learning environment [21]. Interactions in Geddit aim to help the teacher to understand how well students are learning through a color-coded view based on students' responses about how confident they are with what they are being taught [65].

6.2 LIMITATIONS OF THE STUDY

This research investigated the Conflative Learning Environment approach and the Open Monitoring Environment mainly in the context of educational robotics classes. Hence, generalisation of the results to other contexts cannot be made without further investigations. Some of these efforts are described in Section 6.1. The strong emphasis on technical development and experimental nature of the implemented environment led us to conduct case studies with rather small groups of teachers and students. The technical analysis of the environment played an important role in the case studies, and this, together with small target groups, did not leave room for a thorough analysis of the teachers' working processes and the effect of the OME in teaching. This makes generalisation of the results difficult. The OME works well in the robotics settings similar to those of the case studies, but the impact of different kind of data sources need to be addressed in the further studies.

The problem of generalisation is also partially connected to the technical limitations of the Empirical Modelling environment. EM is no doubt very well aligned with the CLE approach. Even though EM tries to capture informal real-life observations and user's empirical observations, it obliges the user to use a formally defined syntax similar to procedural C language. Though EM proved very efficient in implementing simple models (cf. the early OME version presented in Section 5.3), the complex models such as the final OME version presented in Section 4.5, are rather difficult to maintain and enhance due to reasons discussed more deeply in [52].

7 Conclusions

In this thesis, we have developed an Open Monitoring Environment (OME) to help teacher's intervention in educational robotics classes. In parallel with the development of the OME, a novel Conflative Learning Environment (CLE) approach for constructing learning environments was defined. The research problem was divided into four research questions. This work answers to these questions as follows.

7.1 RESEARCH ANSWERS

Q1: *What are teachers' expectations for modelling learning progress in educational robotics environments?*

This question was answered by analysing literature and deriving conclusions based on the lessons that we have learnt during almost 10 years of running educational robotics activities in various settings. Cyclic learning processes, group-oriented working methods, and students' unpredictable paths for problem solving often lead teachers to difficulties in following the activities in the classroom. These challenges and other expectations in educational robotics settings are analysed in Chapter 3.

Q2: *How is it possible to conceptualise the modelling process?*

Existing approaches and applications for following the students emphasise mostly theory-based approaches for building the learning model, and these tools are not usually accessible for the modifications by the teacher. To answer this question, we have defined an alternative approach for building the learning environments. The Conflative Learning Environment (CLE) approach takes the users of the learning environments beyond their traditional roles of 'technology user' and blends the users' activities and working environments with each other. The role blending takes place through cyclic processes where the users contribute to building the learning environment that is developed gradually by modelling the empirical observations arising from the current learning setting. The modelling is an essential part of the ongoing processes in the learning environment, and it can be done without interrupting the process in order to develop the features needed to enhance the learning environment, in contrast to traditional educational technology development process. The CLE approach is discussed in Section 4.4.

Q3: *What are technical requirements for implementing a learning environment based on the CLE approach?*

This question was answered by gradually modelling the activities in the robotics environment with Empirical Modelling tools and experimenting with the models in educational robotics settings. The Open Monitoring Environment consists of various modules and views for observing the students in the robotics classroom based on the rules that are derived from data collected from the background of the students' learning processes. Besides modelling the monitoring environment with the EM tools, the development featured an implementation of an agent architecture to collect students' interactions with the IPPE programming environment and Lego robots. Furthermore, open data mining features were developed for the OME for semi-automatically creating the rules. Feasibility of the developed technical solutions was tested in various robotics settings. The results showed that a) the agents are functional in collecting data, b) the EM tools can be used to build an environment such as OME, and c) data mining methods, and decision trees in particular, are efficient on classifying events and exposing unexpected patterns of actions arising from educational robotics classes. We answer this question by introducing the OME and its features in Section 4.5.

Q4: *How does the OME support teachers' work in unpredictable robotics settings?*

This question was answered by conducting several case studies in real educational robotics settings in Finland and South Africa. The results show that the OME provides relevant additional support especially for teachers who are novices in working in the robotics environments. Furthermore, the Empirical Modelling was proved to be an effective approach for implementing an environment such as the OME. Data mining techniques can be used effectively to implement a mechanism to classify the events occurring in the educational robotics environment. The result of the classification process can be used to support the teacher when he or she explores the explanations behind phenomena in the classroom. The results of the analysis are presented in Chapter 5.

7.2 FURTHER QUESTIONS

We have shown in this research that it is possible to build a supporting environment for the teachers so that the environment encourages role blending between a teacher and a software developer. In the current implementation of the OME, the teacher builds learning progress models by utilising his or her expertise, first-hand experiences from the current learning setting, and intuition. The students, for their part, use the standard environment (the IPPE programming environment and Lego robotics) without any explicit intervention for the learning modelling process. A relevant question for further studies would be how to actively involve the students in the modelling process, and how to make them contribute to the models by blending the roles of a student and a developer, or of a student and a teacher.

The OME versions developed for this research were highly experimental, and deployment in a larger classroom for continuous use may require a considerable effort for rebuilding especially in respect of the teacher's monitoring environment in Empirical Modelling. The Empirical Modelling environment may not be very accessible for regular teachers, and recovery mechanisms for unexpected crashes of the EM environment need to be considered especially carefully in order to avoid unnecessary interruptions and data loss in the modelling process. A bigger question connected to this is whether the EM would be an ideal tool for implementing the OME in larger scale or whether the experimental nature of the EM tools prevents the teachers form utilising the environment efficiently. On the other hand, the unusually open and explorative EM environment supports the key concepts of the CLE approach better than any other development or modelling environment we are aware of. The latest development with the web-enabled Empirical Modelling variant JS-EDEN may provide a feasible option for deploying a more reliable and modern version of the OME. Indications for suitability of the JS-EDEN for such a development [27], as well as our initial thoughts [28] for this approach in the OME can already be found in the literature. As the JS-EDEN is a HTML5 and JavaScript application, it provides support for a wider selection of platforms, including tablet-based computing environments (iOS, Android, Windows RT) whereas the EM version used in this research is limited to desktop use only. Alongside with this development, it will be topical in the future research to study how the OME supports teachers in contexts other than educational robotics.

References

[1] S. Amershi and C. Conati, "Combining Unsupervised and Supervised Classification to Build User Models for Exploratory Learning Environments," *Journal of Educational Data Mining* **1,** 18–71 (2009).

[2] J. Bada and J. Suhonen, "Developing and evaluating computer games: a pedagogical perspective for HIV/AIDS prevention in schools," *Problems of Education in the 21st Century* **33,** 9–27 (2011).

[3] R. S. Baker and K. Yacef, "The State of Educational Data Mining in 2009: A Review and Future Visions," *Journal of Educational Data Mining* **1,** 3–17 (2009).

[4] D. Baum and R. Zurcher, *Definitive Guide to Lego Mindstorms*, Vol. 2, (Apress, 2003).

[5] M. Beynon, "Modelling with Experience: Construal and Construction for Software," in *Ways of Thinking, Ways of Seeing*, Vol. 1, C. Bissell and C. Dillon, eds. (Springer Berlin Heidelberg, 2012), pp. 197-228.

[6] M. Beynon and C. Roe, "Computer Support for Constructionism in Context," in *Proceeding of the 4th IEEE International Conference on Advanced Learning Technologies (ICALT)* (2004), pp. 216–220.

[7] F. Castro, A. Vellido, A. Nebot, and F. Mugica, "Applying Data Mining Techniques to e-Learning Problems," in *Evolution of Teaching and Learning Paradigms in Intelligent Environment*, Vol. 62, L. Jain, R. Tedman, and D. Tedman, eds. (Springer Berlin Heidelberg, 2007), pp. 183–221.

[8] D. Cooper, I. Arroyo, B. Woolf, K. Muldner, W. Burleson, and R. Christopherson, "Sensors Model Student Self Concept in the Classroom," in *User Modeling, Adaptation, and Personalization*, Vol. 5535, G.-J. Houben, G. McCalla, F. Pianesi, and M. Zancanaro, eds. (Springer Berlin Heidelberg, 2009), pp. 30-41.

[9] N. Correll, R. Wing, and D. Coleman, "A One-Year Introductory Robotics Curriculum for Computer Science Upperclassmen," *IEEE Transactions on Education* **56,** 54–60 (2013).

[10] P. De Cristoforis, S. Pedre, M. Nitsche, T. Fischer, F. Pessacg, and C. Di Pietro, "A Behavior-Based Approach for Educational Robotics Activities," *IEEE Transactions on Education* **56,** 61–66 (2013).

[11] J. R. del Solar and R. Avilés, "Robotics courses for children as a motivation tool: the Chilean experience," *IEEE Transactions on Education* **47,** 474–480 (2004).

[12] B. Denis and S. Hubert, "Collaborative learning in an educational robotics environment," *Computers in Human Behavior* **17,** 465 – 480 (2001).

[13] C. Despres and S. George, "Supporting learners' activities in a distance learning environment," *International Journal of Continuing Engineering Education and Life Long Learning* **11,** 261–272 (2001).

[14] B. S. Elenbogen and N. Seliya, "Detecting outsourced student programming assignments," *Journal of Computing Sciences in Colleges* **23,** 50–57 (2008).

[15] "Empirical Modelling Research Group," `http://www2.warwick.ac.uk/fac/sci/dcs/research/em/` (2013), accessed August 22, 2013.

[16] P. J. Eronen, I. Jormanainen, E. Sutinen, and M. Virnes, "Kids' Club Reborn: Evolution of Activities," in *The 5th IEEE Interna-*

tional Conference on Advanced Learning Technology (ICALT 2005) (2005), pp. 545–547.

[17] P. J. Eronen, I. Jormanainen, and M. Virnes, "Virre - Virtual Reflecting Tool," in *Third Annual Finnish/Baltic Sea Conference on Computer Science Education* (2003), pp. 42–47.

[18] P. J. Eronen, E. Sutinen, M. Vesisenaho, and M. Virnes, "Kids' Club as an ICT-based Learning Laboratory," *Informatics in Education* **1,** 61–72 (2002).

[19] B. S. Fagin and L. Merkle, "Quantitative analysis of the effects of robots on introductory Computer Science education," *Journal on Educational Resources in Computing* **2,** 1–17 (2002).

[20] M. Ferrari and G. Ferrari, *Building robots with LEGO Mindstorms NXT* (Syngress, 2011).

[21] "Geddit," `http://www.letsgeddit.com` (2013), accessed August 22, 2013.

[22] R. L. Glass, V. Ramesh, and I. Vessey, "An analysis of research in computing disciplines," *Communications of ACM* **47,** 89–94 (2004).

[23] J. Gobert, M. Sao Pedro, R. Baker, E. Toto, and O. Montalvo, "Leveraging Educational Data Mining for Real-time Performance Assessment of Scientific Inquiry Skills within Microworlds," *Journal of Educational Data Mining* **4,** 111–143 (2012).

[24] W. Hafner and T. Ellis, "Project-Based, Asynchronous Collaborative Learning," in *Proceedings of the 37th Annual Hawaii International Conference on System Sciences* (2004), p. 10015a.

[25] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. Witten, "The WEKA Data Mining Software: An Update," *SIGKDD Explorations* **11** (2009).

[26] A. Harfield, *Empirical Modelling as a new paradigm for educational technology*, PhD thesis (Department of Computer Science, University of Warwick, 2008).

[27] A. Harfield and M. Beynon, "Empirical Modelling for constructionist learning in a Thai secondary school mathematics class," in *Proceedings of the Ninth International Conference on eLearning for Knowledge-Based Society* (2012).

[28] A. Harfield, I. Jormanainen, and J. Rungrattanaubol, "An open monitoring environment for primary school children engaged in tablet-based learning," in *The 10th International Joint Conference on Computer Sciences and Software Engineering (JCSSE 2013)* (2013), accepted for publication.

[29] I. Jormanainen, M. Beynon, and E. Sutinen, "Understanding open learning processes in a robotics class," in *Proceedings of the 9th Koli Calling International Conference on Computing Education Research* (2009), pp. 51–54.

[30] I. Jormanainen, M. Beynon, and E. Sutinen, "An abductive environment enables teachers intervention in a robotics class," in *Proceedings of the 17th International Symposium on Artificial Life and Robotics (AROB 2012)* (2012), pp. 1075–1078.

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

[32] I. Jormanainen, A. Harfield, and E. Sutinen, "Supporting Teacher Intervention in Unpredictable Learning Environments," in *The 9th IEEE International Conference on Advanced Learning Technologies (ICALT 2009)* (2009), pp. 584–588.

[33] I. Jormanainen, O. Kannusmäki, and E. Sutinen, "IPPE - How to Visualize Programming with Robots," in *Second Program Visualization Workshop* (2002), pp. 69–73.

[34] I. Jormanainen and E. Sutinen, "Using Data Mining to Support Teacher's Intervention in a Robotics Class," in *Proceedings of the Fourth IEEE International Conference on Digital Game and Intelligent Toy Enhanced Learning* (2012), pp. 39–46.

[35] I. Jormanainen and E. Sutinen, "Role blending in a learning environment supports facilitation in a robotics class," *Journal of Educational Technology and Society* (2013), accepted for publication.

[36] I. Jormanainen, Y. Zhang, Kinshuk, and E. Sutinen, "Pedagogical Agents for Teacher Intervention in Educational Robotics Classes: Implementation Issues," in *Proceedings of the First IEEE International Workshop on Digital Game and Intelligent Toy Enhanced Learning (DIGITEL 2007)* (2007), pp. 49–56.

[37] E. Kärnä-Lin, K. Pihlainen-Bednarik, E. Sutinen, and M. Virnes, "Technology in Finnish Special Education - Toward Inclusion and Harmonized School Days," *Informatics in Education* **6,** 103–114 (2007).

[38] J. F. Kelly, *Lego Mindstorms NXT-G Programming Guide* (Apress, 2010).

[39] H. Kinshuk, Hong, and A. Patel, "Adaptivity through the Use of Mobile Agents in Web-based Student Modelling," *International Journal of E-Learning* **1,** 55–64 (2002).

[40] A. Krištofič and M. Bieliková, "Improving adaptation in web-based educational hypermedia by means of knowledge discovery," in *Proceedings of the sixteenth ACM conference on Hypertext and hypermedia*, HYPERTEXT '05 (2005), pp. 184–192.

[41] J. R. Landis and G. G. Koch, "The measurement of observer agreement for categorical data," *Biometrics* **33,** 159–174 (1977).

[42] D. Laverde et al., *Programming Lego Mindstorms with Java* (Syngress, 2002).

[43] H. Legeiwe, "How to Use Computers in Qualitative Analysis," in *Basics of Qualitative Research: Techniques and Procedures for Developing Grounded Theory* (SAGE, Thousand Oaks, California, 1996).

[44] H. Lieberman, F. Patern, M. Klann, and V. Wulf, "End-User Development: An Emerging Paradigm," in *End User Development*, Vol. 9, H. Lieberman, F. Patern, and V. Wulf, eds. (Springer Netherlands, 2006), pp. 1-8.

[45] J. Maloney, M. Resnick, N. Rusk, B. Silverman, and E. Eastmond, "The Scratch Programming Language and Environment," *ACM Transactions on Computing Education* **10,** 16:1–16:15 (2010).

[46] R. Marx, C. Blumenfield, J. Krajcik, and E. Soloway, "Enacting Project-Based Science," *The Elementary School Journal* **97,** 341–358 (1997).

[47] F. Mhiri and S. Ratté, "AARTIC: development of an intelligent environment for human learning," in *Proceedings of the 14th annual ACM SIGCSE conference on Innovation and technology in computer science education (ITiCSE)* (2009), pp. 359–359.

[48] S. P. Muir and D. M. Tracy, "Collaborative Essay Testing: Just Try It!," *College Teaching* **47,** 33–35 (1999).

[49] K. Newman, J. Hamblen, and T. Hall, "An introductory digital design course using a low-cost autonomous robot," *IEEE Transactions on Education* **45,** 289–296 (2002).

[50] R. Nkambouand, J. Bourdeau, and R. Mizoguchi, eds., *Advances in Intelligent Tutoring Systems* (Springer-Verlag Berlin Heidelberg, 2010).

[51] S. Papert, *Children's Machine: Rethinking School in the Age of the Computer* (Basic Books, New York, 2003).

[52] N. W. Pope, *Supporting the Migration from Construal to Program: Rethinking Software Development*, PhD thesis (Department of Computer Science, University of Warwick, UK, 2011).

[53] F. Provost, T. Fawcett, and R. Kohavi, "The Case against accuracy estimation for comparing induction algorithms," in *Proceedings of the Fifteenth International Conference on Machine Learning* (1998), pp. 445–453.

[54] L. Qiu and C. Riesbeck, "An incremental model for developing computer-based learning environments for problem-based learning," in *Proceedings of 4th IEEE International Conference on Advanced Learning Technologies (ICALT)* (2004), pp. 171–175.

[55] J. Quinlan, *C4.5: Programs for Machine Learning* (Morgan Kaufmann Publishers, 1993).

[56] J. J. Randolph and P. J. Eronen, "Developing the Learning Door: A case study in youth participatory program planning," *Evaluation and Program Planning* **30,** 55 – 65 (2007).

[57] J. J. Randolph, M. Virnes, I. Jormanainen, and P. J. Eronen, "The effects of a computer-assisted interview tool on data quality," *Educational Technology & Society* **9,** 195–205 (2006).

[58] T. Reeves, "Enhancing the Worth of Instructional Technology Research through "Design Experiments" and Other Development Research Strategies," in *AERA Annual Meeting* (2000), pp. 1–15.

[59] J. Ross, "Informatics creativity: a role for abductive reasoning?," *Communications of ACM* **53,** 144–148 (2010).

[60] E. Sklar, A. Eguchi, and J. Johnson, "RoboCupJunior: learning with educational robotics," *AI Magazine* **24** (2003).

[61] E. Sklar, S. Parsons, and P. Stone, "Using RoboCup in university-level computer science education," *Journal on Educational Resources in Computing* **4,** 1–21 (2004).

[62] E. Sklar and D. Richards, "The use of agents in human learning systems," in *Proceedings of the fifth international joint conference on Autonomous agents and multiagent systems*, AAMAS '06 (2006), pp. 767–774.

[63] L.-K. Soh, N. Khandaker, X. Liu, and H. Jiang, "A computer-supported cooperative learning system with multiagent intelligence," in *AAMAS '06: Proceedings of the fifth international joint conference on Autonomous agents and multiagent systems* (2006), pp. 1556–1563.

[64] M. Spahn, C. Dorner, and V. Wulf, "End User Development: Approaches Towards a Flexible Software Design," in *ECIS 2008 Proceedings* (2008).

[65] A. Troynikov, "SciFest and Geddit workshop," personal communication (2013).

[66] L. Werner, C. McDowell, , and J. Denner, "A First Step in Learning Analytics: Pre-processing Low-Level Alice Logging Data of Middle School Students," *Journal of Educational Data Mining* (2013), in press.

[67] A. Williams, "The qualitative impact of using LEGO MINDSTORMS robots to teach computer engineering," *IEEE Transactions on Education* **46,** 206 (2003).

[68] I. Witten and E. Frank, *Data mining: practical machine learning tools and techniques with Java implementations* (Morgan Kaufmann Publishers, 2000).

[69] M. Wooldridge and N. R. Jennings, "Intelligent Agents: Theory and Practice," *The Knowledge Engineering Review* **10,** 115–152 (1995).

[70] Y. Zhang, Kinshuk, I. Jormanainen, and E. Sutinen, "An Implementation of Agency Architecture in Educational Robotics," in *The 8th IEEE International Conference on Advanced Learning Technologies (ICALT 2008)* (2008), pp. 194–198.

Ilkka Jormanainen

Supporting Teachers in Unpredictable Robotics Learning Environments

Modern technology-enhanced learning environments raise the need for flexible tools for analysing learning data. This dissertation presents the development of an Open Monitoring Environment, which utilises a novel data mining method for building contextualised models for observing learning processes in educational robotics classes. The results can be extended to serve monitoring and analysing learning data in different contexts through a novel Conflative Learning Environment approach.

UNIVERSITY OF
EASTERN FINLAND

Publications of the University of Eastern Finland
Dissertations in Forestry and Natural Sciences

isbn: 978-952-61-1230-5 (printed)
issnl: 1798-5668
issn: 1798-5668
isbn: 978-952-61-1231-2 (pdf)
issnl: 1798-5668
issn: 1798-5668