User Modelling and Adaptation in Exploratory Learning

Mihaela Cocea

September 2011

A Dissertation Submitted to
Birkbeck College, University of London
in Partial Fulfillment of the Requirements
for the Degree of Doctor of Philosophy

Department of Computer Science & Information Systems
Birkbeck College
University of London

Declaration

I declare that this thesis was composed by myself and that the work contained herein is my own except where explicitly stated otherwise in the text.

Signature

(Mihaela Cocea)

Abstract

User modelling in Exploratory Learning Environments (ELEs) is an emerging field with several challenges to be addressed. Due to the freedom given to learners, the amount of information generated is very large, making the modelling process very challenging. Consequently, only relevant information should be used in the user modelling process. This, however, leads to other challenges such as identification of relevant information, finding an optimal knowledge representation and defining an inference mechanism by which this knowledge is used in diagnosing the learner.

This thesis addresses the challenges of user modelling in ELEs by monitoring learners' behaviour and taking into account only relevant actions in the context of an ELE for the domain of mathematical generalisation. An iterative approach was used, in line with the iterative design of the ELE. The modelling mechanism employed a modified version of Case-based Reasoning (CBR) and was evaluated using pedagogical scenarios and data from simulated and real students. This approach has the advantage of storing only relevant information and allows learner diagnosis during as well as at the end of a task.

The user model was further exploited to support learning related activities, such as prioritising feedback and grouping for collaboration. For feedback prioritisation, a mechanism based on Multi-criteria Decision Making was developed and tested with the help of educational experts. The grouping for collaboration approach was inspired from Group Technology, a method from cellular manufacturing systems, and its testing showed it produces meaningful groups. Both the feedback prioritisation and the grouping for collaboration mechanisms propose solutions that are particularly relevant for ELEs by considering pertinent criteria for this type of learning.

To ensure optimal coverage of the knowledge base, the user modelling approach was enhanced with adaptive mechanisms for expanding the knowledge base, which was tested on real and simulated data. This approach ensures that learner diagnostic is possible when the initial knowledge base is small and/or new behaviours are encountered over time.

Acknowledgments

Several people supported me in conducting and completing the research presented in this thesis.

First and foremost, I would like to thank my supervisor Prof. George D. Magoulas for his support and guidance throughout all the stages of this research. He has been a real mentor, helping me improve my scholarly thinking and research skills. I am grateful for his encouragement and support, and for the invaluable time spent discussing research ideas.

I would like to thank the members of the MiGen team for their constructive comments and suggestions. Darren Pearce and Sergio Gutiérrez helped me understand the information logged by the system. Darren also allowed me to use some of his images of the system and Sergio helped me understand the changes between the various versions of the system. Thanks are due to Manolis Mavrikis and Eirini Geraniou for sharing their knowledge of mathematics education in general and mathematical generalisation in particular.

I wish to thank my family and friends in Romania for their encouragement and belief in my abilities. Also, special thanks to my husband and his family for their love, patience, understanding and support.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

User modelling involves an inference process from observable data about a user such as his or her actions, to unobservable information (Zukerman and Albrecht, 2001) such as his or her preferences or intentions. In the context of learning environments the users are learners [1] and the main unobservable inferred information is about the learners' knowledge of a particular domain (Wenger, 1987). There are many reports in the literature on learner modelling research in the area of Intelligent Tutoring Systems (e.g. VanLehn (1988); Anderson et al. (1995); Conati et al. (2002); Mitrovic (2003); VanLehn et al. (2005); Kerly et al. (2008); Baker et al. (2010)), but relatively few in other types of learning environments. In recent years learner modelling in Exploratory Learning Environments (ELEs) has emerged as a new field of research with several challenges to be addressed.

Exploratory learning environments are build based on a constructionist pedagogical approach (Papert, 1993), based on two core ideas: (a) learning is seen as a reconstruction of knowledge rather than as a transmission of knowledge and (b) learning is most effective when it is part of an activity in which learners feel they are constructing a meaningful product (Papert, 1993). The constructionist approach is inspired by Piaget's constructivist theory (Piaget, 1950) which states that learners construct mental models to understand the world around them. Consequently, based on these principles, exploratory learning environments allow learners a high degree of freedom and encourage learners to explore and experiment with different models within the particular learning system. Therefore, these environments are radically different from Intelligent Tutoring Systems in which the learning activities are highly structured and the learner is guided in a stepwise manner.

[1]Depending on the setting, teachers or tutors are also users of such environments; here, however, we focus on the learners as the main users of such systems.

Exploratory or discovery learning has been argued to be particularly beneficial (de Jong and van Joolingen, 1998) in terms of providing opportunities for acquiring deep conceptual and structural knowledge. Pure discovery learning without any guidance and support is, however, less effective than more constrictive environments where the learner is guided step-by-step (Kirschner et al., 2006).

The main challenge in exploratory learning is to balance freedom with control: learners should be given enough freedom so that they can actively engage in constructing and exploring models, but, at the same time, they should be offered enough guidance to assure that their constructions and explorations lead to useful knowledge (Mayer, 2004). Besides this clear and well-acknowledged challenge there are other issues that make the process of learner modelling in exploratory learning environments demanding:

(a) *What to model?* Usually learner models relate to knowledge or skills. In the context of exploratory learning, the knowledge results from constructionist processes and there is a clearer indication of this knowledge at the end of these processes. Nevertheless, support is required both during knowledge construction and at the end of certain processing stages. Thus, a key question is what to model so that the system can produce a representation of the learner's current state of knowledge from the constructions built so far, and, consequently, can provide support during and at the end of the knowledge construction process.

(b) *Value of correct vs. incorrect actions.* In most e-Learning systems, the system's response is related to correctness or incorrectness of answers or actions, while in ELEs learners' explorations are difficult to categorise into correct or incorrect. Moreover, even if such a classification would be possible, incorrect actions may be more valuable for learning than correct ones. In fact, one of the advantages of ELEs is that learners are given the opportunity to realise their own mistakes and learn from them. Thus, rather than pointing out possible mistakes, the system should provide learners with feedback that would encourage reflection on their actions and help them realise that their knowledge construction is not entirely correct.

(c) *Relation between knowledge of abstract concepts and forms of (re)presentation in the system.* ELEs have different ways of (re)presenting and exploring models that should gradually help learners build knowledge of abstract concepts. Each part of the model and each type of exploration (e.g. changing parameters, creating new models, testing models etc.) contributes to this process. Therefore, identification of the relevant abstract concepts is needed as well as their representation in the learner model.

(d) *Identification of underlying strategies from actions or sequences of actions.* Sometimes it is neither realistic nor feasible to include all possible outcomes (correct or incorrect) and ways to achieve them when modelling an extensive knowledge domain. Therefore, a different approach to what is included in the knowledge structure is required. Rather than storing complete information about a task or expert knowledge, key information with informative educational value could be stored, such as strategies for approaching a task, and landmarks indicating a particular strategy or (lack of) knowledge about a particular aspect. The challenge is how to find this information and how to represent it in the knowledge structure.

Given the above mentioned challenges, a classic approach to learner modelling based on concepts would not fit the purposes of ELEs. The classic approach involves a particular scenario: learners are required to study materials about a concept and then their knowledge level is assessed through testing. On the contrary, ELEs involve knowledge discovery by means of constructionist activities and the emphasis is on the process rather that the knowledge itself. Therefore, the learner modelling process should reflect this way of learning. The nature of this process places the focus on the interactions of the learner with the system rather than on their answers to tests. Thus, analysing interactions during knowledge construction and extracting relevant information is an essential part of the learner modelling process that together with knowledge about student's learning processes inferred from their models and their learning progression can play an important role in generating feedback and support.

Individual learner models can be used in ELEs for several purposes:

(a) individual personalised support - provide feedback on individual exploration.

(b) personalised sequencing of activities - depending on knowledge level (but possibly other individual characteristics as well, like motivation for example), the sequence of activities could be personalised. For example, for high ability students, the sequences of activities could include more difficult tasks, while for less able students or novices, the sequence could contain tasks with a lower degree of difficulty (that could be increased gradually). This would be very useful in a classroom with pupils of different ability levels.

(c) collaboration and collaboration support - a learner that has difficulties in his/her exploration could benefit from collaborating with a more able peer (Vygotsky, 1978) exploring the same thing. The individual learner models could be used for grouping learners for collaboration on educational criteria like the 'more able peer' mentioned before; other criteria could be: complementary skills, the stage in a task, the solutions to a task (e.g. two learners that have reached two equivalent solutions using different approaches would benefit from collaboration).

(d) teacher support - inform the teacher about individual learners' progressions and difficulties; give suggestions to teachers of possible courses of action for a specific situation (individual or collaborative). Aggregated information about the whole class could be provided to the teacher, so that s/he can see what are the common solutions found by pupils and identify 'outliers', i.e. pupils that are doing things considerably different from the majority, both correct (e.g. an uncommon solution) and incorrect (e.g. wrong track on solving the given problem).

The rest of this chapter is structured as follows. The next section presents the research questions of the PhD, Section 1.2 outlines the structure of the thesis while Section 1.3 lists the contribution of this PhD research.

1.1 Research questions

The focus of the PhD thesis is on learner modelling in ELEs. The aim is to develop a mechanism that allows modelling of behaviour: (a) during task solving and (b) at the end of tasks. The information stored in the learner model could be used for several purposes, such as: personalised individual feedback, personalised feedback for groups of learners in the context of collaborative learning, informing the teacher about the progress of individual learners as well as groups of learners, presenting the learners with the content of their learner models, prioritisation of feedback, detection of off-task behaviour, forming groups for collaborative activities. From this wide range of possible ways of exploiting the learner model, the PhD research includes the last three.

The research questions are the following:

1. How can relevant interactions be transformed into informative data to be stored in the learner model?

2. What should be stored in the learner model in order to represent the evolution of the learner's constructionist models and their corresponding cognitive processes?

3. How should the learner model be updated in order to reflect both the current knowledge and the evolution of knowledge?

4. How can the learner model be exploited for pedagogical purposes? E.g., how to prioritise between several aspects that require feedback? how to group learners for collaborative activities? how to detect off-task behaviour?

5. How to ensure the information used in the learner modelling mechanism is maintained to provide accurate diagnosis over time?

1.2 Structure of Thesis

The thesis contains literature reviews and research that could be categorised as part of three main strands of work: the development of a learner modelling mechanism (Chapter 2 and 4), exploiting the learner model for pedagogical purposes (Chapter 5 and 6) and enriching the knowledge base (Chapter 7).

Chapter 2 presents the literature review on exploratory learning, user/learner modelling and learner modelling for exploratory learning. The aim of this chapter is to outline the current knowledge about the processes involved in exploratory learning, to give an overview of user modelling in general and learner modelling in particular and, finally, to present previous research in the field of learner modelling for exploratory learning environments. This literature review has been conducted with one goal in mind: to provide us with the necessary knowledge to take an informed decision on the best approach that we can take for learner modelling in the context of our learning environment. This chapter informed the development of our proposed approach for learner modelling by providing the knowledge of the processes involved in discovery learning and the challenges they pose both from modelling point of view and pedagogical point of view, and by informing us on the suitability of the various methods previously used in user/learner modelling both in exploratory learning environments as well as other types of learning environments.

Chapter 3 presents the methodological consideration of the research presented in this thesis. A typical mathematical generalisation problem is presented to facilitate the understanding of the aims and objectives, and of the challenges for user modelling. The context in which the research was conducted is described and the need for iterative design is explained; an overview of iterative design is also included. The methodology is presented for each strand of research in the thesis and an abstract conceptual model is given for the main strand of research, i.e. user modelling.

Chapter 4 presents our proposed approach for learner modelling in an exploratory learning environment for the domain of mathematical generalisation, which was developed in an iterative and incremental manner. The chapter also includes overviews of the domain we are working with, i.e. mathematical generalisation, and of case-based reasoning which is used in our modelling approach. Two versions of the system and of the learner modelling mechanism are presented together with their corresponding evaluation studies. The results are discussed and the chapter ends with a summary and an overview of contribution. The approach presented in this chapter addresses our first three objectives, i.e. identify the relevant information to be stored in the learner model, represent the evolution of the learners' knowledge and develop an updating mechanism. Furthermore, it constitutes the bases for the research presented in the following three chapters.

Chapter 5 and 6 present two exploitations of the learner model for pedagogical purposes: feedback prioritisation and grouping for collaboration. Chapter 5 starts with presenting the problem of feedback prioritisation, continues with an overview of personalised feedback and an overview of Multi-criteria Decision Making focusing on a method called the Analytic Hierarchy Process which is employed in our approach, and, finally presents the two iterations of our proposed mechanism for feedback prioritisation. Evaluation studies were conducted for both versions, and the results are presented and discussed. Chapter 6 starts with the problem of grouping for collaborative activities and previous work in this area, continues with a brief overview of Group Technology from which our approach is inspired, presents our mechanism and its evaluation, and discusses the results. Both chapters end with a summary and an outline of the contribution, and both of them address our fourth objective, i.e. exploit the information in the learner model for pedagogical purposes.

Chapter 7 presents work that aims to improve the coverage of the knowledge base over time. It starts with an overview of adaptive modelling and continues with our proposed mechanism for enriching the knowledge base. Several evaluation studies are presented and discussed, and, finally, the chapter ends with a summary and an overview of the contribution. The research presented in this chapter addresses our last objective, i.e. maintain optimal coverage of the knowledge base.

Finally, Chapter 8 concludes this thesis with a summary of research and findings, and an outline of the thesis contribution. It also identifies directions for future work and ways in which they could be addressed.

1.3 Contribution

In Chapter 4, a learner modelling approach for exploratory learning of mathematical generalisation based on Case-Based Reasoning (CBR) is proposed. The classic CBR approach has been modified to cover problems with multiple solutions, in which the target is to identify which particular solution is used. The proposed CBR-based mechanism identifies the approach taken by a learner for a particular task by comparing the learner's approach with all the approaches stored in the knowledge base for that task. To evaluate the mechanism a set of scenarios of pedagogical importance were identified; the mechanism has been successfully tested on partial and complete approaches, as well as mixed approaches. This way of modelling the learner allows diagnosis *during* the task, as well as at the end of it, thus giving opportunities for meaningful personalised support. Moreover, the mechanism was developed in an iterative and incremental manner in parallel to other components of the system, including the interface. Therefore, this chapter also contributes to the methodological aspects of designing and developing a learner modelling component.

Chapter 5 addresses a problem that is common to exploratory learning environments, i.e. personalised feedback prioritisation. To address this problem, an approach was developed based on the Analytic Hierarchy Process, a method from Multi-criteria Decision Making. The approach takes into consideration several criteria such as the context within a particular task and outputs an ordering among the several task-related aspects on which feedback may me needed. This approach ensures that when there are several aspects to give feedback on, the most relevant ones are prioritised, as displaying feedback on all aspects would be hardly beneficial for the learner and would potentially be more confusing than helpful. Although designed for our particular educational environment, this approach could be applied to other educational systems. The criteria and alternatives, however, would need to be replaced with relevant aspects for the particular environment.

Chapter 6 also addresses a problem encountered in educational environments in general and exploratory learning environments in particular, i.e. grouping students for collaborative activities. Because the students' interactions with an exploratory learning environment are different from interactions with other types of learning environments such as intelligent tutoring systems, different criteria are important when grouping students for collaborative activities. As exploratory learning often entails being aware of alternative approaches to the same problem, the criteria we propose are the approaches used by individual students and the similarities between various approaches for the particular task. Our proposed mechanism outputs clusters of learners using similar approaches, which could be then used by teachers in different ways depending on their aims and their knowledge about the students. Besides the flexibility offered to the teacher, our approach can be used to form both homogeneous and heterogeneous groups.

In Chapter 7 we propose a mechanism for maintaining the optimal coverage of the knowledge base using adaptive modelling. This approach has the advantage of being able to start with a small knowledge base and gradually enrich it. Also, even if initially there is a large knowledge base, over time learners discover different ways of approaching the tasks, rendering the knowledge base suboptimal for generating proper feedback, despite the initially good coverage. With our approach, optimal coverage of the knowledge base is maintained by gradually incorporating the new approaches when they occur.

Chapter 2

Learner Modelling for Exploratory Learning

The term Learner Modelling refers to the process of generating a learner model (Morales Gamboa, 2000) in the context of an intelligent learning environment (Brusilovsky, 1994). A learner model enables the system to adapt to the learner who uses it and ideally includes all information about the learner's behaviour and knowledge that influences their learning and performance (Wenger, 1987). The content of a learner model depends on the learning environment and includes inferred information about aspects such as a learner's goals, plans, knowledge, attitudes and abilities, but the most important information about a learner is his or her knowledge of the subject that is being studied (Brusilovsky, 1994).

Although there are many works describing various approaches for learner modelling in the area of Intelligent Tutoring Systems (VanLehn, 1988; Anderson et al., 1995; Conati et al., 2002; Mitrovic, 2003), there is relatively little research in the area of Exploratory Learning Environments. The aim of this chapter is to present the characteristics of exploratory learning that make them radically different from tutoring systems and, consequently, require a different approach for learner modelling. The literature review presented in this chapter offers a wider context for our aims and objectives and motivates the research questions presented in the Introduction. Moreover, it also situates our work in relation to previous research on learner modelling for exploratory learning environments.

The chapter is divided in three subsections: (a) a literature review on *exploratory learning* where some theoretical aspects of exploratory learning are presented; (b) a literature review on *learner modelling*, where a brief description of the process and a discussion on learner modelling possibilities and usefulness in ELEs are included, and (c) a detailed presentation of the *previous research* *on learner modelling in ELEs.* Finally, the chapter ends with a summary of its contents.

2.1 Exploratory Learning

Exploratory learning is a teaching approach that encourages the learners to explore the domain with much less focus on guided teaching and presentation of already processed information; it is based on the constructionist theory of learning (Papert, 1993). According to Rieber (2004) exploratory learning is founded on the following four principles:

(a) Learners can take control of their learning and are encouraged to do so.

(b) The domain to explore is rich and multidimensional.

(c) The domain should allow exploration in various ways (e.g. multiple equivalent solutions to the same problem).

(d) The learners should be given freedom to explore rather than being guided on a specific path.

Often other names are used for this type of learning, among which are *discovery learning* and *inquiry learning*:

- "Discovery learning is a type of learning where learners construct their own knowledge by experimenting with a domain, and inferring rules from the results of these experiments. The basic idea of this kind of learning is that because learners can design their own experiments in the domain and infer the rules of the domain themselves they are actually constructing their knowledge. Because of these constructive activities, it is assumed they will understand the domain at a higher level than when the necessary information is just presented by a teacher or an expository learning environment." (van Joolingen, 1999, p. 385).

- "Inquiry is an approach to learning that involves a process of exploring the natural or material world, and that leads to asking questions, making discoveries, and rigorously testing those discoveries in the search for new understanding" (National Science Foundation, 2000, p. 2).

Exploratory learning does not completely exclude guidance or constraints; their usage is only different from step-by-step guided learning. The focus is on the fact that the learners finds his/her own path towards a solution and this process includes errors that are beneficial for learning and need not necessarily be corrected before moving to a next step (as it is the case with guided learning).

Discovery learning has been regarded as a 'search problem' within two related spaces (Newell and Simon, 1972; Simon and Lea, 1974; Klahr and Dunbar, 1988): hypotheses space and experiment space. Hypotheses direct the search in the experiment space and the results of experiments influence the search of new (refined) hypotheses. Klahr and Dunbar (1988) describe two search strategies within the two spaces:

(a) *Theorist strategy* starts with a set of hypotheses which are tested and refined based on the results of the experiments; the new hypotheses follow the same course.

(b) *Experimenter strategy* starts with data collection before stating a hypothesis.

The main difference between the two strategies is in prior knowledge (Klahr and Dunbar, 1988). Thus, if a learner can state a hypothesis, s/he is likely to use a theorist approach; if a hypothesis cannot be stated, a learner tends to experiment and observe data and thus, takes an experimenter approach. Hence, where there is no or little prior knowledge the trajectory seems to be from experimenter strategy to theorist strategy, while when having prior knowledge the trajectory includes only the theorist strategy.

van Joolingen and de Jong (1997) extended Klahr and Dunbar's theory by identifying two subspaces of the hypotheses space:

(a) *Learner hypotheses space* that contains all hypotheses that a learner can think of, in terms of variables of the domain and possible relations between them.

(b) *Learner search space* that consists only of the hypotheses considered by the learner as candidates for a particular situation.

For both above mentioned strategies, discovery/inquiry learning involves several cognitive processes (de Jong, 2006); corresponding skills for each process are required for successful learning:

(i) *Orientation*: identification of variables and of relations between them;

(ii) *Hypotheses generation*: formulating a set of statements about relations between variables;

(iii) *Experimentation*: changing values of variables, making predictions, interpreting outcomes;

(iv) *Conclusions*: establishing the validity of hypotheses.

Besides the above mentioned cognitive processes, there are meta-cognitive processes (de Jong, 2006) involved that also have influence on the learning outcomes: (i) *Planning*: outlining a plan of the discovery process; (ii) *Monitoring*: having and updating an overview of the experiments and what has been learned from them; (iii) *Evaluating*: reflection on the learning process and the knowledge that was acquired.

Research indicates that learners encounter difficulties with all these processes:

(i) *Orientation*: learners have problems with choosing the right variables (de Jong and van Joolingen, 1998);

(ii) *Hypotheses generation*: learners have difficulties in formulating testable hypotheses (de Jong and van Joolingen, 1998);

(iii) *Experimentation*:

- difficulties in linking experimental data with hypotheses, as previous ideas tend to persist even when contradicted by data (Chinn and Brewer, 1993);
- difficulties to translate the variable from hypothesis into observable variables in the experiment (Lawson, 2002);
- learners tend to design ineffective experiments, e.g. varying more variables at one time (Keselman, 2003);
- learners fail to make predictions and make mistakes in interpreting data (Lewis et al., 1993);

(iv) *Conclusions*: learners do not draw the right conclusions even when having well designed experiments (de Jong and van Joolingen, 1998);

(v) *Planning*: if they plan at all, learners tend to focus only on short-term planning (Manlove et al., 2006);

(vi) *Monitoring*: learners do not monitor adequately what they do (Manlove et al., 2006);

A particular meta-cognitive skill, self-explaining, i.e. generating explanations to oneself in relation to the studied material and with respect to the underlying domain knowledge (Chi et al., 1989), has been shown to be beneficial for learning (Hausmann and VanLehn, 2007) even when the self-explanation is incorrect (Chi, 2000). In Bunt et al. (2004) the students' self-explanatory behaviour was modelled in the context of an open learning environment and support for self-explanation was provided. Including eye-tracking information in the model of self-explanation has been shown to provide a more accurate assessment of self-explanation. Moreover, modelling self-explanation improves the assessment of student exploratory behaviour (Conati and Merten, 2007).

Working with graphical representations has been shown to be more beneficial than working with textual representation: the students working with graphical representation design more experiments with their own model, formulate more qualitative hypotheses and spend more time evaluating their model (Löhner et al., 2005). Working with a concrete rather than an abstract task has also been shown to be more beneficial for learning (Wilhelm and Beishuizen, 2003).

As ELEs differ in many respects from structured learning environments, a different way of supporting learning is required:

> "...in system controlled, expository environment, the question is which information to present based on the interaction with the learner, in a discovery environment the question is how to assist the learner in selecting and interpreting information from the learning environment" (van Joolingen, 1999, p.385).

Bliss recommend that the support for modelling should be done

> "according to a systematic strategy as well as to specific reasoning activities... a process support tool should offer a description of the processes that need to be performed as well as their interdependencies, but not fix the order of execution of these processes" (Bliss, 1994, p. 458).

Moreover, Bliss (1994) distinguishes two types of modelling:

(a) *Explorative modelling* in which learners explore a *given* model; this corresponds to discovery learning with computer simulations;

(b) *Expressive modelling* in which learners construct their own models.

Discovery learning should allow both types, thought most ELEs use predominantly the first one. The support in the context of this type of modelling makes use of *cognitive tools* (van Joolingen, 1999) that offer information to the learner (e.g. explanations related to the current problem), tools for structuring the task (e.g. notebooks) or for externalizing learning processes (e.g. provide the learners with a description of what they are doing (van Joolingen and de Jong, 1997)). More specifically, these tools are targeted to the key processes of discovery learning:

(i) *Hypothesis generation.* Help with the structure of hypotheses has been proposed by providing the learners with a menu of ready made hypotheses (Michael et al., 1989), which "*perform* the process for the learner" (van Joolingen, 1999, p. 390). Other tools "constrain or extend the search process in the hypothesis space" (van Joolingen, 1999, p. 390), like the *hypothesis scratchpad* (van Joolingen and de Jong, 1997) which helps the learner to formulate hypotheses by constraining the expressions they could state and by showing the hypothesis space to the learner; templates are offered to the learners in which they need to fill in variables and relations.

(ii) *Monitoring experiments.* The tool for monitoring experiments should have the following capabilities: add/delete experiments, sort experiments, replay experiments and change the order of variables. This sort of tool is useful after experiments have been done by the learner and its main function is to relieve the memory of the learner. A secondary function is the offering of an overview of the experiment space that has been searched.

van Joolingen (1999) also talks about *hooking intelligent support*, which refers to offering adaptive support, working with the input from the previously mentioned cognitive tools that make the learning process explicit. This idea was developed in Veermans and van Joolingen (1998, 2004) and Veermans (2003); the monitoring tool recorded the learners' experiments and provided feedback on the quality of these experiments and on whether they supported the hypothesis. The quality of experiments is assessed by a set of rules, while the hypothesis support is assessed using deduction principles from the hypothesis (given in the task) to the experiment space - the experiments' results are checked against the predictions generated by the hypothesis. More details are given in Section 2.3.

This section gave a brief overview of exploratory learning, the processes involved, the difficulties learners encounter with these processes and proposed ways of supporting the learners with cognitive tools corresponding to the processes of discovery learning.

2.2 Learner Modelling

A learner model is a representation of a learner and consists of data about the learner or about what the learner does. Typically, a learner model would store data about a learner's knowledge, preferences, goals, tasks, interest, etc. (Brusilovsky, 2001).

Learner modelling refers to the process of generating a learner model and it typically includes three main tasks (Morales Gamboa, 2000):

(a) *learner diagnosis* that refers to obtaining information about the learner to be included in the learner model;

(b) *model maintenance* which is about encoding and integrating the information in the model and

(c) *model employment* that refers to the usage of the model for various purposes that are meant to be beneficial for learning such as adaptive and personalised feedback, support for collaboration and support for teachers.

Learner modelling has to handle the gap between the information available about a learner and the conclusions that can be drawn from it, and this process includes uncertainty. Owing to the research in artificial intelligence (AI) and especially the development of techniques that handle uncertainty, the learner modelling research has advanced and currently there are many intelligent learning environments that use one or more AI techniques: case-based reasoning (e.g. Stottler and Ramachandran (1999); Han et al. (2005)), Bayesian networks (e.g. Bunt and Conati (2003); Conati et al. (2002)), fuzzy logic (e.g. Vrettos and Stafylopatis (2001)), neural networks (e.g. Beck and Woolf (1998)), genetic and evolutionary algorithms (e.g. Romero et al. (2003)), neuro-fuzzy (e.g. Stathacopoulou et al. (2003)), genetic algorithms and case-based reasoning (e.g. Huang et al. (2007)), etc. A review of recent advances using soft computing techniques for user-adaptive systems can be found in Frías-Martínez et al. (2005).

Discovery learning environments usually do not use learner modelling in the sense of creating a model of the learner's knowledge or skills. Arguments for not using such models are (Veermans, 2003):

(a) It is impossible to model the learner in an ELE because the amount of information would be too large given the freedom the learners have.

(b) The learner modelling is seen as contradictory to the principles of discovery learning and constructivism that considers that each learner has a personal representation of the external world.

According to Veermans (2003) these arguments may explain why traditional ITS techniques such as learner modelling have not been used for ELEs. Although these arguments are founded, counter arguments could be given:

(a) There is no need to model everything that the learner does; only *relevant* information could be stored and thus, considerably decrease the amount of information; also, considering that complete freedom is not beneficial for learning in ELEs (Kirschner et al., 2006) some constraints would be necessary and they would also contribute to limiting the learner's actions, thus decreasing the amount of information as well.

Veermans (2003) proposes a focus on the processes that lead to acquisition of knowledge rather that the knowledge itself; the role of the learner model would be in his view to support the learner in the process of discovery and:

> "To do this, the system would not need to maintain a full cognitive model of the learners domain knowledge, only to infer just enough from steps taken by the learner to support the learner in the process of discovery" (Veermans, 2003, p. 21).

(b) Personal representation does not contradict general knowledge; moreover, a learner model that would register the evolution of learning, with its variations and changes, would be in line with the constructivist theory and would reflect more accurately the learners' knowledge, but also its development.

2.3 Learner Modelling in Exploratory Learning Environments

Previous attempts in learner modelling for ELEs are very few and include: (a) the use of heuristics to guide the learning process in a physics domain (Veermans, 2003); (b) Bayesian networks in a mathematical functions domain (Bunt and Conati, 2003); (c) neuro-fuzzy systems for student diagnosis in a physics domain (Stathacopoulou et al., 2005); (d) Fuzzy sets for modelling cognitive states in a computer-based learning environment for Newtonian dynamics; (e) eye-tracking for modelling meta-cognitive characteristics such as self-explanation; (f) a Dynamic Decision Network approach for a dynamic learner model allowing reasoning about the learners behaviour and interventions across time.

Heuristics. The idea of intelligent support was tackled in Veermans (2003) using induction and deduction, whilst templates were used to generate feedback. The system used, SimQuest, is an authoring environment for developing simulation-based discovery learning environments. A tool was developed to support learning in this environment that is called "Learner modelling-Feedback generation" module; however the tool does not use a learner model in its 'classic' understanding of having an individual model of the domain knowledge (or skills) for each learner which is been updated and used for feedback; it only interprets the behaviour of a learner and uses the interpretation to provide feedback. Thus, the tool is more a 'feedback generation' than a 'learner modelling' module.

The research was conducted in the domain of physics. The principles of induction and deduction and two sources of information were used: (a) the experiments performed and (b) the investigated hypotheses. A stepwise procedure was applied:

1) a set of informative experiments about the relation between input and output variables is filtered from the complete set of performed experiments;

2) a set of informative experiments about the hypothesis is filtered from the previous set; for each hypothesis from the set of hypotheses, a set of informative experiments are selected.

3) for each of the informative sets for a hypothesis, predictions are generated for the output variables.

4) The predictions generated are compared to the values actually found in the experiments.

As previously mentioned, feedback is provided on correctness of hypotheses and on experimentation behaviour. The former serves two purposes: "it prevents construction of incorrect knowledge and serves as input for self-assessment of the exploration process" (Veermans, 2003, p. 43). For the latter, the monitoring tool available in SimQuest was extended. The monitoring tool stored the experiments performed by learners and allowed them to keep track of what they were doing. This tool was extended to provide support by: (a) drawing a graph that provides visualisation of variables, (b) giving feedback on drawing and interpreting graphs and (c) giving feedback on experimenting.

Two types of heuristics were used: general and specific. The general heuristics were used to asses the experiments of a learner, while the specific heuristics were employed to asses the experiments of a learner only if the learner fitted a function on the performed experiments.

One study showed that learners that were given feedback did more experiments compared to the ones that were not given feedback; also they did more unique experiments and tested more hypotheses. Thus, feedback seems to encourage exploratory behaviour.

Bayesian networks. The second approach addresses "effective exploration" (Bunt, 2001), but uses "standard" student modelling in the sense that essential cases for the problems to be explored are used as the equivalent of concepts in classic overlay models. The system used, Adaptive Coach for Exploration (ACE) is an intelligent open learning environment for the domain of mathematical functions (Bunt and Conati, 2003). It includes four modules:

1) a GUI that allows exploration of mathematical functions through three units:

(a) Machine Unit that allows exploration of the relation between input and output variables of a given function;

(b) The Arrow Unit has the same purpose as the Machine Unit, but requires more active thought from the learner that has to pair a specific input with a specific output;

(c) Plot Unit allows exploration of the relation between the graph and the equation of a function.

2) a Knowledge Base that includes concepts about functions and *relevant exploration cases* for each type of function (e.g. constant, linear, polynomial);

3) a Student Model that assesses the effectiveness of the student's exploration;

4) a Coach that provides feedback using the Student Model; two types of feedback are provided: on-demand hints and feedback when the learner moves to a new exercise without sufficient exploration of the current one.

The Student Model includes two separate Bayesian networks: one for the Machine and Arrow unit and one for the Plot Unit. Each network has: (a) *exploration nodes* that represent the effectiveness of the student's exploratory behaviour and (b) *knowledge nodes* that represent the student's understanding of the domain.

Two evaluation studies indicated that the hints generated using the Student Model assessment improved learning; also they gave evidence that the assessment of the Student Model reflected accurately the student's learning. However, the Student Model tended to over-estimate the effectiveness of exploration as with some students it has been observed that even if they performed the right set of actions they still failed to learn.

This last aspect may indicate that they learn the procedure, without fully grasping its purpose or meaning. This is similar to pattern spotting (Küchemann and Hoyles, 2006) that involves spotting a pattern in the way things work, but has nothing to do with reasoning on the actual problem to be solved.

In subsequent research self-explanation was added in the model as a predictor of exploratory behaviour (Bunt et al., 2004). Two types of self-explanation were detected: explicit self-explanation - self-explanation generated by the student by using the menu-based tools available in the interface and implicit self-explanation - generated by students in their heads. For the later time spent on each exploratory action was used as evidence of implicit self-explanation plus the presence of stimuli to self-explain: either the learner's general tendency to self-explain or a hint from the system to self-explain.

Neuro-fuzzy approach. The third approach (Stathacopoulou et al., 2005) combines fuzzy knowledge representation of expertise in teaching physics with training from practical examples when knowledge is not accurate or well-defined. It focuses on student diagnosis and feedback is not addressed.

The system, *Vectors in Physics and Mathematics*, is a discovery learning environment that was build based on the constructivist theory of learning (Grigoriadou et al., 1999a,b). It aims to help teachers and learners with the concepts of vectors in physics and mathematics in secondary schools. It has several thematic units that were considered based on knowledge about the conceptual difficulties of learning vectors and about the difficulties in conceptualising certain phenomenas from physics that could be sources of misconceptions.

A neural network-based fuzzy diagnosis model (Stathacopoulou et al., 2005) is used for the diagnosis process and updating the student model. The neuro-fuzzy approach aims to "imitate" the teachers' way of evaluation a learner's characteristics (capabilities, attitudes, knowledge level, motivation and learning style). Linguistic descriptions of students' behaviour and learning characteristics were elicited from teachers and encoded using fuzzy logic that handles the uncertainty associated with teachers' subjectivity. The neural network adds learning and generalisation capabilities to the fuzzy model.

The learner model provides information about the knowledge level on the domain concepts and learning style with the envisaged purpose of adapting the feedback and the pedagogical strategy to the learning style of students.

Experimental results (Stathacopoulou et al., 2005) using simulated students and teachers expertise showed that the neural network-based fuzzy diagnostic model successfully manages the uncertainty associated with human expertise in diagnosing students learning. Additionally, for marginal cases the model performs particularly well by synthesizing conflicting assessments. The model was also evaluated in real classroom conditions with an overall classification success of 68% (Stathacopoulou et al., 2007).

Fuzzy sets. A fuzzy algorithm was used to follow the cognitive states of students while they were solving a task and to interpret the monitored changes when working in a computer-based learning environment for Newtonian dynamics called FORCES (Andaloro and Bellomonte, 1998).

The students were required to solve problems at the qualitative level. They explored various systems of different complexity subjected to external forces. Each selected system was a task and the students were required to draw a diagram of the corresponding forces. Force vectors could be taken from a menu and applied to the system components. The system allowed comparison between the real motion and the modelled Newtonian motion. Students could repeat a task as many times as they liked.

The learning environment includes two modules: microworlds and student models. Microworlds are simulation that present various systems subjected to external forces. Student models contain representation of the student knowledge states - they are build up through rules that analyse the student.

A pilot study allowed the authors (Andaloro and Bellomonte, 1998) to elicit the student representations and the relationships with their behaviour. The students worked with the simulations and a tutor asked them to draw the corresponding diagrams and to explain their actions, and registered their reasonings and/or intuitions in interview protocols. Relevant diagrams and their mental representations were analysed and stored in the system. They have been restricted to models that were common to different students and consistent within individual responses.

The Student Model infers descriptions of correct, partial and incorrect student knowledge about the forces involved in the selected systems from the diagrams made by the students. The learning process is evaluated using an algorithm based on fuzzy sets. Dynamic fuzzy weights are calculated, assigning positive values to changes of mental representations that are considered significant for acquiring scientific knowledge and negative values to the ones considered insignificant.

Eye-tracking. Eye-tracking has been recently used to model meta-cognitive behaviour and adapt accordingly (Merten and Conati, 2006; Conati and Merten, 2007). The meta-cognitive behaviour included the capability to effectively learn from free exploration (de Jong and van Joolingen, 1998) and the capability to self-explain the material given for instruction. The system used was ACE, an intelligent learning environment for exploration-based learning of mathematical functions (described above).

Eye-tracking was used as an additional source of evidence for self-explanation behaviour. Empirical data on the correspondence between the students' self-explanation, time and attention patterns were collected and analysed. Time was found to be a predictor of self-explaining behaviour. Together with gaze shifts and gaze shifts with time, it was considered as a predictor of self-explanation behaviour. Results showed that gaze shifts alone were best at detecting the absence of self-explanation behaviour and that gaze shifts in combination with time were best at detecting the occurrence of self-explanation behaviour.

An evaluation study was conducted and confirmed that self-explanation was best predicted by gaze shifts in combination with time, followed by gaze shifts alone and, lastly, by time alone. It also showed that modelling self-explanation leads to a better performance of the model for diagnosing the students' exploratory behaviour.

Dynamic Decision Network. A Dynamic Decision Network (DDN) approach (Ting and Phon-Amnuaisuk, 2009) was used to model a dynamic learner model allowing reasoning about the learners' behaviour and interventions across time. The authors argued that DDN is a better approach in acquiring inquiry skills because these skills evolve over time, an aspect that would not be captured in a static network. Their research focused on the factors that influence the performance of a DNN learner model.

The system used is an intelligent computer-based scientific inquiry learning environment, INQPRO, for the physics domain. Skills like *formulating hypothesis* and *identifying and controlling variables* are probabilistically assessed. The system includes two modules: a GUI Module, that presents the learning materials via several GUIs and an animated pedagogical agent, and an Adaptation and Inference Module.

For each GUI, there is a correspondent decision network (DN) that models the static aspects of the inquiry skills. Two approaches were used to create the DDN: (a) combining the DNs of all GUIs and repeating them over time or (b) appending individual DNs corresponding to the accessed GUI. The first approach had the advantage of providing a unified solution and of no need to change the Conditional Probability Tables (CPTs) for the subnetworks that remained unchanged across time slices. The second approach produced fewer nodes per time-slice, allowing the DNN to be updated faster.

In a three-step evaluation of the DNN learner model, three factors were identified to influence the performance of the DNN: the structure of the DNN, the variable instantiation approach and the weights assignment method for two consecutive DNs.

The existing approaches have a number of strengths and limitations that are outlined in the following. Bayesian Networks are an established modelling technique in the context of intelligent tutoring systems; however, it does not fit our purpose as we are not dealing with concepts, but with learner actions. The neuro-fuzzy approach has the advantage of dealing well with uncertainty and of mimicking teachers' reasoning; again, concepts in the form of fuzzy variables are used which do not apply to our situation. Fuzzy sets also deal well with uncertainty, but have a similar drawback - they keep track of the overall knowledge, while we want to keep track of the models for each task in a more detailed manner than a number. The dynamic decision networks have the advantage of dealing with the temporal aspect; however, they are used for modelling skills, while we're interested in modelling knowledge. Moreover, none of these approaches address the need for diagnosing the learner during a task rather than at the end of it, and this is a key limitation that this thesis is addressing.

2.4 Summary

In this chapter, we have presented overviews of exploratory learning, user/learner modelling and learner modelling in exploratory learning environments. We have discussed the main characteristics of exploratory learning and the reasons why learner modelling was considered incompatible with this type of learning, and presented previous work in this area. In previous research, several works modelled knowledge (Veermans, 2003; Stathacopoulou et al., 2005; Andaloro and Bellomonte, 1998), one modelled skills related to exploratory learning (Ting and Phon-Amnuaisuk, 2009), one modelled meta-skills related to exploratory learning (Conati and Merten, 2007) and, finally, one modelled knowledge and effective exploration at the same time (Bunt, 2001; Bunt and Conati, 2003).

Our approach focuses on modelling knowledge rather than skills or meta-skills and was developed in the context of an exploratory learning environment for the domain of mathematical generalisation. This work is presented in the following chapter.

Chapter 3

Methodological Considerations

This research presented is this thesis was conducted using an Exploratory Learning Environment for mathematical generalisation developed in the context of the MiGen Project[1]. The development of the mechanisms presented in the following chapters was done along with the development of the system and this aspect had an influence on the methodology used for the development of the learner modelling techniques. One aspect that had a direct influence on our methodology was that the system was developed in an iterative manner; consequently, our techniques were developed iteratively as well.

Before presenting details of the methodology, a typical generalisation task is described in the following section to illustrate the aspects that need modeling.

3.1 Problem Description

In this thesis we address the user modelling problem in the context of exploratory learning, and more specificaly, in the context of a system developed by the MiGen project for the domain of mathematical generalisation. We describe here a typical problem for the domain of mathematical generalisation (for UK secondary school level) that will facilitate understanding of what is needed to develop a user modelling mechanism for exploratory learning of this domain.

Pond tiling is a mathematical generalisation problem for which the students are presented with a pond typically of rectangular form of a certain width (w)

[1]Funded by ESRC, UK, under TLRP e-Learning Phase-II (RES-139-25-0381); http://www.migen.org

and height (h) (see Figure 3.1) and are asked how many square tiles, of size 1, are needed to surround *any* pond.

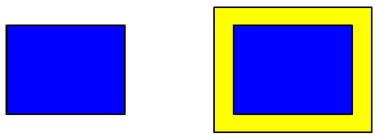


Figure 3.1: Pond tiling problem - pond and surrounded pond.

The algebraic solution for this problem is that the number of tiles needed to surround any rectangular pond is $2*w+2*h+4$. The MiGen system goal was to provide an environment in which such tasks can be solved that would facilitate the transition from a construction to an algebraic formula, thus providing an understanding of how algebraic rules are derived, i.e. of generalisation.

Thus, the system would present such tasks to the learners and would provide affordances that allow the learners to build constructions and express algebraic-rule. Therefore, a user modelling mechanism needs to enable diagnosis on both aspects. Moreover, the information stored in the user models should facilitate other pedagogical decisions such as feedback or collaborative activities.

The next section presents the aim and objectives of the thesis and the following section discussed iterative design, as this is an important aspect of the overall development of the system, as well as of the user modelling component. Section 3.4 presents the methodology for all strands of research in this thesis, while Section 3.5 gives a brief overview of the high-level approach taken for the main stand of research, i.e. user modelling.

3.2 Aims and Objectives

This PhD thesis focuses on learner modelling in Exploratory Learning Environments and aims to develop an approach that allows modelling of the learners' behaviour not only at the end of tasks, but also during the task. Moreover, we want to exploit the information in the learner model for educational purposes such as feedback prioritisation, group formation for collaborative activities and off-task detection.

To meet the above mentioned aims the following objectives were defined:

1. Identify relevant interactions and create suitable transformations for producing informative data that would be stored in the learner model.

2. Represent the evolution of learners' constructionist models.

3. Design and develop an updating mechanism for the learner model that would reflect both the current knowledge and the evolution of knowledge.

4. Exploit the information in the learner model for purposes such as personalised feedback prioritisation, grouping for collaboration and detection of off-task behaviour.

5. Maintain optimal coverage of the knowledge base.

3.3 Iterative Design

Iterative design refers to the practice of making continuous improvements in the design of a product on the basis of testing, expert evaluation, and consumer feedback ("Iterative Design", 2006). In the context of software products, the design is only one of the processes involved, along with others such as requirements gathering, implementation and testing. There are several software development models (Sommerville, 2001), some including iterative approaches (e.g. spiral model, iterative and incremental model, agile computing) and some not (e.g. waterfall model). As the development of the ELE for mathematical generalisation and of the learner modelling component followed the iterative and incremental model, a detailed outline of this model is given, while the other models are only briefly described.

The Waterfall model was the first explicit model of the software development process and its name is due to the sequential manner in which a phase follows from another. This model is criticised for its rigidity, as in practice a phase is rarely completed when moving to the next (Sommerville, 2001). In the Spiral model (Boehm, 1986), each loop represents a phase (e.g. system requirements, system design etc.) and risk analysis is performed in each phase. This model is used for large and complicated projects. Agile software development (Martin, 2002) combines teamwork, development and adaptability throughout the project. It values individuals and interactions over processes and tools, working software over comprehensive documentation, customer collaboration over contract negotiation and responding to change over following a plan.

The iterative and incremental development model refers to a cyclic development process in which the incremental development refers to a staging and scheduling strategy in which various parts of the system are developed at different times or rates, and integrated as they are completed, and the iterative development refers to a rework scheduling strategy in which time is set aside to revise and improve parts of the system (Cockburn, 2008). The software is developed incrementally, allowing to build on the knowledge from previous versions of the system - from both development and deployment. It starts with a simple implementation, i.e. prototype, that covers key aspects and requirements, that is then iteratively enhanced in subsequent versions until the full system is implemented. At each iteration, design modifications are made and new functional capabilities are added.

The procedure includes an initialisation step, an iteration step and the so-called Project Control List. The initialisation step involves a first version of the system with basic capabilities that underpin the key aspects of the problem that are easy to understand and implement. The main purpose of this basic version is to have something that users can react to and to learn from their responses. The project control list guides the iterative process and includes items such as new features to be implemented and aspects of the existing solution that need to be redesigned. This list is revised in each analysis phase. The iteration step involves the analysis of the current version of the system and the redesign and/or implementation of items from the project control list. The analysis of each iteration is based on user feedback and program analysis (e.g. structure, modularity, reliability) and the results inform modification in the project control list. The process is illustrated in Figure 3.2.

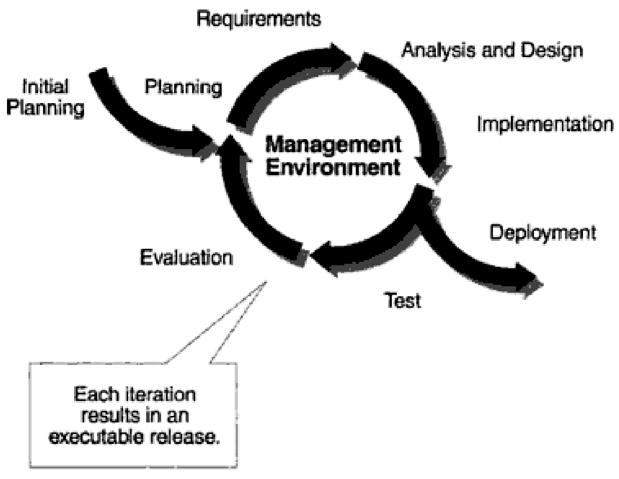


Figure 3.2: An iterative and incremental process - reproduced from Krutchen (2003)

There are several advantaged in using an iterative and incremental approach (Krutchen, 2003, p. 8):

1. Serious misunderstandings are made evident at the beginning of the lifecycle, allowing them to be dealt with more easily than if discovered at a later stage.

2. To elicit the system's real requirements, user feedback is enabled and

encouraged.

3. The development is focused on the most critical issues to the project and is shielded from other issues that may be a distraction from the project's real risks.

4. Continuous, iterative testing enables an objective assessment of the project's status.

5. Inconsistencies between requirements, design and implementation are detected early.

6. The workload of the team is spread more evenly throughout the project's life-cycle.

7. The team can leverage lessons learned and therefore can continuously improve the process.

8. Concrete evidence of the project's status can be given to stakeholders throughout its life-cycle.

Apart from software engineering approaches, a lot of research has emerged within the last 10-15 years in the area of learner-centred design, arguing the learners' involvement in the design of intelligent educational systems, especially when learners are children, as adults have a limited knowledge about how children make sense of software. Following is an overview of several approaches proposed in the area of iterative design with children for educational systems.

Some proposed approaches for learner-centred design focus on the design *product* (the educational system) (e.g. Soloway et al. (1994, 1996)), while others focus on the design *process* (e.g. Druin (2002); Scaife and Rogers (1998)).

The TILT model (tasks, interfaces, learner's needs, tools) (Soloway et al., 1994) was inspired from user-centred design that uses three of the aforementioned concepts, i.e. tasks, tools and interfaces, and adds a new concept that the authors argue as necessary for learner-centred design, i.e. learner's needs.

The Persistent Collaboration Methodology(PCM) (Conlon and Pain, 1996) focuses on the process of designing intelligent educational systems. Teachers, researchers and technologists are involved in a cycle of observation, reflection, design and action. This approach is considered by Good and Robertson (2006) to be more *school-centred* than *learner-centred* because learners were not part of the design team.

The term participatory design in which end users are involved and in which the users are children has been used by Druin (1999, 2002) who defined a methodology called cooperative inquiry. It involves a four-step process:

1. the contextual inquiry phase which involves collection of data in users' own environment;

2. the 'sticky note critiquing' phase during which children and adults critique an exiting piece of technology and, using sticky note pads record their likes, dislikes and a third category, e.g. surprises;

3. the participatory design phase in which the design team, including children, takes part in low-tech prototyping sessions;

4. the technology immersion phase which involves creating a space where children are able to access and use the existing technologies over a sustained period of time with researchers observing children's activity patterns in an unconstrained setting.

Several forms of involvement (Druin, 2002) are proposed to include children in the design of learning environments, which are given in a gradual scale. At the bottom of the scale the children's involvement is small as they act as *users* of technology. On the next steps, the children are more involved, acting as *testers* of prototype software and as *informants*, i.e. giving input in the design process. At the top of the scale, the children have the status of design *partners* acting as equal stakeholders throughout the design process.

Good and Robertson (2006) pointed out that the focus of cooperative inquiry is on children as *technology users*, while learner-centred design has a more constrained focus on children as *technology learners*, i.e. children who use the technology as a vehicle for learning.

The Informant Design Framework (Scaife and Rogers, 1998) considers several stakeholders including children, teachers, software designers and psychologists to contribute to the design of the interactive learning environment. It starts with specifying the learning goals and teaching practices for the domain and translate the specification initially into low-tech and later into high-tech designs. The expertise of the different stakeholders is used on specific aspects of the learning environment throughout the design process rather than having all the stakeholders working as an integrated team at all stages of the project.

The CARSS framework (Context, Activities, Roles, Stakeholders, Skills) (Good and Robertson, 2006) was specifically developed for participatory, learner-centred design with children. The context refers to the awareness of the broader context in which the design activity takes place. The activities describe the sequence of events that occur in the typical educational software design cycle. The roles describe the various functions that a member of the design team can fulfill, which each member possibly fulfilling more than one role. The stakeholders cover all the individuals who have a vested interest in the design process, and the skills refer to personal attributes and dispositions necessary to conduct successful design sessions. This framework can be applied to both intelligent and non-intelligent learning environments. It attempts to be fully inclusive and to be used for the design of interactive learning environments for children.

Another methodology entitled Identification-Development-Refinement (IDR) methodology (Winters and Mor, 2008) was proposed to address the issues related to interdisciplinary design. It aims to look at the full cycle of the design process and not just the software output and thus to include other outputs such as design patterns and pedagogical plans. Also, it focuses on engaging participants to reflect on their previous successful practices and to scaffold this reflection to generalizable solutions useful to the wider community. This methodology includes three stages: (a) the aim of the first stage is to identify potential patterns through the use of typologies and case studies; (b) the second stage looks at developing a set of patterns based on design evidence from the case studies; (c) the third stage aims to improve the patterns through collaborative discussion and reworking. The patterns are meant to *mediate* the interdisciplinary design process through their identification, development and refinement by the project participants.

The development of our modelling mechanism was directly influenced by the development of the exploratory learning system, as "user models cannot and should not be separated from the software systems that use them" (Chin, 2001, p. 183). Therefore, the development of the modelling mechanism needed to take an iterative approach, in line with the iterative development of the exploratory learning system. This, in turn, had an influence on the methodological approach, which is described in the following section.

3.4 Methodology

Our overall strategy for answering the research questions mentioned in Section 1.1 was a process including the following four stages: (a) knowledge elicitation; (b) identification of potentially suitable techniques; (c) experimentation with data based on real situations and (d) evaluation. There are some variations in the evaluations for the different techniques that will be pointed out below.

As mentioned in the previous section, our methodology was influenced by the iterative development of the system, leading to a similar iterative approach. The learning environment that was used is an exploratory learning environment in which the learners construct models and test them, which is a common feature of all ELEs. All the learning activities in this system included two parts: (a) the building of a construction (e.g. a rectangle, a T-shape) and (b) the development of an algebraic-like rule based on the construction (an example of such a learning activity was given above in Section 3.1; more details on this task are presented in Chapter 4, page 57).

As explained in Section 2.1, in exploratory learning the environment is more open-ended with no clear-cut right or wrong answers. Thus the interaction is more unstructured and an approach which allows flexibility in the development is needed, which had an influence on the choice of technique for learner modelling.

For the development of the learner modelling mechanism, the knowledge elicitation process involved collecting information about the tasks from educational experts and observing pupils using the system in small-scale studies that took place in schools or in the lab where the author was based. Case-Based Reasoning (CBR) was identified as a potentially good technique for modelling, as it offers the advantage of storing only relevant information and of being able to diagnose learners even if they have not completed a task. Moreover, CBR approaches are flexible and extendable, which was an important factor given the envisaged iterative development of the learner modelling mechanism. The CBR-based technique was tested using scenarios based on realistic data; the scenarios were identified with the help of educational experts.

Exploratory learning leads to situations when learners may need feedback on several aspects, therefore, raising the issue of what should feedback address if the learner asks for help. If an automatic feedback component would be integrated in the system, the problem of feedback prioritisation would also need an automatic mechanism, which was addressed in this thesis.

For the development of the feedback prioritisation mechanism, the knowledge was elicited from educational experts. One technique was identifies as being potentially useful for the problem of prioritisation, i.e. Analytic Hierarchy Process, a method from multi-criteria decision making. Multi-criteria decision making offers a methodology for making decision when several aspects need to be considered and several alternatives are possible. Moreover, it is also employed when expert knowledge is necessary. As feedback prioritisation is a complex decision problem influenced by many factors, with several possible alternatives and in need of pedagogical expertise, a considered that a multi-criteria decision approach was appropriate for addressing this issue. The evaluation of this mechanism was done by comparing the prioritisation delivered by the AHP technique with prioritisations provided by educational experts.

Group learning activities are more and more common and this is also the case for exploratory learning, where they play an important role. To enable formation of meaningful groups, a grouping mechanism that considers relevant criteria is needed. In this thesis, we addressed this issued following the methodology described below.

The knowledge elicitation process for grouping for collaborative learning activities involved working with pedagogical experts to identify the relevant criteria for grouping and the types of groups to be formed, e.g. homogeneous or heterogeneous. After eliciting this information, we have identified Group Technology as a potentially suitable technique to model our problem. Consequently, we have developed a mechanism based on Group Technology which outputs clusters of learners based on the approach they used to solve a particular task and on the similarity between the various approaches. Group Technology was chosen over clustering methods because of its capability of forming groups based on several criteria and of being able to modify the definition of criteria in the sense that a criterion could be defined in a range varying from very strict to very relaxed. This offers the advantage of being able to alter the grouping mechanism depending on the different collaborative activities settings and on the learner characteristics. The evaluation was done only at the data level (using real data from a classroom session), i.e. the output of the mechanism was checked against the desired output.

To ensure the user modelling mechanism produces relevant results, the system's knowledge needs a maintenance mechanism. For maintaining the optimal coverage of the knowledge base, information about the tasks was gathered from educational experts. Based on this information, a mechanism was developed that detects new relevant information from learners' actions and stores it in the knowledge base for future use. The mechanism uses the similarity of new information to the one already stored in the knowledge base and, when dissimilarities are observed, several checks are performed to ensure the new information is relevant. If the checks are successful, the new information is stored in the knowledge base. This adaptive modelling approach was chosen because detecting new relevant information is essential for the user modelling mechanism to provide accurate diagnosis over time. Also, as the relevant information is about tasks, the checks are related to characteristics of the tasks. The evaluation was done using real and simulated data.

For the learner modelling and feedback prioritisation mechanisms, an iterative approach was used, i.e. the mechanisms were developed for two versions of the system. For the learner modelling mechanism, there were other intermediate versions; however, these are not presented in the thesis. Therefore, these mechanisms were developed knowing that some modifications would be needed for the following versions of the systems. This influenced our choice of modelling techniques in the sense that we gave preference to flexible and extendable techniques. The general approach was to develop the initial version, to evaluate it and then move to the subsequent version of the system and made changes in the learner modelling and feedback prioritisation mechanisms in line with the the new features of the system. After the modifications were made, the mechanisms were evaluated again.

3.5 High-level CBR approach for user modelling

Based on information about mathematical generalisation tasks, a high-level approach for user modelling was developed based on case-based reasoning (CBR), which was chosen for the advantages outlined above (Section 3.4). The following chapter presents a review of CBR and how the high-level model presented below was refined in two iterative developments. Here we present only a very brief overview of CBR to explain the terminology in the high-level user modelling approach.

CBR stores information in form of cases. When a new case is detected (in our case a construction of a learner), it is compared with all stored cases to find the best match. Consequently, in our situation, we want to compare learners' constructions with some cases previously stored. These cases refer to solutions of the tasks - an example is given below.

We make the assumption that to build a construction learners need to identify a structure and work with several parts that put together constitute a solution (this is based on the way such tasks are typically solved with pen and paper). For example, in the pond tiling task described in Section 3.1, an example of possible components is displayed in Figure 3.3.

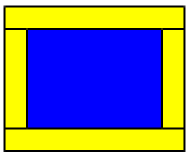


Figure 3.3: Example of a pond-tiling task structure

Thus, the example in Figure 3.3 has 5 parts: the pond, the horizontal top and bottom bars and the vertical bars on the left and right of the pond. To be able to compare the learners' constructions with solutions of the task, we need to store these solutions - both as part level, as well as a whole. Thus we are dealing with *simple cases* such as the pond or the vertical bar, and *composite cases* such as the example in Figure 3.3, i.e. the simple cases plus the relations between them that allow the formation of that particular structure.

Consequently, the following concepts were defined:

Definition 3.1 *A **case** is defined as*

$$C_i = \{F_i, RA_i, RC_i\},$$

where C_i represents a case, F_i is a set of attributes, RA_i is a set of relations between attributes and RC_i is a set of relations between cases.

The attributes include aspects such as the width and height. An example of a relation between attributes for the pond-tiling task is that the width of the horizontal bar depends on the width of the pond, and more precisely is defined as 'the width of the pond plus 2'. Therefore, there could be two types of relations - one that reflects the dependency and one that reflects the value. Relations between cases refer to the order in which the different parts were constructed - thus, two such relations can be defined: previous and next. The first case will have only a next relation, the last one will have only a previous relation and the intermediate ones will have both.

Definition 3.2 *A composite case or* ***strategy*** *is defined as*

$$S_u = \{N_u(C), N_u(RA), N_u(RC)\},$$

$u = \overline{1, r}$ *, where*
$N_u(C)$ *is a set of (simple) cases,*
$N_u(RA)$ *is a set of relation between attributes*
$N_u(RC)$ *is a set of relation between cases*

Thus, a composite case or strategy should have all the needed information for defining a particular structure such as the one displayed in Figure 3.3. Therefore, the modelling principle is to store solutions of tasks (i.e. strategies) in a knowledge base and compare the learners' constructions with the stored ones to identify what they are doing.

This high-level conceptual model was developed further by incorporating elements of the system, and especially of the interface which define the way students are to interact with the system. These details are presented in the following chapter.

3.6 Summary

This chapter discussed the methodological issues of this research in relation to the aims and objectives and gave an overview of iterative design. The overall methodological approach was also presented together with other methodological details that were specific to the different research objectives.

Chapter 4

Learner Modelling for Exploratory Learning of Mathematical Generalisation

This chapter presents two iterative versions of a learner modelling mechanism based on case-based reasoning developed in the context of an exploratory learning environment for the domain of mathematical generalisation. To facilitate the understanding of the difficulties of teaching and learning mathematical generalisation, of the challenges involved in designing a learner modelling component in the context of iterative development of other components of the system, and of the techniques used, overviews of mathematical generalisation, iterative design and case-based reasoning are presented. These overviews are followed by the two versions of the learner modelling mechanism, each including a description of the version of the system used, the knowledge representation, the similarity metrics and an evaluation. This is followed by a discussion of the results, a summary and an outline of the contribution of the chapter.

4.1 Mathematical Generalisation

Mathematical generalisation has been defined or described in several ways, varying from philosophical views that could be applied to any type of generalisation to views very specific to mathematics. Examples from the first category are:

- "an object and a means of thinking and communicating" (Dorfler, 1991, p. 63);
- "applying an argument in a broader context" (Harel and Tall, 1991, p. 38).

An example from the second category is: "Generalizing problems, also known as numeric sequences or geometric growing sequences, present patterns of growth in different contexts. Students are asked to find the underlying structure and express it as an explicit function or 'rule'." (Moss and Beatty, 2006, p. 442).

Mathematical generalisation is at the centre of algebraic expressions, as "algebra is, in one sense, the language of generalisation of quantity. It provides experience of, and a language for, expressing generality, manipulating generality, and reasoning about generality" (Mason, 2002, p. 105). This relation, however, together with the idea of recognising and analysing patterns and articulating structure, seems to be elusive to students who fail to understand algebra and its purpose (Geraniou et al., 2008). Students are unable to express a general pattern or relationship in natural language or in algebraic form (Hoyles and Küchemann, 2002).

Students, however, are able to identify and predict patterns (Mason, 2002) and there are claims that it is not the generalisation problems that are causing difficulties to students, but the way these are presented and the limitations of the teaching approaches used (Moss and Beatty, 2006). Typically, "generalising problems are usually presented as numeric or geometric sequences, and typically ask students to predict the number of elements in any position in the sequence and to articulate that as a rule" (Moss and Beatty, 2006, p. 443). A common strategy is "the construction of a table of values from which a closed-form formula is extracted and checked with one or two examples" (Bednarz et al., 1991, p. 7), introducing a tendency towards pattern spotting and emphasizing its numerical aspect (Noss et al., 1997; Noss and Hoyles, 1996). This approach obscures the variables involved, "which severely limits students ability to conceptualise the functional relationship between variables, explain and justify the rules that they find, and use the rules in a meaningful way for problem solving" (Moss and Beatty, 2006, p. 444).

Another approach that affects students' understanding of generalisation is the focus on mathematical products rather than mathematical processes (Warren and Cooper, 2008; Malara and Navarra, 2003). Malara and Navarra (2003) argue that students should be taught to distance themselves from the result and the operations needed to obtain that result, and to reach a higher level of thinking by focusing on the structure of a problem.

Another difficulty encountered in teaching mathematical generalisation is the students' difficulty to use letters that stand for the unknown (Küchemann,

1991) and to realise that letters represent values (Duke and Graham, 2007). Secondary school students also tend to lack a mathematical vocabulary for expressing generality (Geraniou et al., 2008) and research reports on students' lack of precision in written responses (Warren and Cooper, 2008).

Taking this aspects into account, a system was developed using an iterative process that involved designing with students and teachers. The main aim was to develop an environment that provides the students with the means for expressing generality rather than considering special cases or spotting pattern.

Several pedagogical requirements were derived from the literature on mathematical generalisation, of which a brief overview was given above. Although expressed differently in different iterative versions of the system, these requirements remained invariable (Geraniou et al., 2009; Noss et al., 2009):

1. Providing a rational for generality;
2. Supporting model construction and analysis simultaneously;
3. Scaffolding the route from numbers to variables;
4. Working on a specific case 'with an eye' on the general;
5. Supporting reflection on derived expressions.

The way these requirements were expressed for each iterative version is described when presenting the versions of the system in the sections following the overview of iterative design and case-based reasoning. The next section gives an overview of case-based reasoning (CBR) and its advantages, and includes a review of CBR applications in general and CBR application in the educational domain, in particular.

4.2 Case-Based Reasoning

In case-based reasoning (CBR) (Kolodner, 1993) the knowledge is stored as cases, typically including the description of a problem and the corresponding solution. When a new problem is encountered, similar cases are searched and the solution is adapted from one or more of the most similar cases. The CBR cycle (see Figure 4.1) typically included four processes (Aamodt and Plaza, 1994):

1. *Retrieve* cases that are similar to the current situation.
2. *Reuse* the cases and adapt them to solve the current situation.
3. *Revise* the proposed solution if necessary.
4. *Retain* the new solution as part of a new case.

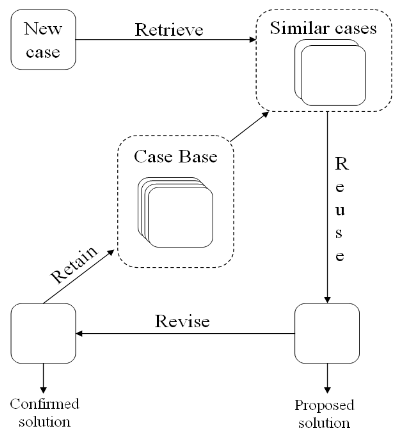


Figure 4.1: CBR cycle.

A number of advantages of CBR over other approaches have been pointed out (Kolodner, 1993; Watson, 1997); some of them could be particularly useful in ELEs:

- CBR is suitable for problem solving even with partial domain knowledge; this is particularly advantageous for domains suitable for exploration, which tend to be vast and often ill-structured;
- CBR allows shortcuts in reasoning; identifying a suitable case is usually accompanied by a solution that can be proposed; in ELEs, when a learner is getting far from a useful learning trajectory, an automatic intervention could occur 'on the spot'.
- CBR can keep a record of each situation that has occurred for future reference and could also learn from errors.

One may argue that a rule-based approach would be better than CBR. There are several reasons for not taking this route (Nickles, 1998):

(a) It is easier to elicit knowledge of concrete cases from experts; it is also easier to teach and learn this kind of knowledge; sources of information like textbooks and research papers are repositories of such cases.

(b) CBR offers a rational for its decisions; with rule-based reasoning (RBR) such a rational is less intelligible.

(c) CBR can be faster than RBR as "it relies heavily on past experience and does not constantly 'reinvent the wheel'." (Nickles, 1998, p. 72).

(d) CBR provides a sort of analysis and decomposition of a given situation by highlighting the differences between the current problem and the case-base, generating in this way a set of subproblems to be solved in a new round of CBR.

(e) While rules are produced by abstraction from contextual detail, CBR provides more "off the shelf" help for current problems, thus not depending on the relevant specific details of a particular situation.

(f) CB systems scale better than rule-based systems, as RBR needs completeness in the set of rule to be reliable, but CBR can be useful even with a very small case-base (virtually from the first case). In general, increasing the size of the knowledge base slows rule-based systems more than case-based systems.

Although CBR has been successfully used in applications for domains like legal reasoning (Aleven, 2003), stock market prediction (Chun and Park, 2005), recommender systems (Kumar et al., 2005), and other areas, there is little research on using CBR for e-Learning environments. For example, Han et al. (2005) use CBR in the learner modelling process and call this approach case-based student modelling, while Huang et al. (2007) use CBR and genetic algorithms to construct an optimal learning path for each learner. CBR is used also in Stottler and Ramachandran (1999) within a case-based instruction scenario rather than a method for learner modelling. We have not found any references in the literature to ELEs that use CBR or CBR combined with other intelligent methods.

The advantage of CBR for learning environments and especially for ELEs is that the system does not rely only on the general knowledge of a domain, but it can also use specific knowledge previously experienced (Han et al., 2005). It also seems promising for improving the effectiveness of complex and unstructured decision making (Huang et al., 2007), especially in combination with soft computing methods.

The following section presents the first iterative version of the exploratory learning environment for mathematical generalisation and the modelling framework that was developed for this version.

4.3 Iterative Design Version 1

This section presents the first version of the system and of the learner modelling mechanism together with the evaluation of the latter. The system is described outlining the principles mentioned in Section 4.1.

4.3.1 Version 1 of the Exploratory Environment: the *ShapeBuilder*

The first version of the software focuses on facilitating structured algebra thinking in children by allowing them to create and identify patterns and articulate structures in order to recognise, express and justify generality, a concept that lies at the centre of mathematical thinking. It is named *ShapeBuilder* and has been designed for classroom use and targets pupils of 11 to 14 year-olds. Each task involves two main phases: constructing a model and deriving an algebraic-like rule from it. For example, one such task in entitled 'pond-tiling' and requires from the learners to construct a rectangular pond, to surround it with tiles and find an algebraic rule that illustrates the relation between the number of tiles needed to surround the pond and the dimensions of the pond. To solve this task, the learners construct the pond and surround it by using the affordances of the system that are detailed below.

Figure 4.2 illustrated the interface of *ShapeBuilder* which includes an Expression Toolbar (a), a Shape List (b), and the Expression Palette (d).

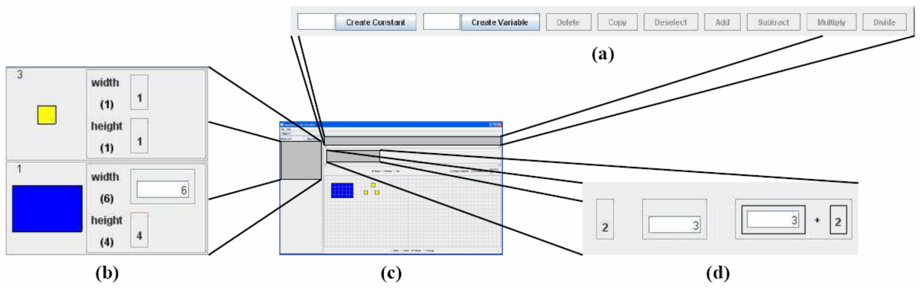


Figure 4.2: (a) the Expression Toolbar; (b) the ShapeList; (c) the overall *ShapeBuilder* interface (the gridded area is the interaction canvas); (d)the Expression Palette (image obtained from Dr. Darren Pearce and reproduced with permission).

ShapeBuilder allows construction of different shapes, e.g. rectangles, L-shapes, T-shapes, and supports *numeric*, *iconic* and *symbolic representations*. *Numeric representations* include numbers (constants or variables) and expressions with numbers; *iconic representations* correspond to icon variables; *symbolic representations* are names or symbols given by users to variables or expressions. An icon variable has the value of a dimension of a shape (e.g. width, height) and can be obtained by double-clicking on the corresponding edge of the shape. It is represented as an icon of the shape with the corresponding edge highlighted (see Figure 4.3a).

The Expression Toolbar allows the creation of constants, variables and composite expressions using addition, subtraction, multiplication and division. These

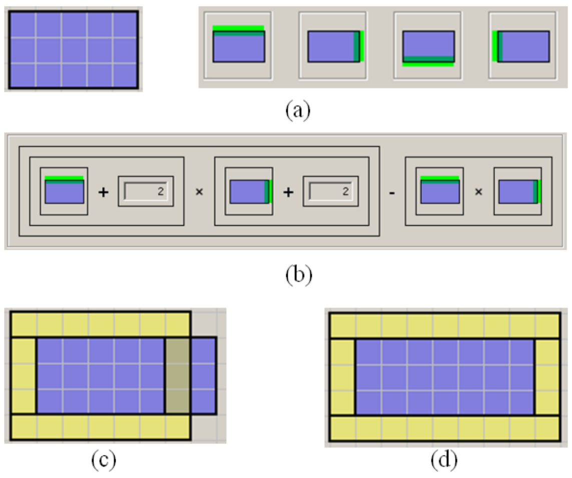


Figure 4.3: (a) A rectangular shape and its icon variable; (b) an expression using icon variables; (c) 'messing up'; (d) general solution that does not 'mess up'.

are placed in the Expression Palette and can be used for defining an expression for the task at hand or to define the properties of the shapes in the ShapeList. The ShapeList displays the shapes that currently exist on the gridded canvas and allows the creation of new shapes. The latter is done by dragging expressions from the Expression Palette for the specific shape properties, e.g. width and height for rectangles, thickness and size for a T-shape, etc. Once created, the thumbnail of a shape displayed in the ShapeList can be dragged on the canvas. If several copies of the shape are needed, they can be dragged in the same way on the canvas; the number above the thumbnail of the shape indicates the number of shapes existing on the canvas - for example, in Figure 4.2b the '3' above the yellow (lighter colour) thumbnail indicates there are 3 such shapes on the canvas (see Figure 4.2c), while the '1' above the blue (darker colour) thumbnail indicates there is only one such shape on the canvas. Existing shapes can be manipulated on the interaction canvas - they can be moved and attached to other shapes, and can be resized by either using the mouse or changing their properties in the ShapeList. When a shape has several copies and the properties of one of them is changed, all copies are updated appropriately.

The properties of shapes in the ShapeList facilitate the derivation of the algebraic-like expression for the task at hand by providing parts of the final expression which is formed by putting together various properties of the shapes used in the construction. Thus, *ShapeBuilder* supports *simultaneously model construction and analysis*.

Constants, variables and numeric expressions lead to specific constructions, while icon variables and expressions using icon variables lead to general constructions. Through the use of icon variables, *ShapeBuilder* encourages structured algebra thinking, connecting the visual with the abstract (algebraic) representation, as “each expression of generality expresses a way of seeing” (Mason, 2002, p. 116) (see Figure 4.3b). Thus, icon variables *scaffold the route from numbers to variables*, by providing a visual link between a value (i.e. number), an image and a name (i.e. variable).

An important concept for *ShapeBuilder* is the ‘messing up’ metaphor (Healy et al., 1994) that consists of asking the learner to resize a construction and observe the consequences; the model will ‘mess up’ only if it is not general (see Figures 4.3c and d). By challenging the learners to produce a construction that cannot be messed up, *a rational for generality* is provided. Moreover, the interaction affordances of *ShapeBuilder* encourage the learners to start with a specific construction and gradually make it general, and to test its generality using the ‘messing up’ metaphor. Therefore, the *working on a specific case with ‘an eye’ on the general* requirement is fulfilled. The learners are also encouraged to reflect on the correspondence between their construction and their expression, and to check the generality of their expression; thus, the system promotes *reflection on derived expressions*.

The following section presents the modelling framework that was developed to monitor the learners’ actions and identify aspects of pedagogical importance, including the corresponding knowledge representation and similarity metrics employed.

4.3.2 Learner Modelling Version 1

As it is neither possible nor beneficial to model everything that a learner may do using the system, we chose to focus on underlying various strategies that learners may follow when solving a task. An important characteristic of the domain we are working with is the fact that the tasks presented to the learners have several equally valid solutions and that each task covers one or more learning objectives of the mathematical generalisation domain. We started our approach for learner modelling by outlining a high level modelling process or framework that is displayed in Figure 4.4.

The Learner Model has three components: (a) a Short-Term Model (STM) where the recent actions of the learner are stored; these are pre-processed and compared to the information in the Task Model; (b) a Task Long-Term Model (Task LTM) that contains information about the tasks performed by the learner and which is updated with the information from the matching process; (c) a

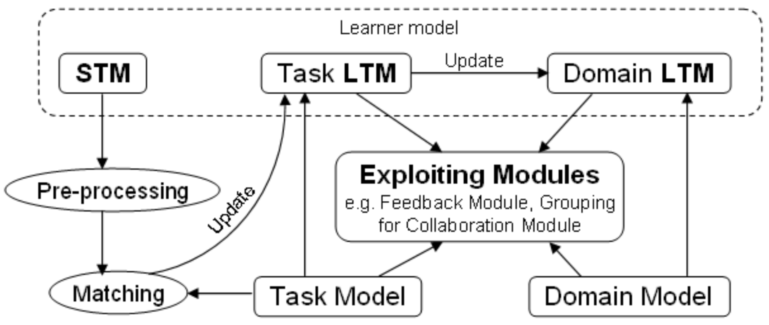


Figure 4.4: Learner modelling process

Domain Long-Term model (Domain LTM) which is an overlay model of the Domain Model. The information stored in the Learner Model is then used by the Exploiting Modules such as the Feedback Module and the Grouping for Collaboration Module. From this framework developed initially to guide our research, we focused on the Short-Term Model and the Task LTM and their usage by the Exploiting Modules.

The Task Model includes several tasks and each task includes different types of information:

(a) *strategies* for approaching the task;

(b) algebraic-like *rules* corresponding to each strategy;

(c) *landmarks*, i.e. relevant aspects or critical events occurring during the exploratory process; these are not task specific, but are registered for each task, end therefore, are included in this structure.

(d) *contexts* which refer to particular stages within a task; again, these are not task-specific, but are registered for each task.

The Task LTM includes the information mentioned above for each of the tasks performed by the learner.

Below we present how the proposed framework addresses the aims and objectives mentioned in Section 3.2:

(a) *Relevant interactions.* A representation of the relevant interactions of the learner with the system is required and several questions need to be addressed, e.g. identifying the relevant actions or sequences of actions. The proposed framework addresses this by storing the short-term actions of the learner and filtering out the relevant ones to be stored in the Task LTM. To identify the relevant actions in the context of *ShapeBuilder*, teachers' expertise and observations of children working with the ELE in the context of small-scale exploratory studies are used. This information was used in the development of the knowledge representation, which is presented below under the *Knowledge Representation* heading.

(b) *Representing the evolution of the learning process.* The proposed learner modelling process presented in Figure 4.4 addresses this issue. To this end, the structure of the learner model and the updating process follow the model of human memory often used in user modelling (e.g. Li et al. (2007); Anand and Mobasher (2007); Acquaviva et al. (2005)), and includes two components: a short-term model (STM) and a long-term model (LTM). The STM includes recent actions of the learner. The LTM contains information about the tasks and the domain and, thus, has two parts: the Task LTM that has the same structure as the task model, and the Domain LTM, which is an overlay model of the domain and maintains the knowledge of the learning outcomes associated with the learning process as inferred from the learner's constructions. In this thesis we did not work with the domain long-term model, and, thus, the Learner Model covered only the short-term model and the long-term task model.

(c) *Updating the model.* The learner model updating mechanisms are illustrated in Figure 4.4. During each task, the actions of the learner are stored in the STM and pre-processed. This process aims to transform the raw data into intermediate level data that will be used to identify (match) the strategies, landmarks and rules of a learner in the current task. Knowledge of the domain and teachers' expertise together with findings from pilot studies are used to derive these aspects for every task and define a 'light-weight' model for mathematical generalisation. Thus, the modelling process reflects the constructionist approach of incremental knowledge acquisition. The way in which the information to be stored in the long-term task model is identified is explained further on under the *Similarity Metrics* heading.

(d) *Exploitation of the learner model.* As illustrated in Figure 4.4, the information in the Learner Model feeds into Exploiting Modules. For example, the Feedback Module could use the information stored in the learner model to deliver personalised feedback. Details on how the exploitations happens at computational level are given in Section 4.4.3, and Chapters 5 and 6.

Knowledge Representation

For the reasons outlined in Section 4.2, Case-based Reasoning is used in the learner modelling process. The cases contain information describing constructions that learners build using *ShapeBuilder*. Different strategies in approaching a task, i.e. building a construction to meet a particular learning objective, are represented as a series of cases that reflect possible exploratory trajectories of learners as they build constructions during the various tasks.

Definition 4.1 *A* ***case*** *is defined as*

$$C_i = \{F_i, RA_i, RC_i\},$$

where C_i represents a case, F_i is a set of attributes, RA_i is a set of relations between attributes of cases and RC_i is a set of relations between cases.

Definition 4.2 *The* ***set of attributes*** *is represented as*

$$F_i = \{\alpha_{i_1}, \alpha_{i_2}, \ldots, \alpha_{i_N}\}.$$

It includes two types of attributes: (a) numeric and (b) variables. Variables refer to different string values that an attribute can take. Some numeric attributes are binary, indicating whether a case can be considered in formulating a particular strategy or not. This is be represented as a 'part of strategy' function: $PartOfS_u : C_i \to \{0, 1\}$,

$$PartOfS_u = \begin{cases} 1 & \text{if } C_i \in S_u \\ 0 & \text{if } C_i \notin S_u, \end{cases}$$

where S_u represents a strategy and is defined further on. The set of attributes of a generic case for *ShapeBuilder* is presented in Table 4.1. The first v attributes ($\alpha_{i_j}, j = \overline{1, v}$) are variables, the ones from $v + 1$ to w are numeric ($\alpha_{i_j}, j = \overline{v + 1, w}$) and the rest are binary ($\alpha_{i_j}, j = \overline{w + 1, N}$). As all cases share the same structure, the case base is *homogeneous* (Watson, 1997).

Definition 4.3 *The* ***set of relations between attributes*** *(see Table 4.2) of the current case with attributes of other cases (including the attributes of the current case) is represented as*

$$RA_i = \{RA_{i_1}, RA_{i_2}, \ldots, RA_{i_M}\},$$

where at least one of the attributes in each relation $RA_{i_m}, \forall m = \overline{1, M}$, is from F_i, the set of attributes of the current case C_i.

Two types of relations between attributes are used: *dependency relations* and *value relations.*

Definition 4.4 *A* ***dependency relation*** *(D_{i_s}) is defined as*

$$(\alpha_{i_k}, \alpha_{j_l}) \in D_{i_s} \Leftrightarrow \alpha_{i_k} = DEP(\alpha_{j_l}),$$

Table 4.1: The set of attributes (F_i) of a case.

Category	Name	Label	Possible Values
Shape	Shape type	α_{i_1}	Rectangle(/L-Shape/T-Shape)
Dimensions of shape	Width type	α_{i_2}	constant (c)/variable (v)/ icon variable (iv)/ numeric expression (n_exp)/ expression with iv(s) (iv_exp)
	Height type	α_{i_3}	c /v /iv /n_exp /iv_exp
	$\vdots$	$\vdots$	$\vdots$
	Thickness type	α_{i_v}	c /v /iv /n_exp /iv_exp
	Width value	$\alpha_{i_{v+1}}$	numeric value
	Height value	$\alpha_{i_{v+2}}$	numeric value
	$\vdots$	$\vdots$	$\vdots$
	Thickness value	α_{i_w}	c /v /iv /n_exp /iv_exp
Part of Strategy	$PartOfS_1$	$\alpha_{i_{w+1}}$	1
	$PartOfS_2$	$\alpha_{i_{w+2}}$	0
	$\vdots$	$\vdots$	$\vdots$
	$PartOfS_r$	α_{i_N}	0

where $DEP : \alpha_{i_k} \rightarrow \alpha_{j_l}$ for attributes α_{i_k} and α_{j_l} that are variables of cases i and j (where $i = j$ or $i \neq j$), and means that α_{i_k} depends on α_{j_l} (if $i = j$, $k \neq l$ is a condition as to avoid circular dependencies) (e.g. the width type of a case is depends on the height type of the same case; the width type of a case depends on the width type of another case, etc.);

Definition 4.5 *A* ***value relation*** *(V_{i_s}) is defined as*

$$(\alpha_{i_k}, \alpha_{j_l}) \in V_{i_s} \Leftrightarrow \alpha_{i_k} = f(\alpha_{j_l}),$$

where α_{i_k} and α_{j_l} are numeric attributes and f is a function and could have different forms depending on context (e.g. the height of a shape is two times its width; the width of a shape is three times the height of another shape, etc.).

These relations can occur between the current case and itself or between the current case and other cases (the index j does not have to be the same in each relation, though that is one of the possible situations).

Definition 4.6 *The* ***set of relations between cases*** *is represented as*

$$RC_i = \{RC_{i_1}, RC_{i_2}, \ldots, RC_{i_P}\},$$

where one of the cases in each relation $RC_{i_p}, \forall p = \overline{1, P}$ is the current case (C_i).

Table 4.2: The set of relations between attributes (RA_i) of cases.

Relation	Label	Example
Dependency relation	$D_{i_1}\ (RA_{i_1})$	$(\alpha_{i_k}, \alpha_{j_l})\,; k, l = \overline{2, v}; \forall j$
	$\vdots$	$\vdots$
	$D_{i_t}\ (RA_{i_t})$	$(\alpha_{i_k}, \alpha_{j_l})\,; k, l = \overline{2, v}; \forall j$
Value relation	$V_{i_1}\ (RA_{i_{t+1}})$	$(\alpha_{i_k}, \alpha_{j_l})\,; k, l = \overline{v+1, w}; \forall j$
	$\vdots$	$\vdots$
	$V_{i_z}\ (RA_{i_M})$	$(\alpha_{i_k}, \alpha_{j_l})\,; k, l = \overline{v+1, w}; \forall j$

Two relations about order in time are defined: $Prev$ and $Next$ relations.

Definition 4.7 *A $Prev$ relation indicates the previous case with respect to the current case:*

$$(C_i, C_j) \in Prev \ \textit{if}\ t\,(C_j) < t\,(C_i)$$

Definition 4.8 *A $Next$ relation indicates the next case with respect to the current case:*

$$(C_i, C_k) \in Next \ \textit{if}\ t\,(C_i) < t\,(C_k)\,.$$

Each case includes at most one of each of these two relations ($p \leq 2$).

Definition 4.9 *A **strategy** is defined as*

$$S_u = \{N_u(C), N_u(RA), N_u(RC)\}\,,$$

$u = \overline{1, r}$ *, where*
$N_u(C)$ *is a set of cases,*
$N_u(RA)$ *is a set of relation between attributes of cases and*
$N_u(RC)$ *is a set of relations between cases.*

There are two types of strategies depending on the degree of generality: specific and general. Specific strategies include value relations, but no dependency relations. The general strategies can be distinguished by the presence of the dependency relations and by the fact that the dimension type of at least one of the dimensions of the case is an icon variable or an expression using icon variable(s). The presence or absence of the above mentioned aspects apply to all cases that form the composite case with the exception of the first case (which is independent).

To illustrate our approach we use a mathematical generalisation task called 'pond tiling', which is common in the English secondary school curriculum and expects learners to produce a general expression for finding out how many tiles are required for surrounding any rectangular pond. In the context of *ShapeBuilder*, the task involves building a construction of the pond, surrounding it and deriving an algebraic-like expression from the construction.

An overview of the Task Model for the 'pond tiling' task is given below. This information has been obtained from small-scale studies with pupils and contains:

1. *Strategies* - these are correct or desirable ways of building a construction and are displayed in Figure 4.5;
2. *Rules* or expressions corresponding to the strategies - for example, the rule for the 'Area' strategy is $(w + 2) * (h + 2) - w * h$, where w and h stand for the width and the height of the pond, respectively.
3. *Landmarks* - these are events of pedagogical importance that happen during the exploratory process. Two such events were identified and they correspond to inefficient approaches: (i) building a construction using individual tiles and placing them one-by-one on the canvas - we refer to these as one-by-one constructions; two examples are displayed in Figure 4.6a and

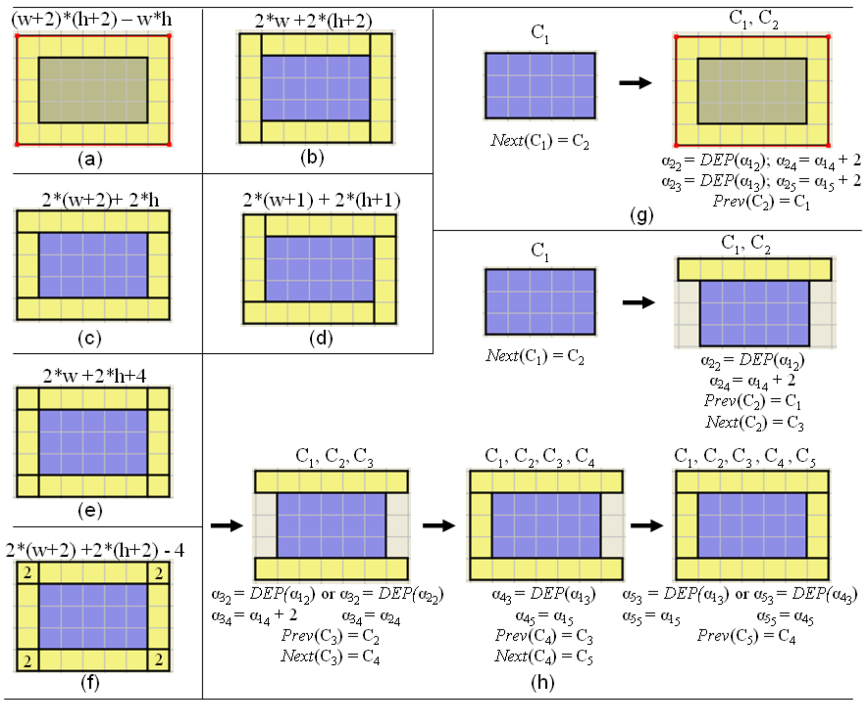

(w+2)*(h+2) – w*h
(a)
2*w +2*(h+2)
(b)
2*(w+2)+ 2*h
(c)
2*(w+1) + 2*(h+1)
(d)
2*w +2*h+4
(e)
2*(w+2) +2*(h+2) - 4
2
2
2
2
(f)
C_1
$Next(C_1) = C_2$
C_1, C_2
$\alpha_{2_2} = DEP(\alpha_{1_2})$; $\alpha_{2_4} = \alpha_{1_4} + 2$
$\alpha_{2_3} = DEP(\alpha_{1_3})$; $\alpha_{2_5} = \alpha_{1_5} + 2$
$Prev(C_2) = C_1$
(g)
C_1
$Next(C_1) = C_2$
C_1, C_2
$\alpha_{2_2} = DEP(\alpha_{1_2})$
$\alpha_{2_4} = \alpha_{1_4} + 2$
$Prev(C_2) = C_1$
$Next(C_2) = C_3$
C_1, C_2, C_3
$\alpha_{3_2} = DEP(\alpha_{1_2})$ or $\alpha_{3_2} = DEP(\alpha_{2_2})$
$\alpha_{3_4} = \alpha_{1_4} + 2$ $\alpha_{3_4} = \alpha_{2_4}$
$Prev(C_3) = C_2$
$Next(C_3) = C_4$
C_1, C_2, C_3, C_4
$\alpha_{4_3} = DEP(\alpha_{1_3})$
$\alpha_{4_5} = \alpha_{1_5}$
$Prev(C_4) = C_3$
$Next(C_4) = C_5$
C_1, C_2, C_3, C_4, C_5
$\alpha_{5_3} = DEP(\alpha_{1_3})$ or $\alpha_{5_3} = DEP(\alpha_{4_3})$
$\alpha_{5_5} = \alpha_{1_5}$ $\alpha_{5_5} = \alpha_{4_5}$
$Prev(C_5) = C_4$
(h)

Figure 4.5: (a) 'Area' strategy; (b) 'I' strategy; (c) 'H' strategy; (d) 'Spiral' strategy; (e) '+4' strategy; (f) '−4' strategy; (g) Steps and relations of 'Area' strategy; (h) Steps and relations of 'I' strategy.

b, and (ii) building a construction using 'bits and pieces', meaning that the construction is 'patched up' rather than constructed with a structure in mind - an example is given is Figure 4.6c. These ways of building a construction are considered inefficient because they cannot be generalised.

4. *Contexts* - two distinct contexts within a task were identified: (i) specific, i.e. the learners are working with a specific construction and (ii) general, i.e. the learners are working with a partially general or general construction.

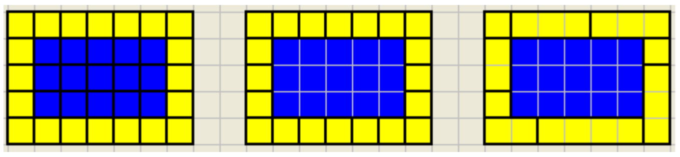


Figure 4.6: Landmarks: (a) one-by-one construction, including the pond; (b) one-by-one construction, excluding the pond; (c) 'bits and pieces' construction.

From studies with pupils several desirable strategies were identified. These strategies and their associated solutions (the general expressions for surrounding any rectangular pond) are displayed in Figure 4.5(a to f). Two strategies are presented in detail: the 'Area' strategy (S_1) and the 'I' strategy (S_3). The attributes of cases that are part of these two strategies are presented in Table 4.3 and Table 4.4, respectively. The steps for building these strategies, including the sets of relations between attributes and between cases for each step are displayed in Figure 4.5g and Figure 4.5h, respectively. The two presented strategies are *general*; their *specific* correspondent strategies would not have dependency relations and the types of the dimensions of each case would not include any icon variable (iv) or expression using icon variable (exp_iv).

Table 4.3: The set of attributes (F_i) for the cases in the 'Area' strategy.

Name	Label	C_1	C_2
Shape type	α_{i_1}	Rectangle	Rectangle
Width type	α_{i_2}	c/v/n_exp	iv/iv_exp
Height type	α_{i_3}	c/v/n_exp	iv/iv_exp
Width value	α_{i_4}	5	7
Height value	α_{i_5}	3	5
$PartOfS_1$	α_{i_6}	1	1
⋮	⋮	⋮	⋮
$PartOfS_2$	α_{i_7}	1	0
$PartOfS_6$	α_{i_8}	1	0

Table 4.4: The set of attributes (F_i) for the cases in the 'I' strategy.

Label	C_1	C_2	C_3	C_4	C_5
α_{i_1}	Rectangle	Rectangle	Rectangle	Rectangle	Rectangle
α_{i_2}	c/v/n_exp	iv /iv_exp	iv /iv_exp	c/v/n_exp	c/v/n_exp
α_{i_3}	c/v/n_exp	c/v/n_exp	c/v/n_exp	iv /iv_exp	iv /iv_exp
α_{i_4}	5	7	7	1	1
α_{i_5}	3	1	1	3	3
α_{i_6}	1	0	0	0	0
α_{i_7}	1	1	1	1	1
$\vdots$	$\vdots$	$\vdots$	$\vdots$		
α_{i_8}	1	0	0	1	1

Similarity Metrics

In Case-based Reasoning, the role of the similarity metrics is to measure how close the input case is to the stored ones and to retrieve one or several similar cases from the case-base. The most common definition of similarity is a weighted sum of similarities of attributes of cases (Kolodner, 1993):

$$S_{IR} = \frac{\sum_{i=1}^{N} o_i \times sim(f_i^I, f_i^R)}{\sum_{i=1}^{N} o_i},$$

where o_i represents the weight of each attribute, sim is a similarity function, and I and R stand for input and retrieved cases, respectively. For our purpose, four similarity measures are defined for comparing cases:

1. Euclidean distance is used for comparing numeric attributes: $D_{IR} = \sqrt{\sum_{j=v+1}^{w} (\alpha_{I_j} - \alpha_{R_j})^2}$

2. The following metric is used for variables: $V_{IR} = \frac{\sum_{j=1}^{v} g(\alpha_{I_j}, \alpha_{R_j})}{v}$, where g is defined as:
$$g(\alpha_{I_j}, \alpha_{R_j}) = \begin{cases} 1 & \text{if } \alpha_{I_j} = \alpha_{R_j} \\ 0 & \text{if } \alpha_{I_j} \neq \alpha_{R_j}, \end{cases}$$

3. Jaccard's index is used for comparing relations between attributes: $A_{IR} = \frac{|RA_I \cap RA_R|}{|RA_I \cup RA_R|}$. A_{IR} is the number of relations between attributes that the input and retrieved case have in common divided by the the total number of relations between attributes of the two cases.

4. Similarly, for relations between cases a Jaccard index is used: $B_{IR} = \frac{|RC_I \cap RC_R|}{|RC_I \cup RC_R|}$, where B_{IR} is the number of relations between cases that the input and retrieved case have in common divided by the the total number of relations between cases of I and R.

In order to identify the closest strategy to the one employed by a learner, cumulative similarity measures are used for each of the four types of similarity:

1. Numeric attributes - as this metric has a reversed meaning compared to the other ones, i.e. a smaller number means a greater similarity, the following function is used to bring it to the same meaning as the other three similarity measures, i.e. a greater number means greater similarity.

$$F_1 = \begin{cases} \frac{z}{\sum_{i=1}^{z} D_{I_i R_i}} & \text{if } \sum_{i=1}^{z} D_{I_i R_i} \neq 0 \\ z & \text{if } \sum_{i=1}^{z} D_{I_i R_i} = 0, \end{cases}$$

2. Variables: $F_2 = (\sum_{i=1}^{z} V_{I_i R_i})/z$.

3. Relations between attributes: $F_3 = (\sum_{i=1}^{z} P_{I_i R_i})/y$.

4. Relations between cases. $F_4 = (\sum_{i=1}^{z} T_{I_i R_i})/z$.

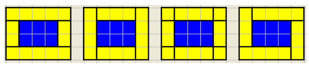
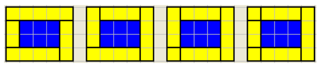
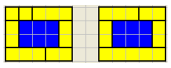
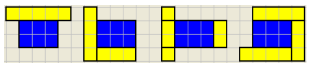
where z represents the minimum number of cases of the two strategies that are compared and y represents the number of cases from the retrieved strategy that have relations between attributes. The strength of similarity between the current strategy and the various stored strategies is defined as the combined similarity of these four measures: $Sim = F_1 + F_2 + F_3 + F_4$.

4.3.3 Evaluation

To evaluate the proposed mechanism for identification of strategies, we have identified several scenarios of pedagogical importance and performed an evaluation for each scenario. Scenario-based techniques (Carroll and Rosson, 1992; Carroll, 2000) are used in the design and evaluation of systems or systems components due to their ability to provide stakeholders with valid units on which to anchor their analysis because scenarios refer to instances of system use that can extend across space, time, people and system features (Haynes et al., 2004).

The scenarios were identified based on different types of user behaviour observed in small scale trials combined with pedagogical information about the importance of identifying certain behaviours. These scenarios, corresponding to categories of user strategies are given in the first column of Table 4.5, example user constructions that belong to each scenario are shown in the second column, whilst the third column provides the pedagogical rational for monitoring the particular strategy category, e.g. providing appropriate scaffoldings for users that demonstrate a particular behaviour. For example, the constructions in the first row of Table 4.5 represent complete strategies, commonly encountered in trials (Gutiérrez et al., 2008): 'I' strategy, 'H' strategy, '+4' strategy and 'Spiral' strategy (also displayed previously in Figure 4.5).

Table 4.5: Scenarios corresponding to user behaviours observed in small scale trials.

Scenario	Example Constructions	Pedagogical Rational
Complete strategies		Identifying if the learner is working with the specific or with the general.
Mixed strategies		Identifying used strategies to guide learners towards a particular one should they have difficulties to generalise.
Non-systematic strategies		Guiding the user toward a symmetric strategy should they have difficulties to generalise
Partial Strategies		Guiding the learners by building on the strategy they started with should they be stuck or ask for help.

The first category, i.e. **complete strategies**, is potentially the most important for the purpose of detecting whether users demonstrate learning behaviours that could lead to generalisation. In *ShapeBuilder* learners can build constructions either in a *specific* way or in a *general* way. A *completely general construction* is characterised by relations between all its variable parts, e.g. for the 'I' strategy, the rows of tiles at the top and bottom need to be linked to the width of the pond (this indicates that the learner established a type of *dependency relation* between the variables) and have their width equal to the width of the pond plus 2 (an indication that the user has created a *value relation* between variables); similarly, the left and right columns of tiles should be linked to the height of the pond (an indication that the user identified another *dependency relation*) and their height should be equal to the height of the pond (i.e. a *value relation*). For the '+4' strategy, the top, bottom, left and right parts are variable and should be linked to the width and height of the pond respectively, but the four corners are not variable and need not be linked to the pond. If none of the variable parts of the construction are linked, the construction is *specific* - the value relation is still present but there are no dependency relations. If some variable parts of the construction are linked, while others are not, the construction is *partially general*.

Modelling users' behaviour expressed through strategies adopted when building a construction, i.e. specific, partially general or completely general, could be exploited pedagogically. Especially at the very beginning of the exploratory learning, i.e. when the learners are in a "novice state", it was observed that learners first build constructions that are specific and only when completing the specific construction attempt to make it general. Moreover, in practice, after building a specific construction pupils find it quite challenging to create the first link between the components of that construction (i.e. a dependency relation).

When learners are working with a symmetric, 'elegant' construction the process of generalisation becomes easier because the dependency and value relations are the same for several components of the strategy (see the complete strategies in Table 4.5) and, therefore, the transition to an algebraic-like rule is facilitated. However, learners use a variety of strategies when building their constructions, i.e. they adopt strategies of the second category, the so-called **mixed strategies**. In that case producing a construction that can be generalised may become more difficult because of the added complexity of having, for example, four different expressions in a rule instead of two. Trials with pupils using *ShapeBuilder* showed that although some learners can still generalise from such a structure, most learners faced additional difficulties. It was also observed that the tendency to adopt mixed strategies might relate to the so-called "novice" approach, as pupils with some experience appear to take advantage of the symmetry in their construction to generalise. The examples provided in Table 4.5 are, from left to right: 'I' and 'Spiral' strategies, 'H' and 'Spiral' strategies, 'H', '+4' and 'Spiral' strategies, and '+4' and 'H' strategy.

Non-systematic strategies are constructions that are partly made of 'bits and pieces', even when the overall construction indicates that a symmetric strategy has been partly followed - like in the second example displayed in Table 4.5 for this category, where the user's strategy resembles the 'I' strategy, but small bars of tiles were also used to add up to the desired number of tiles (in the particular example, the pupil used two bars of three and two tiles, respectively). Incorporating 'bits and pieces' in a construction makes it difficult to explicitly identify the strategy the learner has followed - as in the first example, where the learner could be working with the 'H' or the '+4' strategy. Nevertheless, trying to surround a particular pond using 'bits and pieces' of tiles, without thinking whether the solution would work for a pond of any dimension, manifests a particular learner behaviour that is easily recognisable by tutors and that could also be automatically detected, as they are 'symptoms' of a particular problem. This was previously introduced as a *landmark*. In the case of constructions made of 'bits and pieces', the problem is that the learner is not thinking generally and only trying to reproduce a construction without thinking about its structure. Unlike construction entirely made of 'bits and pieces', non-systematic strategies have some structure in place, as well as some parts that are made of 'bits and pieces'.

Identifying **partial strategies**, i.e. the fourth scenario, is important for situations when the learner is stuck or requests help. To make the intervention beneficial, identifying the strategy they started to work with is valuable. The examples shown for this scenario go from very little information (only on row of tiles in the first example) to almost a complete strategy (the last example).

The first three scenarios are more focused on generalisation and how to address difficulties that learners face with this domain, while the fourth one is more about helping the learner to get to one of the stages described in the first three scenarios, as some learners get stuck at an early stage.

In the following we present outputs of our mechanism for each scenario, using constructions from classroom trials with pupils. In this evaluation we used data from 10 pupils, where each pupil built one construction. The mechanism used the input from log files and its output (i.e. most similar strategy or strategies) was checked by the author against screen videos that were collected for all pupils.

Complete strategies. We present three situations: (a) a complete strategy that is specific; (b) a complete strategy that is general and (c) a complete strategy that is partially general. For the first two, we use the 'H' strategy displayed in Figure 4.5c and compare it with all stored strategies; the results are displayed in Table 4.6. For the third situation, we use the '-4' strategy with 2 general cases and 2 specific cases - see Figure 4.7; the similarities between this strategy and all stored strategies are displayed in Table 4.7.

Table 4.6: Similarity metrics for the 'H' strategy.

Stored Strategies	**'H' strategy specific**	**'H' strategy general**
Area	1.87	2.37
I	2.17	2.97
H	**7.20**	**8.00**
Spiral	2.37	3.17
+4	2.89	3.69
-4	2.89	3.69

The results show that the 'H' strategy is correctly identified as being the most similar from the stored strategies, for both the specific and the general constructions. All strategies are stored in their general form, i.e. they have all the required dependency relations.

The results in Table 4.7 show that the '-4' strategy is correctly identified as the most similar strategy to the partially general strategy in Figure 4.7.

Mixed Strategies. Two examples are displayed in Figure 4.8. The similarity metrics of these two constructions when compared with all stored strategies are

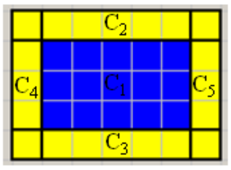


Figure 4.7: '-4' strategy combining specific and general cases: C_2 and C_3 are general; C_4 and C_5 are specific.

Table 4.7: Similarity metrics for a partially general '-4' strategy.

Stored Strategies	**Partially general '-4' strategy**
Area	2.72
I	3.09
H	3.00
Spiral	2.49
+4	2.39
-4	**7.35**

given in Table 4.8.

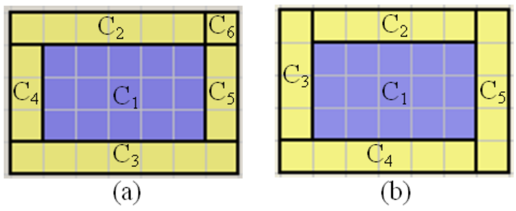


Figure 4.8: Mixed strategies: (a) combination of 'I', '+4' and 'Spiral' strategies; (b) combination of 'Spiral' and 'H' strategies.

Table 4.8: Similarity metrics for the mixed strategies in Figure 4.8.

Stored Strategies	**Mixed strategy a**	**Mixed strategy b**
Area	1.71	1.67
I	**4.46**	1.75
H	1.70	**2.79**
Spiral	**2.20**	**2.79**
+4	**2.79**	1.98
-4	2.03	2.14

The first example (Figure 4.8a), has 4 cases in common with two strategies: the 'I' strategy (C_1, C_3, C_4, C_5) and the '+4' strategy (C_1, C_4, C_5, C_6). Moreover, it also has two case in common with the 'Spiral' strategy (C_1, C_2). The results in Table 4.8 show that these three strategies are the most similar ones, with the 'I' strategy being most similar, followed by the '+4' strategy and the 'Spiral' strategy. All the other stored strategies have lower similarities.

The second example (Figure 4.8b), has 3 cases in common with two strategies: the 'Spiral' strategy (C_1, C_3, C_4) and the 'H' strategy (C_1, C_2, C_5). The results in Table 4.8 show that these two strategies are the most similar ones. They are also equally similar, due to the symmetry of the construction. As in the previous example, all the other stored strategies have lower similarities. Therefore, when mixed strategies are used, the most similar strategies are correctly identified.

Non-systematic strategies. Two examples are displayed in Figure 4.9 and their similarities when compared with all stored strategies are given in Table 4.9.

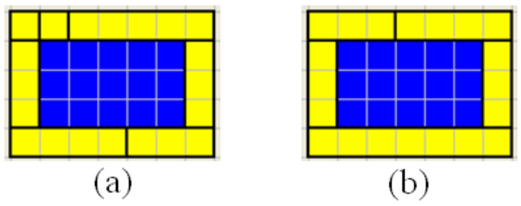


Figure 4.9: Non-systematic strategies: (a) combination of 'I' and '+4' strategies with 'bits and pieces'; (b) 'I' strategy with 'bits and pieces'.

Table 4.9: Similarity metrics for the non-systematic strategies in Figure 4.9.

Stored Strategies	**Strategy a**	**Strategy b**
Area	1.69	1.64
I	**2.28**	**2.68**
H	1.69	1.63
Spiral	1.75	1.71
+4	**3.17**	**2.57**
-4	1.57	1.75

For the construction in Figure 4.9a, which is a combination of 'I' and '+4' strategies that includes 'bits and pieces', the results in Table 4.9 show the '+4' strategy is most similar to this construction (having four cases that correspond to this construction), followed by the 'I' strategy (which has 3 cases that correspond to this construction). For the construction in Figure 4.9b, which is an 'I' strategy with 'bits and pieces', the results in Table 4.9 show that the 'I' strategy is identified as the most similar strategy.

Partial strategies. Several examples of partial strategies are given in Figures 4.10, 4.11 and 4.12, varying from incipient construction with only one more case besides the pond to almost complete constructions (i.e. only one case missing). Similarly to the previous scenarios, these examples are based on observations of learners' behaviour in classroom context. Using logs of their actions, snapshots of their constructions were taken at different points during the task to form the partial constructions. The similarities of these constructions with all stored strategies are given in Tables 4.10, 4.11 and 4.12. Depending on how advanced the construction is and on the strategy used, some strategies can be identified from an early stage while others need to be more advanced for the identification mechanism to output one most similar strategy. This is discussed below.

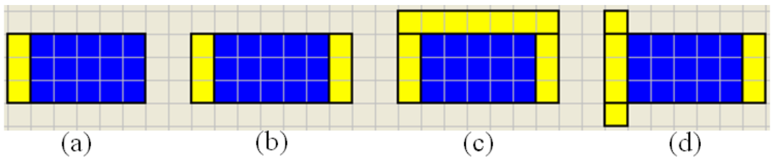


Figure 4.10: Partial strategies: (a) 'I' or '+ 4' partial strategy (2 cases); (b) 'I' or '+ 4' partial strategy (3 cases); (c) partial 'I' strategy; (d) partial '+4' strategy.

Table 4.10: Similarity metrics for the partial strategies in Figure 4.10.

Stored Strategies	**Strategy a**	**Strategy b**	**Strategy c**	**Strategy d**
Area	1.55	1.55	1.72	1.55
I	**3.75**	**4.67**	**5.63**	1.86
H	2.01	1.73	1.63	1.57
Spiral	2.25	1.92	1.82	1.55
+4	**3.75**	**4.67**	2.97	**7.00**
-4	2.01	1.73	2.05	1.48

The constructions in Figure 4.10a and b could belong to the 'I' or the '+4' strategy and this is reflected in the results displayed in Table 4.10 by the fact that the similarity to these two strategies is the same with a value of 3.75 and 4.67, respectively. On the other hand, the constructions in Figure 4.10c and d are clearly most similar to one of the two strategies: the construction in Figure 4.10c is most similar to the 'I' strategy, while the construction in Figure 4.10d is most similar to '+4' strategy. This is accurately reflected in the results given by the identification mechanism, which are displayed in Table 4.10. Therefore, for the 'I' and '+4' strategies that have 2 cases (except the pond) in common, a strategy containing these common cases can be distinguished as most similar to one of the two when at least one case that clearly belongs only to one of the two strategies is also present.

Similarly to the previous examples, the constructions in Figure 4.11a and b

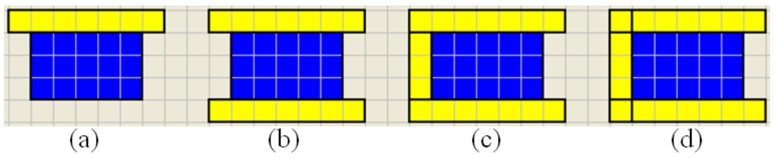


Figure 4.11: Partial strategies: (a) ‘I’ or ‘-4’ partial strategy (2 cases); (b) ‘I’ or ‘-4’ partial strategy (3 cases); (c) partial ‘I’ strategy; (d) partial ‘-4’ strategy.

Table 4.11: Similarity metrics for the partial strategies in Figure 4.11.

Stored Strategies	**Strategy a**	**Strategy b**	**Strategy c**	**Strategy d**
Area	1.72	1.72	1.72	.72
I	**3.75**	**4.67**	**5.63**	2.97
H	2.01	1.73	1.63	2.05
Spiral	2.40	2.03	1.86	1.89
+4	2.01	1.73	2.05	1.63
-4	**3.75**	**4.67**	2.97	**5.63**

could belong to either the ‘I’ strategy or the ‘-4’ strategy, while the constructions in Figure 4.11c and d clearly belong to the ‘I’ strategy and to the ‘-4’ strategy, respectively. The results displayed in Table 4.11 accurately reflect this: both the ‘I’ and the ‘-4’ strategies are equally similar to the constructions in Figure 4.11a and b, while the construction in Figure 4.11c is most similar to the ‘I’ strategy and the construction in Figure 4.11d is most similar to the ‘-4’ strategy.

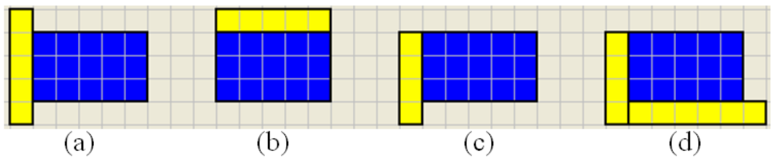


Figure 4.12: Partial strategies: (a) ‘H’ or ‘-4’ partial strategy; (b) ‘H’ or ‘+4’ partial strategy; (c) partial ‘Spiral’ strategy (2 cases); (d) partial ‘Spiral’ strategy (3 cases).

Table 4.12: Similarity metrics for the partial strategies in Figure 4.12.

Stored Strategies	**Strategy a**	**Strategy b**	**Strategy c**	**Strategy d**
Area	1.57	1.67	1.57	1.57
I	2.01	2.01	2.25	1.97
H	**3.75**	**3.75**	2.40	1.87
Spiral	2.40	2.25	**3.75**	**4.67**
+4	2.01	**3.75**	2.25	1.92
-4	**3.75**	2.01	2.40	2.03

The examples in Figure 4.12 cover one construction that could belong to either the ‘H’ or the ‘-4’ strategy (Figure 4.12a), one construction that could belong to either the ‘H’ or the ‘+4’ strategy (Figure 4.12b) and two partial ‘Spiral’ strategies (Figure 4.12c and d). The first two constructions are similar to the previous examples given in Figure 4.10a and Figure 4.11a, where the construction could belong to two strategies. This is again accurately reflected in the similarities displayed in Table 4.12. For the partial ‘Spiral’ strategies, the similarities in Table 4.12 show that the ‘Spiral’ strategy is clearly identified to be the most similar even when the construction is in its very incipient stages, i.e. only one or two cases (besides the pond) are used - see Figure 4.12c and d.

In the examples given in Figures 4.10 and 4.11, constructions including two cases that could equally belong to two strategies, were identified to be equally most similar to those corresponding strategies (see Figure 4.10a, Figures 4.10b, Figure 4.11a, Figure 4.11b and the similarities metrics in Table 4.10 and Table 4.11). Also, when one more case was added that belonged to only one of the two strategies, one strategy become most similar and was correctly identified by the similarity metrics - see Figure 4.10c, Figures 4.10d, Figure 4.11c, Figure 4.11d, Table 4.10 and Table 4.11.

To further test the identification mechanism, we looked at incipient partial strategies that have only two cases (excluding the pond) for which when taken separately at least one of the two cases could belong to more than one strategy, but when combined they belong to only one strategy. Several such partial strategies are displayed in Figure 4.13. For example, when taken separately, both cases (except the pond) in Figure 4.13a could belong to two strategies: the top bar of tiles could belong to the ‘I’ and ‘-4’ strategies, while the left bar of tiles could belong to the ‘I’ or the ‘+4’ strategies. For the construction in Figure 4.13b, only one case (the left bar of tiles) could belong to more than one strategy (more specifically, to the ‘I’ or the ‘+4’ strategies), while the other case, i.e. the tile in the top left corner belongs only to the ‘+4’ strategy. For the constructions in Figure 4.13c and d, both cases taken separately could belong to two different strategies. When combined, however, these cases could belong only to one strategy and this is accurately reflected in the similarity metrics

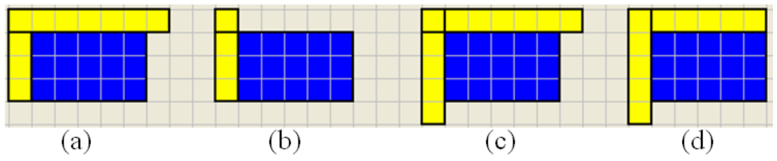


Figure 4.13: Partial strategies: (a) partial ‘I’ strategy; (b) partial ‘+4’ strategy; (c) partial ‘-4’ strategy (the highlighted corner is due to the overlap between the horizontal and the vertical bars of tiles); (d) partial ‘H’ strategy.

displayed in Table 4.13. Moreover, as the construction in Figure 4.13b is the least ambiguous of the four examples, i.e. only one case of the two could belong to more than one strategy, its similarity to the most similar strategy (5.00) is higher than for the other three examples (4.67) for which both cases taken separately could belong to more than one strategy.

Table 4.13: Similarity metrics for the partial strategies in Figure 4.13.

Stored Strategies	**Strategy a**	**Strategy b**	**Strategy c**	**Strategy d**
Area	1.72	1.55	1.72	1.67
I	**4.67**	1.90	2.55	1.73
H	1.73	1.61	2.55	**4.67**
Spiral	1.97	1.58	2.03	1.97
+4	2.55	**5.00**	1.73	2.55
-4	2.55	1.51	**4.67**	2.55

In conclusion, the strategy identification mechanism based on the similarity metrics presented in Section 4.3.2 can identify several situations of pedagogical importance with 100% success rate for the 38 tested strategies: complete specific, partially general or general strategies (6 strategies for each category - see Figure 4.5 which represents the general strategies), mixed strategies (2 strategies - see Figure 4.8), non-systematic strategies (2 strategies - see Figure 4.9) and partial strategies (16 strategies - see Figures 4.10, 4.11,4.12 and 4.13).

The next section presents the second iterative version of the exploratory learning environment and of the learner modelling mechanism.

4.4 Iterative Design Version 2

This section presents the next iterative version of the learner modelling mechanism. This was driven by the modification in the exploratory learning environment in general, and in the interface in particular. The presented version of the system has been developed over several smaller iterations, but we focus on a later version that covers all the features included gradually in the previous versions. Although what a learner can do in the system did not change at conceptual level, there were changes at practical level which have implications for the learner modelling mechanism. Therefore, the new mechanism is a refinement of the first version triggered by the modifications in the way learners interact with the system.

In the next section, an overview of the new system is given, outlining the changes from *ShapeBuilder* and the requirements presented in Section 4.1. The modification made to the learner modelling mechanism are also presented, together with an evaluation of the new version.

4.4.1 Version 2 of the Exploratory Environment: the *eXpresser*

Like the first version of the system, the second version is designed for classroom use and targets pupils of 11 to 14 year-olds. Also, each task involves two main phases: constructing a model and deriving an algebraic-like rule from it. The features of the new version of the system, named *eXpresser*, have been informed by studies with pupils and teachers. Several changes took place that are presented below:

1. *eXpresser* allows the construction of patterns rather than shapes; therefore, *eXpresser* is more general than *ShapeBuilder* in terms of what can be constructed.
2. The ShapeList has been removed and property lists have been "attached" to each pattern that enable their creation and the inspection of their properties.
3. *Icon variables* are replaced by the so-called *T-boxes*; they serve the same purpose as the icon variables, but are defined to represent any of the properties of a pattern. Unlike icon variables that made a dependency relation unidirectional, T-boxes define multi-directional relations, i.e. when the variable defined by the T-box changes, the change is reflected in all related properties.
4. Two 'worlds' are included in *eXpresser* - the student's world where the student builds his/her constructions and rules, and the general world where a different instance of the student's construction is displayed. Also, the construction in the general world can be animated to display various instances of the same construction.
5. To enable the animation of patterns, in *eXpresser* the rules required by the task need to be defined and at least one dependency relations needs to be in place.
6. *eXpresser* supports collaborative activities, as well as individual ones.

Fig. 4.14 illustrates the system, the *property list* of a pattern (linked to another one) and an example of a rule. The screenshot on the left includes two windows: (a) the students' world, where the students build their constructions and (b) the general world that displays the same construction with a different value for the variable(s) involved in the task (placed in the area 'I need to vary' in both worlds), and where students can check the generality of their construction by animating their patterns (using the Play buttons). The presence of these two spaces allows learners to *work on a specific case (in the students' world) 'with an eye' on the general (in the general world).*

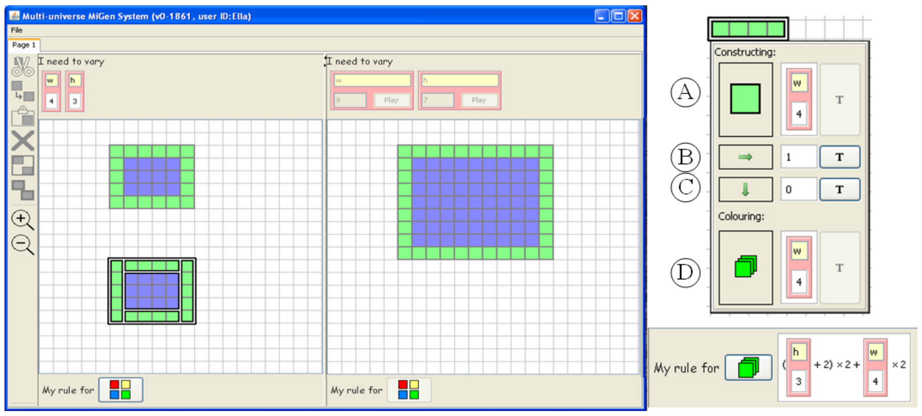


Figure 4.14: *eXpresser* screenshots. The screenshot on the left includes a toolbar, the students' world and the general world. The screenshot on the top right shows the property list of a pattern. The bottom right screenshot illustrates a rule.

We illustrate the affordances of *eXpresser* using the 'pond tiling' task previously introduced in Section 4.3.2 and displayed in the students' world with a 4 by 3 blue (darker colour) pond and in the general world with a 9 by 7 pond (the task requires to surround the pond and find a general rule for the number of tiles needed for this purpose). Here we illustrate the 'H' strategy; the components of this strategy are highlighted in the students' world for ease of visualisation: the 4 by 3 pond, 2 horizontal green (lighter colour) rows of 4 tiles and 2 vertical green bars of 5 tiles.

To *provide a rational for generality*, the tasks are presented dynamically. For example, the 'pond tiling' task is presented with the image displayed in Fig. 4.14 in the student's world without any highlighting of structure; this image changes regularly to show different instances of the pattern. This dynamic presentation provides "a rationale for deriving a rule that outputs the number of green tiles for any instance of the pattern, i.e. a 'general' rule giving concrete instantiation to the meaning of 'any' " (Geraniou et al., 2009, p. 52).

The property list of one of the horizontal bars is displayed in the top right screenshot. The first property (Ⓐ) specifies the number of iterations of the building-block, i.e. the basic unit of a pattern, which is displayed as an icon; the value for this attribute is set to the value of the width of the pond by using a

T-box (that includes a name and a value); by using a T-box, the two (or more) properties are made dependent, i.e. when the value in the T-box changes in one property, it also changes in the other one(s). The next properties are *move-right* (Ⓑ), which is set to 1, and *move-down* (Ⓒ), which is set to 0. The last property (Ⓓ) establishes the number for colouring all the tiles in the pattern - for this simple pattern the value is the same as the iterations and is also related to the width of the pond through the use of a T-box. The bottom right screenshot displays a rule for the number of green tiles: $(h + 2)$ x $2 + w$ x 2, where h and w stand for the T-boxes in the area 'I need to vary' (the same as the ones in property lists); a T-box can be displayed with name only, value only or both.

Through their multiple representation, the T-boxes are *scaffolding the route from numbers to variables*, emphasizing the idea that variables represent values, but those values do not need to be known - hence the enabled display of a T-box with value only, name only or both. Thus the transition from a specific value to a value that also has a name to a name only (i.e. variable) is facilitated: "this stands in contrast to the standard approach in which generalisations are constructed from special cases, and the path to the variable 'n' appears as a separate (often nonnegotiable) cognitive leap" (Geraniou et al., 2009, p. 54).

To make a construction general, T-boxes are needed to link the different parts of the construction. Without these links, a construction is specific, i.e. it is valid only as a particular instance of the task pattern; a construction can also have some links in place, while others are missing, i.e. the construction is partially general. This is essentially the same as in *ShapeBuilder*, except for the replacement of icon variables with T-boxes.

The use of property lists to construct patterns facilitates the derivation of the algebraic-like rule by the presence of the couloring property which refers to the number of tiles needed for certain parts of the construction; the rule is essentially formed by putting together the values of the colouring properties of all parts of a construction. Thus, the system supports *simultaneously model construction and analysis.*

To enable the dynamic presentation of a construction in the general world, the learners need to define a rule for the number of green tiles. This step was designed "to encourage students' reflection on their own actions. This process allows students to validate the generality of their final rule as well as a means to express their generalisations 'symbolically' " (Geraniou et al., 2009, p.55). Thus, the system supports *reflection on derived expressions.*

4.4.2 Learner Modelling Version 2

Although, the high level modelling mechanism essentially remained the same, several modifications were made to the structure of a case and a modification in the aggregated measure of similarity was introduced. For ease of readability, the knowledge representation and similarity metrics are given in full, rather than just outlining the changes from the previous version.

Case Representation

As in the previous version, we used Case-based Reasoning (Kolodner, 1993) as a starting point and modified the classic approach to fit our problems with multiple solutions. More specifically, the different strategies that can be used for building a construction are represented as series of cases with certain relations between them.

Definition 4.10 *A* ***case*** *is defined as $C_i = \{F_i, RA_i, RC_i\}$, where C_i represents the case and F_i is a set of attributes, corresponding to the property list of a pattern. RA_i is a set of relations between attributes and RC_i is a set of relations between C_i and other cases, respectively.*

Definition 4.11 *The set of attributes is defined as $F_i = \{\alpha_{i_1}, \alpha_{i_2}, \ldots, \alpha_{i_N}\}$.*

The set F_i includes two types of attributes: numeric and variables. The variables refer to different string values (i.e. type) that an attribute can take. Some numeric attributes are binary, indicating whether a case is a group of patterns, or can be considered in formulating a particular strategy through a 'part of strategy' function $PartOfS_u : C_i \to \{0, 1\}$

$$PartOfS_u = \begin{cases} 1 & \text{if } C_i \in S_u \\ 0 & \text{if } C_i \notin S_u, \end{cases}$$

where S_u represents a strategy and is defined further on. The set of attributes of a generic case for *eXpresser* is presented in Table 4.14. The first v attributes ($\alpha_{i_j}, j = \overline{1, v}$) are variables, the ones from $v + 1$ to w are numeric ($\alpha_{i_j}, j = \overline{v + 1, w}$) and the rest are binary ($\alpha_{i_j}, j = \overline{w + 1, N}$).

The complete list of attributes that have a type (variable) and a value (numeric) is: width, height, iterations, 'move left', 'move right' and colour. All except the first two are from the property list of a pattern.

Compared with the first version, the new one has the same definition for a case, but several modifications occurred in the set of attributes:

Table 4.14: The set of attributes (F_i) of a case.

Category	**Name**	**Label**	**Possible Values**
Patterns properties	Colour	α_{i_1}	Red/Green/Blue/Yellow
	Width type	α_{i_2}	Number(n)/T-box(Tb)/ Numeric expression(ne)/ Expression with T-box(es)(eTb)
	Height type	α_{i_3}	n /Tb /ne /eTb
	$\vdots$	$\vdots$	$\vdots$
	Colour type	α_{i_v}	n /Tb /ne /eTb
	Width value	$\alpha_{i_{v+1}}$	numeric value
	Height value	$\alpha_{i_{v+2}}$	numeric value
	$\vdots$	$\vdots$	$\vdots$
	Colour value	$\alpha_{i_{w-2}}$	numeric value
Location coordinates	x	$\alpha_{i_{w-1}}$	numeric value
	y	α_{i_w}	numeric value
Group flag	isGroup	$\alpha_{i_{w+1}}$	0/1
Part of Strategy	$PartOfS_1$	$\alpha_{i_{w+2}}$	0/1
	$PartOfS_2$	$\alpha_{i_{w+3}}$	0/1
	$\vdots$	$\vdots$	$\vdots$
	$PartOfS_r$	α_{i_N}	0/1

1. Shape type was taken out because in *eXpresser* patterns are used rather than shapes. Also, patterns can be made by grouping smaller patterns - this is now reflected in the Group flag;

2. All the attributes under the Dimensions of shape category in the first version were replaces with attributes about the properties of patterns; only two of these attributes are the same in both versions, i.e. width and height;

3. Locations coordinates were added; these play a role in defining a strategy, as detailed below.

Definition 4.12 *The* ***set of relations between attributes*** *of the current case and attributes of other cases is represented as $RA_i = \{RA_{i_1}, RA_{i_2}, \ldots, RA_{i_M}\}$, where at least one of the attributes in each relation $RA_{i_m}, \forall m = \overline{1,M}$, is from F_i, the set of attributes of the current case C_i.*

Two types of binary relations are used: *dependency relations* and *value relations.*

Definition 4.13 *A* ***dependency relation*** *(D_{i_s}) is defined as $(\alpha_{i_k}, \alpha_{j_l}) \in D_{i_s} \Leftrightarrow \alpha_{i_k} = DEP(\alpha_{j_l})$, where $DEP : \alpha_{i_k} \rightarrow \alpha_{j_l}$ for attributes α_{i_k} and α_{j_l} that are variables of cases i and j (where $i = j$ or $i \neq j$).*

This means that α_{i_k} depends on α_{j_l} (if $i = j$, $k \neq l$ is a condition as to avoid circular dependencies) (e.g. the iterations of a case is linked to the colouring of another case, the width a case is linked to the height of the same case). The dependency relations are defined by the learners through the use of T-boxes.

Definition 4.14 *A* ***value relation*** *(V_{i_s}) is defined as $(\alpha_{i_k}, \alpha_{j_l}) \in V_{i_s} \Leftrightarrow \alpha_{i_k} = f\left(\alpha_{j_l}\right)$, where α_{i_k} and α_{j_l} are numeric attributes and f is a (linear) function and could have different forms depending on context (e.g. the iterations of a case are x times the iterations of another case, the height of a shape is its width plus y).*

Definition 4.15 *The* ***set of relations between cases*** *is represented as $RC_i = \{RC_{i_1}, RC_{i_2}, \ldots, RC_{i_P}\}$, where one of the cases in each relation $RC_{i_j}, \forall j = \overline{1, P}$ is the current case (C_i).*

The relations between cases are defined based on time; therefore, two time-relations are used: $Prev$ and $Next$. Each case includes at most one of each of these two relations.

Definition 4.16 *A $Prev$ relation indicates the previous case with respect to the current case: $(C_i, C_j) \in Prev$ if $t\left(C_j\right) < t\left(C_i\right)$.*

Definition 4.17 *A $Next$ relation indicates the next case with respect to the current case: $(C_i, C_k) \in Next$ if $t\left(C_i\right) < t\left(C_k\right)$.*

Definition 4.18 *A* ***strategy*** *is defined as $S_u = \{N_u(C), N_u(RA), N_u(RC), N_u(LC)\}$, $u = \overline{1, r}$, where*
$N_u(C)$ is a set of cases,
$N_u(RA)$ is a set of relation between attributes of cases,
$N_u(RC)$ is a set of relations between cases, and
$N_u(LC)$ is a set of location constraints. These location constraints have the following form: $x_{C_j} = f(x_{C_i})$ and $y_{C_j} = f(x_{C_i})$, where x and y represent location coordinates, f and g are linear functions, and C_i is a referential case (a case that constitutes a reference for other cases). Therefore, the set of constraints is defined as $LC = \left\{\left(x_{C_j}, y_{C_j}\right)\right\}$, where j takes values from 1 to the number of cases in S_u and $j \neq i$.

Compared with the first version, the definition of a strategy in the new version has a new component, i.e. a set of location constraints.

In the following, a task called 'stepping stones' is used to illustrate the case-base representation. It requires to build a construction such as the one in Figure 4.15a and to find a rule for the green (lighter colour) tiles in relation to the red (darker) tiles, i.e. the 'stepping stones'. Several strategies to surround such a pattern are possible; however, we present here only the ones observed in studies with students (see Figure 4.15). Some constructions are expanded for ease of visualisation and the variable 'red' refers to the number of red tiles. In these figures, the internal structure of the constructions has been highlighted for clarity. In *eXpresser* all constructions would look the same in the normal course of the task. Three of these strategies (i.e. 'C' strategy, 'HParallel' strategy and 'VParallel' strategy) are considered more 'desirable' in the sense that they facilitate generalisation (especially when moving from the construction to defining a rule); the 'Squares' strategy is less desirable because it involves overlaps, which have been observed to be ignored by many students, consequently leading to wrong rules.

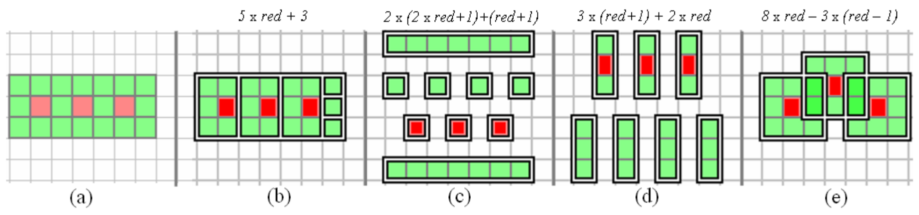


Figure 4.15: 'Stepping stones' task constructions and associated rules: (a) the task construction regardless of structure; (b) the 'C' strategy; (c) the 'HParallel' strategy; (d) the 'VParallel' strategy; (e) the 'Squares' strategy

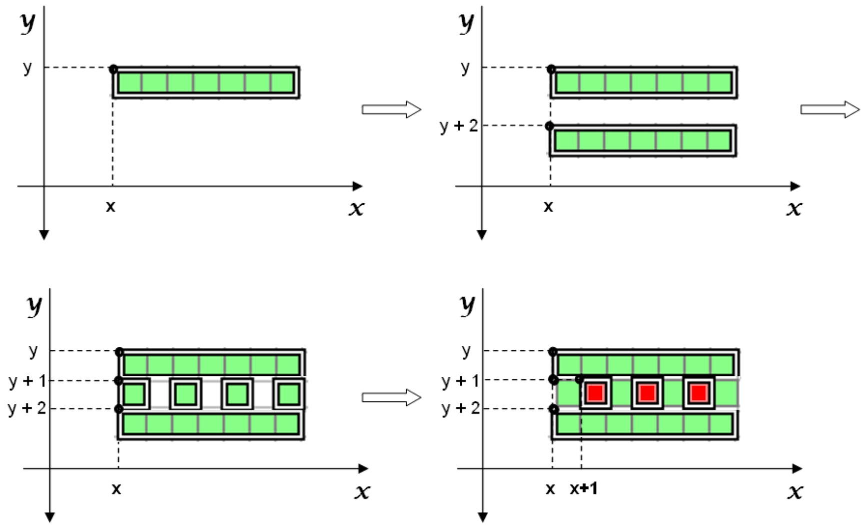


Figure 4.16: Possible steps for 'HParallel' strategy with location details.

Besides multiple possible constructions, there are several ways of reaching the same construction. A possible trajectory is illustrated for the 'HParallel' strategy in Fig. 4.16, including location details. The learner may start with an horizontal bar of green tiles, copy it and place it below and then add a middle bar of green tiles with gaps between them. Finally, a bar of red tiles with gaps between them is placed in the gaps of the middle bar of green tiles. The definition of the strategy is given for each step of this particular construction in Table 4.15.

Table 4.15: S_u definition for each step of the 'HParallel' strategy.

S_u	$N_u\ (C)$	$N_u\ (RA)$	$N_u\ (RC)$	$N_u\ (LC)$
Step 1	C_1	-	-	-
Step 2	C_1, C_2	-	$Next(C_1) = C_2$ $Prev(C_2) = C_1$	$x_{C_2} = x_{C_1}$ $y_{C_2} = y_{C_1} + 2$
Step 3	$C_1, C_2,$ C_3	-	$Next(C_i) = C_{i+1}$ for $i = 1, 2$ $Prev(C_{i+1}) = C_i$ for $i = 1, 2$	$x_{C_i} = x_{C_1}$ for $i = 2, 3$ $y_{C_2} = y_{C_1} + 2$ $y_{C_3} = y_{C_1} + 1$
Step 4	C_1, C_2 C_3, C_4	$\alpha_{i_4} = 2 * \alpha_{4_4}$ for $i = 1, 2$ $\alpha_{3_4} = \alpha_{4_4} + 1$	$Next(C_i) = C_{i+1}$ for $i = 1, 3$ $Prev(C_{i+1}) = C_i$ for $i = 1, 3$	$x_{C_i} = x_{C_1}$ for $i = 2, 3$ $x_{C_4} = x_{C_1} + 1$ $y_{C_2} = y_{C_1} + 2$ $y_{C_3} = y_{C_1} + 1$
Step 5	C_1, C_2 C_3, C_4	$\alpha_{i_4} = 2 * \alpha_{4_4}$ for $i = 1, 2$ $\alpha_{3_4} = \alpha_{4_4} + 1$ for $i = 3$ $\alpha_{i_4} = DEP(\alpha_{4_4})$ for $i = 1, 2$ $\alpha_{3_4} = DEP(\alpha_{4_4})$	$Next(C_i) = C_{i+1}$ for $i = 1, 3$ $Prev(C_{i+1}) = C_i$ for $i = 1, 3$	$x_{C_i} = x_{C_1}$ for $i = 2, 3$ $x_{C_4} = x_{C_1} + 1$ $y_{C_2} = y_{C_1} + 2$ $y_{C_3} = y_{C_1} + 1$

Up to Step 4, the construction is build in a specific way. In Step 5 the construction is general - all dependencies are presented, although in a real situation, they would be built in separate steps (three in this case); to avoid repetition, they are aggregated into one step.

Similarity Metrics

The same four similarity measures that were defined for *ShapeBuilder* are used for *eXpresser*:

(a) *Numeric attributes* - Euclidean distance: $D_{IR} = \sqrt{\sum_{j=v+1}^{w} (\alpha_{I_j} - \alpha_{R_j})^2}$ (I and R stand for input and retrieved cases, respectively);

(b) *Variables*: $V_{IR} = \sum_{j=1}^{v} g(\alpha_{I_j}, \alpha_{R_j})/v$, where g is defined as: $g(\alpha_{I_j}, \alpha_{R_j}) = 1$ if $\alpha_{I_j} = \alpha_{R_j}$ and $g(\alpha_{I_j}, \alpha_{R_j}) = 0$ if $\alpha_{I_j} \neq \alpha_{R_j}$.

(c) *Relations between attributes* - Jaccard's coefficient: $A_{IR} = \frac{|RA_I \cap RA_R|}{|RA_I \cup RA_R|}$. A_{IR} is the number of relations between attributes that the input and retrieved case have in common divided by the total number of relations between attributes of the two cases;

(d) *Relations between cases* - Jaccard's coefficient: $B_{IR} = \frac{|RC_I \cap RC_R|}{|RC_I \cup RC_R|}$, where B_{IR} is the number of relations between cases that the input and retrieved case have in common divided by the the total number of relations between cases of I and R.

To identify the closest strategy to the one employed by a learner, cumulative similarity measures are used for each of the four similarity types:

(a) Numeric attributes - as this metric has a reversed meaning compared to the other ones, i.e. a smaller number means a greater similarity, the following function is used to bring it to the same meaning as the other three similarity measures, i.e. a greater number means greater similarity:

$$F_1 = \begin{cases} \frac{z}{\sum_{i=1}^{z} D_{I_i R_i}} & \text{if } \sum_{i=1}^{z} D_{I_i R_i} \neq 0 \\ z & \text{if } \sum_{i=1}^{z} D_{I_i R_i} = 0, \end{cases}$$

(b) Variables: $F_2 = (\sum_{i=1}^{z} V_{I_i R_i})/z$;

(c) Relations between attributes: $F_3 = (\sum_{i=1}^{z} A_{I_i R_i})/y$;

(d) Relations between cases. $F_4 = (\sum_{i=1}^{z} B_{I_i R_i})/z$.

where z represents the minimum number of cases among the two compared strategies and y represents the number of cases in the retrieved strategy that have relations between attributes; for example, the 'C' strategy has three cases and only one that has relations between attributes, i.e. $y = 1$.

The similarity between the current strategy and a stored strategy is defined as the sum of these four measures after normalisation is applied as explained below. As the numeric strategy has a different range from the other three similarity metrics, normalisation is applied to have a common measurement scale, i.e. $[0, 1]$. This is done using linear scaling to unit range (Aksoy and Haralick, 2001) by applying the following function: $\overline{x} = \frac{x-l}{u-l}$, where x is the value to be normalised, l is the lower bound and u is the upper bound for that particular value. Consequently, as the range of F_1 is $[0, z]$, the normalisation function is: $\overline{F_1} = F_1/z$.

After normalisation, weights are applied to the four similarity metrics to express the central aspect of the construction - the structure. This is mostly reflected by the $\overline{F_1}$ metric; the F_3 metric is also important as the structure of a construction is also reflected in the relations between attributes of component cases. The other two metrics (F_2 and F_4), although important for the generality of construction and the order of cases, respectively, have less impact on the structure. Considering these aspects, the aggregated similarity is computed using the following function: $Sim = 6 * \overline{F_1} + F_2 + 2 * F_3 + F_4$.

In *ShapeBuilder* no normalisation was applied, which introduced a positive weighting on the F_1 metric. For *eXpresser* the normalisation and weighting was introduced to better control the contribution of each metric in identifying the most similar strategy.

4.4.3 Evaluation

In this section we present experiments conducted to validate the second version of the modelling mechanism by doing a low-level testing with a focus on how this approach operates in practice to identify various types of exploratory learning behaviour. To this end, real data were used from two classroom sessions using *eXpresser* that ran in July 2008 in a secondary school in London. There were 18 students in each session (the same students took part in both sessions) who were previously familiarised with an earlier version of the software (Gutiérrez et al., 2008).

The purpose of the first session was to familiarise the students with the new version of the system and little time was spent on the 'stepping stones' task. The second session focused on the 'stepping stones' task and two more tasks were available should the students finish the main task. The distribution of the students with respect to the strategies they followed is displayed in Table 4.16. Some learners built several constructions using different strategies (4 learners in session 1 and 3 learners in session 2) ; for these students their last construction was used, and thus we used only one construction from each learner. No student generalised their construction in the first session, while in the second one only three students reached generalised constructions.

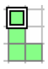
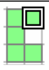
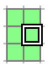
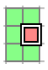
Table 4.16: Distribution of students according to the followed strategy

Strategies	Session 1	Session 2
C	5	6
HParallel	2	4
VParallel	2	2
Squares	2	1
Other	7	5

Log extract

Two extracts of one log file are presented in Tables 4.17 and 4.18 illustrating the actions of one of the students when solving the 'stepping stones' task using the 'C' strategy. The corresponding constructions are also displayed in the tables, with the pattern of the current step highlighted - for example, Step 2 includes a construction with two tiles - the leftmost one is highlighted and the corresponding part of the log displays the information that is logged when that

Table 4.17: Log extract - part 1

No	Log extracts	Construction
	15:05:21,116 - This is the eXpresser log for 'BR199037'.	
1	15:10:30,354 - (Model1) Pattern Shape351 created.	
	15:10:30,369 - Attribute 'Att2562' ('x') has value '13'.	
	15:10:30,385 - Attribute 'Att2563' ('y') has value '3'.	
	15:10:30,385 - Attribute 'Att2564' ('width') has value '1'.	
	15:10:30,385 - Attribute 'Att2565' ('height') has value '1'.	
	15:10:30,400 - Attribute 'Att2581' ('inc x') has value '0'.	
	15:10:30,416 - Attribute 'Att2582' ('inc y') has value '0'.	
	15:10:30,416 - Attribute 'Att2568' ('iterations') has value '1'.	
	15:10:30,416 - Attribute 'Att2566' ('colour') has value '[r=0,g=255,b=0]'. *(green)*	
	15:10:30,432 - Attribute 'Att2561' ('name') has value ''.	
	15:10:30,432 - Attribute 'Att2567' ('shape') has ID 'Shape352'.	
2	15:10:31,258 - (Model1) Pattern Shape354 created.	
	15:10:31,258 - Attribute 'Att2584' ('x') has value '12'.	
	15:10:31,258 - Attribute 'Att2585' ('y') has value '3'.	
	15:10:31,273 - Attribute 'Att2586' ('width') has value '1'.	
	15:10:31,289 - Attribute 'Att2587' ('height') has value '1'.	
	15:10:31,289 - Attribute 'Att2603' ('inc x') has value '0'.	
	15:10:31,289 - Attribute 'Att2604' ('inc y') has value '0'.	
	15:10:31,304 - Attribute 'Att2590' ('iterations') has value '1'.	
	15:10:31,304 - Attribute 'Att2588' ('colour') has value '[r=0,g=255,b=0]'. *(green)*	
	15:10:31,304 - Attribute 'Att2583' ('name') has value ''.	
	15:10:31,304 - Attribute 'Att2589' ('shape') has ID 'Shape355'.	
3	15:10:31,990 - (Model1) Pattern Shape357 created.	
4	15:10:33,674 - (Model1) Pattern Shape360 created.	
5	15:10:34,593 - (Model1) Pattern Shape363 created.	
6	15:10:35,404 - (Model1) Pattern Shape366 created.	
	15:10:35,419 - Attribute 'Att2672' ('x') has value '13'.	
	15:10:35,419 - Attribute 'Att2673' ('y') has value '4'.	
	15:10:35,419 - Attribute 'Att2674' ('width') has value '1'.	
	15:10:35,419 - Attribute 'Att2675' ('height') has value '1'.	
	15:10:35,435 - Attribute 'Att2691' ('inc x') has value '0'.	
	15:10:35,435 - Attribute 'Att2692' ('inc y') has value '0'.	
	15:10:35,435 - Attribute 'Att2678' ('iterations') has value '1'.	
	15:10:35,450 - Attribute 'Att2676' ('colour') has value '[r=0,g=255,b=0]'. *(green)*	
	15:10:35,450 - Attribute 'Att2671' ('name') has value ''.	
	15:10:35,450 - Attribute 'Att2677' ('shape') has ID 'Shape367'.	
7	15:10:35,466 - (Model1) Object Shape366 has been selected.	
	15:10:38,334 - (Model1) Attribute 'Att2676' ('colour') has changed from '[r=0,g=255,b=0]' *(green)* to '[r=255,g=0,b=0]]' *(red).*	
	15:10:38,350 - Shape Shape366 now has colour java.awt.Color[r=255,g=0,b=0]. *(red)*	

tile is created. To avoid unnecessary repetition, some of the steps in Table 4.17 do not have all logged details. This applies only for steps where the missing information could be easily inferred - for example, the details of Step 3 are similar to the ones of the previous two steps (the only variations are the IDs of attributes/shapes and the values of the location coordinates).

Steps 1 to 6 and 10 to 12 illustrate the creation of very simple patterns - namely, one tile. When such a pattern is created, all its attributes are logged: location coordinates (x and y), width, height, 'move left' (inc x), 'move down' (inc y), the number of iterations, the colour and the name if one has been given to it. Step 7 illustrates what is logged when an attribute of a shape is changed. In this particular case, the colour has changed from green to red. Step 8 illustrates the creation of a group. The attributes of the group are displayed: location coordinates, width, height, colour and name. The shapes that are part of the group are also listed. Step 9 illustrates the iteration of a pattern that is a group - the new shape created by repeating the previously created group is logged with all its attributes.

We also illustrate how the log translates into the formal definitions of cases and strategies - see Table 4.19. These are given for Step 1, 8 and 12, that illustrate respectively a simple case that is not a group, a simple case that is a group and a composite case, i.e. a strategy.

Identification of strategies

This section presents the identification of strategies for each of the scenarios introduced earlier in the evaluation for *ShapeBuilder* (Section 4.3.3). The same procedure was used as in the evaluation of *ShapeBuilder*, i.e. the outputs of the identification mechanism were checked by the author against the screen videos that were collected for all learners.

One more scenario was added that covers off-task behaviour, i.e. engaging in behaviour that does not involve the system or the learning task (Baker et al., 2004) as an indicator of other issues such as lack of challenging tasks, lack of motivation or disliking of mathematics, the teacher or the system. Off-task behaviour has been proved to lead to poor learning (Baker, 2007) in intelligent tutoring systems and therefore, it is one of the important aspects to monitor; this may be even more of an issue in exploratory learning environments (ELEs) due to the freedom of exploration given in ELEs. The findings of the experiments are discussed for each category.

Complete strategies. Complete strategies refer to complete constructions that were built using one of the stored strategies. They could be *specific*, *gen-*

Table 4.18: Log extract - part 2

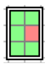
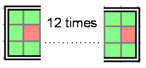
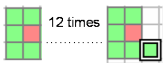
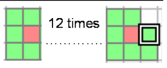
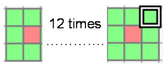
No	Log extracts	Construction
8	15:10:44,460 - Pressable ‘Group’ pressed. 15:10:44,475 - (Model1) Group Shape369 created. 15:10:44,475 - Attribute ‘Att2694’ (‘x’) has value ‘12’. 15:10:44,491 - Attribute ‘Att2695’ (‘y’) has value ‘3’. 15:10:44,491 - Attribute ‘Att2696’ (‘width’) has value ‘2’. 15:10:44,491 - Attribute ‘Att2697’ (‘height’) has value ‘3’. 15:10:44,537 - Attribute ‘Att2698’ (‘colour’) has value ‘null’. 15:10:44,537 - Attribute ‘Att2693’ (‘name’) has value ‘’. 15:10:44,537 - Shape-1: Shape363. 15:10:44,553 - Shape-2: Shape351. 15:10:44,553 - Shape-3: Shape360. 15:10:44,553 - Shape-4: Shape366. 15:10:44,569 - Shape-5: Shape357. 15:10:44,569 - Shape-6: Shape354.	
9	15:10:49,073 - Button ‘Create horizontal’ used. 15:10:49,167 - (Model1) Pattern Shape370 created. 15:10:49,167 - Attribute ‘Att2700’ (‘x’) has value ‘12’. 15:10:49,182 - Attribute ‘Att2701’ (‘y’) has value ‘3’. 15:10:49,182 - Attribute ‘Att2702’ (‘width’) has value ‘24’. 15:10:49,182 - Attribute ‘Att2703’ (‘height’) has value ‘3’. 15:10:49,182 - Attribute ‘Att2845’ (‘inc x’) has value ‘2’. 15:10:49,198 - Attribute ‘Att2846’ (‘inc y’) has value ‘0’. 15:10:49,198 - Attribute ‘Att2706’ (‘iterations’) has value ‘12’. 15:10:49,198 - Attribute ‘Att2704’ (‘colour’) has value ‘[null]’. 15:10:49,213 - Attribute ‘Att2699’ (‘name’) has value ‘’. 15:10:49,213 - Attribute ‘Att2705’ (‘shape’) has ID ‘Shape369’.	
10	15:10:51,536 - (Model1) Pattern Shape599 created. 15:10:51,536 - Attribute ‘Att4366’ (‘x’) has value ‘36’. 15:10:51,536 - Attribute ‘Att4367’ (‘y’) has value ‘3’. 15:10:51,551 - Attribute ‘Att4368’ (‘width’) has value ‘1’. 15:10:51,551 - Attribute ‘Att4369’ (‘height’) has value ‘1’. 15:10:51,551 - Attribute ‘Att4385’ (‘inc x’) has value ‘0’. 15:10:51,551 - Attribute ‘Att4386’ (‘inc y’) has value ‘0’. 15:10:51,567 - Attribute ‘Att4372’ (‘iterations’) has value ‘1’. 15:10:51,567 - Attribute ‘Att4370’ (‘colour’) has value ‘[r=0,g=255,b=0]’. 15:10:51,567 - Attribute ‘Att4365’ (‘name’) has value ‘’. 15:10:51,583 - Attribute ‘Att4371’ (‘shape’)‘ has ID ‘Shape600’.	
11	15:10:52,487 - (Model1) Pattern Shape602 created.	
12	15:10:53,796 - (Model1) Pattern Shape605 created. 15:10:53,812 - Attribute ‘Att4410’ (‘x’) has value ‘36’. 15:10:53,812 - Attribute ‘Att4411’ (‘y’) has value ‘5’. 15:10:53,812 - Attribute ‘Att4412’ (‘width’) has value ‘1’. 15:10:53,827 - Attribute ‘Att4413’ (‘height’) has value ‘1’. 15:10:53,827 - Attribute ‘Att4429’ (‘inc x’) has value ‘0’. 15:10:53,827 - Attribute ‘Att4430’ (‘inc y’) has value ‘0’. 15:10:53,843 - Attribute ‘Att4416’ (‘iterations’) has value ‘1’. 15:10:53,843 - Attribute ‘Att4414’ (‘colour’) has value ‘[r=0,g=255,b=0]’. 15:10:53,843 - Attribute ‘Att4409’ (‘name’) has value ‘’. 15:10:53,858 - Attribute ‘Att4415’ (‘shape’) has ID ‘Shape606’.	

Table 4.19: Knowledge representation for Steps 1, 8 and 12

	Step 1
F_1	$\{green, n, n, n, n, n, n, 1, 1, 1, 0, 0, 1; 13, 3; 0; 1, 0, 0, 0\}$
RA_1	ϕ
RC_1	ϕ
	Step 8
F_1	$\{green, red, n, n, n, n, n, n, 2, 3, 1, 0, 0, 5, 1; 12, 3; 1; 1, 0, 0, 0\}$
RA_1	$\{no\_greens = 5 * no\_reds\}$
RC_1	ϕ
	Step 12
$N_1(C)$	$\{C_1, C_2, C_3, C_4\}$
$N_1(RA)$	$\{iterations(C_1) = no\_reds\}$
$N_1(RC)$	$\{Next(C_i) = C_{i+1}, Prev(Ci + 1) = C_i, i = \overline{1,3}\}$
$N_1(LC)$	$\{x(C_i) = x(C_1) + width(C_1), i = \overline{2,4}; y(C_2) = y(C_1 + 2, y(C_3) = y(C_1) + 1,$ $y(C_4) = y(C_1)\}$

eral or *partially general*. One of the most important steps in generalisation is the transition from a specific to a general construction, an aspect that was observed to require support in the studies with pupils. Therefore, detecting if the learners are working towards a specific or a general construction is an important feature of the proposed modelling approach. This is particularly important in two situations: (a) when a learner makes the transition or 'mental jump' from specific to general, so that help with this transition can be provided if needed, and (b) when this transition has been made with at least one element of the construction (depending on the strategy, several parts of the constructions need to be made general, e.g. the 'HParallel' strategy), but the learner is unsure about it and is reluctant to proceed without feedback.

Findings of the experiments show that if the construction has been built in a general way, the similarity with the corresponding stored strategies has the maximum value (i.e. 10); if built in a specific way (no general component), the similarity is 9.33. If some components of the constructions are general and some are specific, the similarity value varies between 9.33 and 10. The 'Squares' strategy is an exception to this, as it involves only one pattern. The similarity metrics for each specific and general strategy compared to the stored general strategies are displayed in Table 4.20.

Mixed strategies. Often learners are not systematic in their approach to solving a task, which can lead to: (a) inefficient strategies such as one-by-one constructions in which the learners use only individual tiles, placing them one after the other to obtain the desired look of a construction, without thinking about the structure or the generality of their approach; (b) combinations of different strategies that are more easily generalisable when used on their own rather than mixed with others. As mentioned previously, the one-by-one ap-

Table 4.20: Similarity metrics for complete specific and general user strategies. Note that the stored strategies are general.

		Stored Strategies			
		C	HParallel	VParallel	Squares
User strategies	C specific	**9.33**	2.18	2.33	1.66
	C general	**10.00**	2.51	2.83	2.66
	HParallel specific	2.18	**9.33**	2.34	1.34
	HParallel general	3.18	**10.00**	3.34	2.34
	VParallel specific	2.33	2.34	**9.33**	1.48
	VParallel general	3.33	2.84	**10.00**	2.48
	Squares	1.66	1.34	1.48	**7.00**

proach is considered as a landmark and is detected from raw data produced by the learner actions and does not need any involvement of the CBR mechanism. We concentrate here on the detection of mixed strategies, whose pedagogical value is threefold: (a) support the learner in deriving a general expression if their arithmetics abilities are good; (b) guide the learners towards one of the strategies that is reflected in their construction if the learners' arithmetics abilities are low and, therefore, the use of only one strategy will facilitate deriving a general expression; (c) point out that systematic and symmetric approaches are desirable.

In the two sessions, only one case was observed where a combination of strategies occurred and the student finalised the construction (but did not generalised it). The construction is displayed in Fig. 4.17 - the various components are presented apart from each other for ease of visualisation. Nevertheless, in studies using the 'pond tiling' task, mixed strategies occurred more frequently as reported in Section 4.3.3.

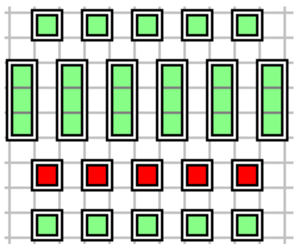


Figure 4.17: Combination of 'VParallel' and 'HParallel' strategies.

The similarity metrics for the comparison of the construction in Fig. 4.17 with all stored strategies are displayed in Table 4.21. The construction is adequately identified as being most similar to the 'VParallel' and 'HParallel' strategies.

Table 4.21: Similarity of the construction in Fig. 4.17 to stored strategies

	C	HParallel	VParallel	Squares
Fig. 4.17 construction	2.28	**2.37**	**3.14**	1.48

Non-systematic strategies. As mentioned previously, learners are often non-systematic in their approach to a task, leading to constructions made of 'bits and pieces' that are difficult to generalise. Sometimes they start with such an approach and gradually move towards something more systematic; however, they do not always replace all the 'bits and pieces' they started with.

An example of such a situation is given in Fig. 4.18a. The overall approach corresponds to the 'HParallel' strategy, but the top raw of green tiles is composed of 'bits and pieces': one row of six tiles plus one tile; also the raw of six tiles is composed as a group of three tiles repeated twice, as can be seen in the property list displayed in Fig. 4.18b.

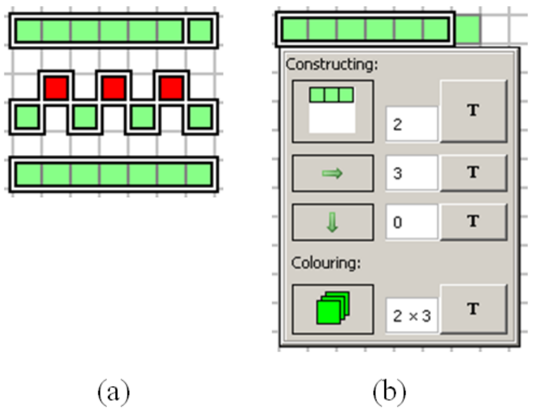


Figure 4.18: Strategy with non-systematic parts: (a) the structure of the strategy; (b) the property list of the top component.

Some students will change this as they are moving towards making a construction general; however, others will not reach that level - often not because they do not know what to do (as they have done it on other components of the construction) but because they forget that they have built that part of the construction with 'bits and pieces'. Identification of this type of learning behaviour could be useful in practice. For some students, a simple highlighting of the structure could be enough to draw their attention and no further feedback would be necessary. Others, however, might move to the rule extraction stage without changing the construction and this could potentially confuse them at a later time. For the particular student whose strategy is presented in Fig. 4.18a, the values of the similarity metric with respect of stored strategies are shown in Table 4.22. The maximum value reflects the similarity in structure to the 'HParallel' strategy; however, the fact that the value is in the lower end of the similarity range indicates the students' construction has parts that do not entirely correspond to the 'HParallel' strategy.

Table 4.22: Similarities of the construction in Fig. 4.18 to stored strategies

	C	HParallel	VParallel	Squares
Fig. 4.18 construction	2.09	**4.31**	2.30	1.32

Detection of non-systematic strategies would be useful also as an opportunity to point out to learners that systematic approaches and symmetry facilitate the process of generalisation. For example, presenting the learners with different expressions corresponding to symmetric/systematic[1] and non-symmetric/non-systematic constructions could encourage learners' reflection on the usefulness of symmetric and systematic approaches. The expressions corresponding to the non-systematic strategy illustrated in Fig. 4.18a and its most similar strategy are illustrated in Table 4.23.

Table 4.23: Algebraic-like rules for symmetric and non-symmetric constructions.

Construction	Rule
Fig. 4.18a	$(2 * red + 1) + (2 * red) + 1 + (red + 1)$
Fig. 4.15c	$2 * (2 * red + 1) + (red + 1)$

Partial strategies. Some students start to solve the task following a certain strategy, but do not continue with that approach immediately. In our experiments, learners sometimes tried different things, corresponding to different strategies, and after this period of experimentation, they adopted one particular approach or a combination of approaches. Therefore, it is important to identify what strategy the learners are currently working with should they ask for help at that point. For each strategy two partial constructions are illustrated in Fig. 4.19, corresponding to different stages in the learners' process of building a construction. They are essentially snapshots of learners' constructions at different points in time, where the first snapshot, corresponding to version 1 is taken fairly early, while the second one, corresponding to version 2, is taken later, but

[1]Systematic constructions are constructions that facilitate generalisation in terms of ease of translation to an algebraic-like rule; some systematic constructions are also symmetric, such as the strategies for the 'pond tiling' task or the 'HParallel' and 'VParallel' strategies for the 'stepping stones' task, while others are not, such as the 'C' and 'Squares' strategies for the 'stepping stones' task.

before the learners have completed the construction. The purpose of this is to test the metrics in relation to how early they can detect a partial construction as similar to a particular strategy. The similarity metrics comparing each partial construction of Fig. 4.19 with the stored strategies are displayed in Table 4.24.

Because the ‘Squares’ strategy has only one component, partial constructions cannot occur; therefore, this strategy was not included in this section. One may argue that different stages of building the ‘square’, i.e. the building block of this strategy, could be considered a partial construction; however, in our definition of partial constructions, it is essential to work with groups or patterns rather than single tiles. In other words, we consider partial constructions as incipient systematic constructions rather than building something that looks like the desired construction but is made of one-by-one tiles or of ‘bits and pieces’.

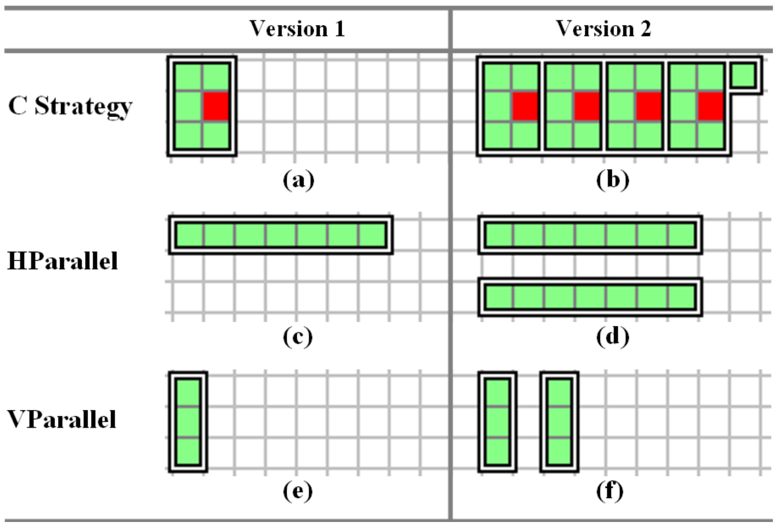


Figure 4.19: Partial strategies: (a) partial ‘C’ v1; (b) partial ‘C’ v2; (c) partial ‘HParallel’ v1; (d) partial ‘HParallel’ v2; (e) partial ‘VParallel’ v1; (f) partial ‘VParallel’ v2.

Table 4.24: Similarities of the construction in Fig. 4.19 to stored strategies

	C	HParallel	VParallel	Squares
Partial C v1 (Fig. 4.19a)	**7.00**	2.41	2.90	2.73
Partial C v2 (Fig. 4.19b)	**7.83**	2.06	2.35	1.49
Partial HParallel v1 (Fig. 4.19c)	1.61	**7.00**	1.93	1.33
Partial HParallel v2 (Fig. 4.19d)	2.13	**7.83**	2.49	1.33
Partial VParallel v1 (Fig. 4.19e)	2.80	2.50	2.41	2.08
Partial VParallel v2 (Fig. 4.19f)	2.28	1.93	**7.00**	1.48

The construction in Fig. 4.19e has a low similarity with all stored strategies, as it could be the starting point for three of them (listed in the order of the similarity metrics in Table 4.24): 'C' strategy, 'VParallel' and 'Squares'. The results, however, show that 'HParallel' strategy has a higher similarity than 'VParallel' and 'Squares' strategy. Therefore, at this stage, it is too early to assess the strategy the learner is following and the results of the identification mechanism are not reliable for finding the most similar strategy. The similarity metrics, however, show very close values for all strategies, which could be an indication that with the available information no strategy can be considered most similar to the learner's construction. One the other hand, the construction in Fig. 4.19e is clearly most similar to the 'VParallel' strategy. Consequently, assessment of learners' constructions should not be made too early and if help is requested at such an early stage, the students should be encouraged to continue with building the construction.

Detecting off-task exploratory behaviour. Detecting whether students are off-task or not is a challenge in ELEs. On one hand, the freedom for exploration offered to students can lead to increased off-task behaviour. On the other hand, the various types of exploratory behaviour observed, ranging from explorations of how the system works to task–related explorations, should be effectively distinguished from off-task behaviour as they can be potentially useful to the learning process. Below we present an attempt to identify off-task behaviour using the data provided by the monitoring mechanism and a few rules.

We use an example observed in the second classroom session that falls in the category of off-task, and we compare it with a behaviour which, although on-task, is inefficient in terms of its generalisation capability. The constructions corresponding to these two situations are illustrated in Fig. 4.20. The construction in Fig. 4.20b looks very much like the 'C' strategy in the final outcome, but it is constructed from a group of ten individual tiles that are repeated twice which makes it difficult to generalise. Nevertheless, unlike the construction in Fig. 4.20a, this construction is task-related.

The rules for the off-task detection employ two sources of information: (a) the characteristics of the task in terms of variations of width and height, and the relation between them if there is any, and (b) the relative position of patterns. The information about the task acts as a valuable filter when random tiles are left on the pattern construction area along with a construction that is related to the task. For example, in the footpath task these characteristics are: the height is three, the width can vary (with a minimum of three), and there is no particular relation between the height and the width. With this information,

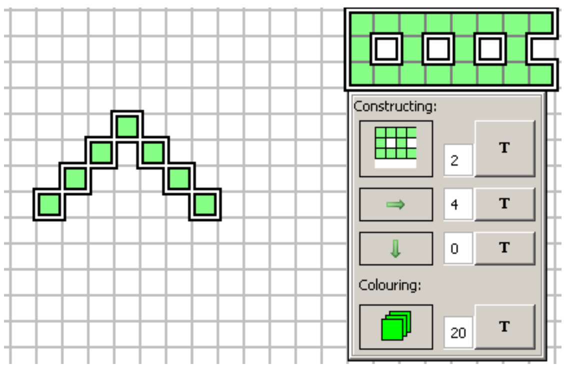


Figure 4.20: (a) off-task; (b) on-task, inefficient strategy

from Fig. 4.20a two different constructions can be obtained by filtering out either the top or the bottom row to satisfy the ‘height equals three’ characteristic. Then, the second source of information about the relative position of patterns is used. The relative positions are mapped to the position constraints of the stored strategies and if there is at least a partial correspondence, the similarity metrics are calculated. If there is no correspondence, no similarity metrics are computed. In our example, the constructions obtained by filtering the top or the bottom row of Fig. 4.20a do not map the position constraints of any stored strategy. Therefore, using both sources of information we conclude that the learner is off-task. Algorithm 1, presented below, describes the proposed approach.

Algorithm 1 OffTask($StrategiesCaseBase$, $InputStrategy$)

```
if $Task.Width$ is fixed then
    if $InputStrategy.Width > Task.Width$ then
        cap $InputStrategy$ {cap left or right}
    end if
end if
if $Task.Height$ is fixed then
    if $InputStrategy.Height > Task.Height$ then
        cap $InputStrategy$ {cap top or bottom}
    end if
end if
for all combinations of capped constructions do
    verify position constraints
    if at least one constraint is satisfied then
        calculate similarity to all strategies in $StrategiesCaseBase$
    else
        return student is off-task
    end if
end for
```

For the construction in Fig. 4.20b, the height of the construction corresponds to the task characteristics and the relative positions of the patterns partially map the position constraints of two strategies. Therefore, the similarity metrics are computed and the ‘C’ strategy is identified as the most similar strategy with a value of 3.01.

In conclusion, the second version of the mechanism can also identify situations of pedagogical importance with 100% success rate for the 26 strategies that were tested: complete specific, partially general or general strategies (4, 2 and 3 strategies respectivly - see Figure 4.15), mixed strategies (1 startegy - see Figure 4.17), non-systematic strategies (10 strategies - see one example in Figure 4.18) and partial strategies (6 strategies - see Figure 4.19).

4.5 Discussion

The results of the evaluation studies conducted for the two iterative versions indicate the CBR-based model for monitoring the exploratory learning behaviour is able to identify situations of pedagogical importance, such as working in a specific or a general way, working with several strategies at one time, working with non-systematic strategies, working with partially constructed strategies and going off-task.

The evaluation involved 10 pupils for the first version and 18 pupils for the second version. Although this may seem a small sample, the access to participants needs to be considered. School systems rarely allow researchers access to students, limit the number of participants and put constraints on the setting in which evaluation can take place (Hossain and Brooks, 2008). Thus, a balance was needed between having the evaluation in a realistic situation and the number of students available. Therefore, the limitation in the sample size was compensated by 'real-life' data.

This work shows that, despite the challenges raised by the nature of interaction in exploratory learning environments, monitoring learners' activity and identification of particular situations that carry a pedagogical meaning is possible. Moreover, learner diagnosis is possible *during* the process of solving a task rather than at the end, opening possibilities for more effective personalised intelligent support. Our work also addresses other issues generally encountered in exploratory learning environments such as inactivity, working with the right variables, interpretation of data and generalisations. These aspects are discussed below for *eXpresser* and the second version of the modelling mechanism.

As the learner's actions are monitored continuously, *inactivity* can be easily detected. A further assessment of the stage within a task at the occurrence of inactivity could indicate possible causes of it. For example, if the learner has built a construction that is specific and then is inactive, two causes are likely: (a) the learner does not know what to do next in terms of the requirements of the task; (b) the learner understands that the next step is to generalise his/her construction but does not know how to do that. The former is a conceptual problem, while the latter is pragmatic - knowing what one needs to do but not knowing how to do it using the tools of the system. There could also be situations where the learner is inactive from the very beginning, in which case it is very difficult to infer any possible causes. Besides causes related to the task, there could also be other types of causes that lead to learners' inactivity and they may have to do with motivational status (Pintrich and Schunk, 2002) and attitudes to subject matter (Beal et al., 2006). Having some knowledge about these could shed some light on the causes of inactivity, especially when this occurs from the beginning of the task.

Because of the way strategies are defined, it is possible to detect if the students are *choosing the right variables* to works with. For example, in the 'C' strategy of the 'stepping stones' task (Fig. 4.15b), there are one or more 'correct' (or rather useful) variables they can choose: the number of iterations of the 6-tile group and the number of colour allocations for the red tiles. Similarly, for the 'HParallel' strategy (Fig. 4.15c) the useful variables are the number of iterations of the red tiles and the number of colour allocations for the red tiles. Therefore, the monitoring mechanism is able to detect if the learners are choosing the right variables because this information is encoded in the strategies.

Interpretation of data in exploratory learning is usually associated with experiments - by looking at the various results of several experiments the students are to accept or reject a hypothesis. In the context of *eXpresser*, the high level hypothesis corresponds to the algebraic-like rule that the learners need to find, which is built from the relations between patterns expressed as T-boxes. The interpretation of data corresponds to observing changes in their construction when learners vary the values of the patterns' properties, and the relation between these changes and the algebraic-rule, i.e. is the rule valid or not? Moreover, lower level hypotheses are tested before the extraction of an algebraic-like rule, when the learners are 'figuring out' the relations between patterns that lead them to a construction that 'looks right', in which case the hypotheses correspond to the relations between patterns. For example, learners may test numerical relations, such as "for a footpath of 3 red tiles I need 4 iterations of the vertical bar of 3 green tiles" to act according to the 'VParallel' strategy (Fig. 4.15d) or general relations such as "the iterations of the vertical bar of 3 tiles need to be the number of red tiles plus 1" to act according to the same strategy. This is tightly related to generalisation and helping students generalise. Our monitoring mechanism can identify if the learner's construction is specific (i.e. they are working with numeric relations), general (i.e. all the general relations are in place) of partially general (i.e. some general relations are in place, while others are missing), thus providing the necessary information for enabling personalised feedback related to interpretation of data and generalisation.

Although in a limited way, the monitoring mechanism can detect *off-task* and inform the teacher who can take finer-grained decisions depending on whether the learner is really off-task or just off-track. Therefore, the off-task detection needs additional improvement to reduce further the workload of the teacher. Factors that could be incorporated in a model for off-task detection are: the colours used in the construction, the number of cases in the learner's construction with respect to the minimum and maximum number of cases of the stored strategies for that particular task, and the speed at which the cases are constructed. For example, if a learner is quickly building many 'bits and pieces' of which some have many different colours (as opposed to fewer colours in the task specification), the learner may be constructing something for their own amusement rather than for solving the task.

A potential *limitation* of our approach is that it depends on the activity of the learner - a minimal construction is needed to infer what strategy the learner is using, to detect if a learner works with the specific or the general and also, to infer if a learner is off-task. As mentioned previously, although inactivity can be detected, its causes are difficult to identify when the learner has not constructed anything. Another limitation of our mechanism is that apart from being error-prone, the off-task detection is relying on the teacher to make sure that the learner is truly off-task; on the other hand, the system is designed for classroom use with the teacher present, making it less of a problem.

Currently, the *aggregated similarity metric* for strategies has fixed weights that maximise the identification of structural similarity; however, depending on the stage within a task, these could vary to maximise identification of other aspects like the generality of construction. This direction of research is part of our future work. Our work could also be extended in the direction of Open Learner Models, as reflection is an important part of discovery learning (de Jong, 2006) and the open learner models have been shown to encourage learners' reflection on their own learning (Bull et al., 2006; Bull and Kay, 2007).

The *pedagogical scenarios* defined in Section 4.3.3 played a central role in the evaluation of both versions of the learner modelling mechanism. These were defined with input from teachers and educational experts in a series of iterative user studies that could be best described as part of the Persistent Collaboration Methodology (PCM) (Conlon and Pain, 1996). These scenarios became requirements for the development of the learner modelling mechanism and were gradually introduced in an incremental manner.

As an *iterative and incremental approach* was adopted due to the iterative design of the system and changes of the interface affordances were expected, most of the effort was directed to design the initial version of the learner modelling mechanism in a flexible manner that could cope with such changes. As CBR offered flexibility and extendibility, it was chosen for our purpose; however, typical CBR covers problems with uniques solutions, while we needed a mechanism that would identify the most similar solution from a case-base of solutions for the same problem. Consequently, we modified the typical CBR approach to make it suitable for problems with multiple solutions.

Due to the iterative approach, *changes* were expected for both the knowledge representation and the similarity metrics. Changes in the structure and/or attributes of cases were expected because of the iterative design of the system in general and of the interface in particular. In fact, these changes occurred both in the structure and the attributes of a case. For example, the change in structure involved adding new attributes (e.g. move-right, more-down), while the change in attributes involved changes in the values that an attribute could take (e.g. the type values for *ShapeBuilder* that were constant/variable/icon variable/numeric expression/expression with icon variable, were replaced for *eXpresser* with number/T-box/numeric expression/expression with T-box(es)). Also, changes in the similarity metrics were expected; however, these changes were made in the combination of the four metrics used rather than the basic metrics themselves.

4.6 Summary and Contribution of the Chapter

In this chapter overviews of mathematical generalisation, iterative design and case-based reasoning were presented, together with the two iterative versions of an exploratory learning environment and their corresponding versions of a learner modelling mechanism.

The learner modelling mechanism has at its centre the idea of *strategy* - a particular way of reaching a solution. In our approach, a strategy is defined as a series of cases related between them by certain properties. While the learner is solving a task, his/her construction is compared with stored strategies - for this purpose, similarity metrics have been defined to compare cases and strategies. In the second version, using different weights for the four similarities defined to compare cases, the emphasis of the aggregated similarity metric for strategies was given to *structure*, as the most important aspect of a construction.

Pedagogically-driven scenarios were defined that the monitoring mechanism should be able to identify: complete strategies, mixed strategies, non-systematic strategies, partial strategies and off-task behaviour. All except the last one rely exclusively on the similarity metrics to identify the mentioned situations. Using data from small case studies and classroom sessions, the proposed monitoring mechanism was validated for all the above mentioned scenarios.

Several challenges were encountered in the development of the proposed monitoring mechanism. The first and most difficult one was to decide what aspects of the learners' interactive behaviour to monitor. Early trials with pupils showed that one important aspect was the way they visualised a construction. Therefore, the focus was given to strategies as different ways to reach the same construction. The next challenge was how to represent and reason about strategies in such a way that it would be possible to have an idea about the strategy a learner used even if that strategy was not complete. In other words, it was important to identify the approach of a learner *during* a task rather than at the end of it. To address this challenge, we have modified the classic case-based reasoning to fit our purpose: a case was defined as part of a solution and strategies were defined as series of cases that constitute a solution.

Another challenge was the handling of non-systematic strategies, by which we mean building a construction with no particular strategy in mind. For example, in the 'pond tiling' task surrounding the pond with random 'bits and pieces'. As mentioned in the second scenario, i.e. identification of non-systematic strategies, it is difficult to generalise when a construction has these 'bits and pieces'. This was defined as a *landmark* to look for in the learners' actions, as it indicates lack of structure and potential difficulty for generalisation. To handle this, the following rule was used: if the construction was completely made of these 'bits and pieces', similarities with the stored strategies were not computed at this point unless some position constraints were satisfied. If the construction was entirely build of 'one-by-one' tiles, i.e. another landmark, no similarity was computed either.

Another challenge was the detection of off-task behaviour. Although error-prone and virtually reliant on the teacher to ascertain that the learner is actually off-task, our mechanism can detect off-task behaviour, an aspect that should be especially considered in exploratory learning environments because of their very core characteristic: the freedom given to the learner. Future work will look into ways to improve the off-task detection and make it less reliant on the teacher.

Although the mechanism we propose is tailored to *eXpresser*, it could be generalised to other exploratory learning environments where a learner is asked to build a model (which in the case of *eXpresser* is a pattern construction) made of several parts and where these parts could be joined in different ways.

To summarise, the contribution of this chapter consists of:

1. A learner modelling mechanism based on a version of CBR suitable for problems with multiple solutions that allows diagnosis during the task as

well as at the end of it. Although developed for a particular learning environment, this mechanism could be generalised to other exploratory learning systems. Moreover, it could be extended beyond educational systems to manufacturing problems such as block assembly or to design problems with reusable parts.

2. A set of scenarios of pedagogical importance for the evaluation of the learner modelling mechanism;

3. An iterative approach to learner modelling, in which the flexibility and extendibility of the approach is essential to enable modifications across the iterative versions, because "user models cannot and should not be separated from the software systems that use them" (Chin, 2001, p. 183)

Chapter 5

Feedback Prioritisation

In the previous chapter we have described how we diagnose what a learner is doing in the learning environment by using a learner modelling mechanism. This information is further used in this chapter to address a pedagogical issue, i.e. feedback prioritisation.

In exploratory learning, tasks can be approached in many different ways and are often characterised by some key points the learner needs to address or be aware of. The actions of learners can indicate what they need help with, but their personal characteristics may not guarantee the effectiveness of help. Also, context could bring valuable information that would make help more appropriate and, thus, more effective. Context-awareness has been studied in a diversity of domains like artificial intelligence (Akman et al., 2001), ubiquitous computing (Kwon, 2006), educational psychology (Wang et al., 2008) and recommender systems (Anand and Mobasher, 2007). The definition of context is also diverse, varying from the wide social context to the specificity of network characteristics.

This chapter presents a context-dependent personalised feedback prioritisation mechanism using the Analytic Hierarchy Process (Saaty, 1980), a popular method in Multi-criteria Decision Making (Zopounidis and Doumpos, 2002). Two versions are presented, corresponding to the versions of the system used - *ShapeBuilder* and *eXpresser*. In the proposed approach context refers to the stages within a task for both versions; each task has two stages depending on the generality of construction and expression, i.e. a stage where learners work with specific constructions and expressions, and one where they work with partially or completely general constructions and expressions. The second version also refers to the learning mode, i.e. individual or collaborative. Both versions were tested using scenarios based on data from classroom trials and the second version was validated by education experts.

For both versions, the aspects to give feedback on were decided based on pedagogical information about the tasks and the affordances of the system, and can be applied to all tasks. Also, for both versions, the relations between the feedback prioritisation module and the other relevant components are the same - see Figure 5.1. The feedback prioritisation module takes into consideration the learner characteristics and information about the task (from Task Model) to calculate priorities, which are filtered based on the information in the Task LTM. These relations are illustrated with double line arrows. The single line arrows illustrate relations that do not directly involve the feedback prioritisation component of the feedback module. For example, there is a two-way relation between the Learner Model and the Feedback Module, i.e. the Feedback Module takes into consideration information from the Learner Model, and the output of the Feedback Module is registered in the Learner Model (the learner model includes a history of the feedback that was given[1]).

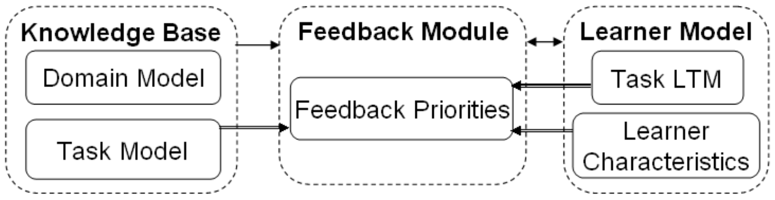


Figure 5.1: Relations between the Feedback Prioritisation Module and other relevant components.

The next section presents a brief overview of personalised feedback. Section 5.2 introduces Multi-criteria Decision Making and presents in detail one method called the Analytic Hierarchy Process which is used in the proposed mechanism. Section 5.3 presents the first version of the proposed mechanism, while Section 5.4 presents the second version.

5.1 Personalised Feedback

Feedback is usually a response to the actions of a learner aiming to correct future iterations of the actions (Mason and Bruning, 2001). It includes information about what happened or did not happen as a consequence of the user's actions in relations to the goal (Wiggins, 2008). This information is given to the users to compare their performance with the expected one (Johnson and Johnson, 1993) and to make use of it in the following attempt (Wiggins, 2008).

There are several types of feedback reported in the literature. Table 5.1 (adapted from Gouli et al. (2006)), summarises several adaptive feedback approaches according to their context (theoretical level or computer-based learn-

[1]this is out of the scope of this thesis and is, therefore, not discussed any further.

ing environments), the underlying domain and goals/processes served (guiding, tutoring, reflection), the types of feedback supported, and the adaptation process (adaptivity and adaptability). The adaptivity mechanism supported is also given (gradual provision of the same type of feedback or different types of feedback and/or adaptation of feedback according to one or more of the learner's individual characteristics).

The adaptive feedback mechanisms presented in Table 5.1 accommodate mainly the learners' knowledge level while a limited degree of flexibility is provided so that learners can adjust and intervene in the feedback presentation process. In case of gradual provision of feedback, usually the same type of feedback is provided in different steps, while the amount of feedback is differentiated. Also, the research approaches are mainly focusing on the guiding and tutoring processes and are usually restricting the help support in a domain-specific way. Thus, open issues in the area are (i) the design of a framework which supports the provision of adaptive as well adaptable feedback in a way that enhances learning and serves processes such as reflection, and (ii) the design of a general domain-independent form of feedback able to be incorporated in different learning environments and to serve a variety of domains.

The feedback mechanisms presented above were all developed in the context of structured environments and we are not aware of any similar work in the context of exploratory learning environments. Our aim is not to fill this gap by deriving a framework for personalised feedback as in Gouli et al. (2006), but to focus on a particular aspect of the feedback generation process, i.e. prioritisation, and explain why this is particularly important in ELEs.

From the feedback types mentioned in Table 5.1, we present below some of the more general ones, which we believe are relevant for exploratory learning environments:

(a) *Informative feedback* refers to the following type of information: correctness or incorrectness of response; confirmation in case of a correct response; performance feedback - usually in the form of a percentage or a number on a scale;

(b) *Tutoring feedback* refers to more information than the simple outcome; it imitates the actions of a human tutor. There are two levels for this type on feedback: the exploratory level that takes place during problem solving and the explanation of response level that take place after an answer has been submitted. The exploratory level includes feedback in several forms: an image, an example, a similar problem followed by its solution, solutions of others given to a specific problem. The explanation of response includes why the wrong answer is wrong and why the right answer is right;

Table 5.1: Overview of research on personalised feedback.

Research Efforts	Context	Domain	Goal/Processes	Types of Feedback supported	Adaptation Process
Mason and Bruning (2001)	Theoretical Framework	Domain Independent	Assist developers and instructors in developing effective feedback in computer-based educational settings	(i) Knowledge-of-correct-response with response-contingent (ii) Knowledge-of-correct-response with topic-contingent (iii) Knowledge-of-response with topic-contingent (iv) Knowledge-of-response with delayed knowledge-of-correct-response plus response-contingent (v) Answer-until-correct with delayed topic-contingent	Adaptivity based on the learners' knowledge level and prior knowledge Variables such as task complexity and timing of feedback are taken into consideration for the adaptation of feedback
Narciss and Huth (2004)	An adaptive tutoring feedback algorithm is proposed and implemented in the context of a multimedia learning environment	Mathematics	Tutoring/Guiding	(i) Knowledge of response (ii) Bug-related feedback	Gradual provision of different types of feedback following a 3-step feedback procedure
Excel Tutor (Mathan and Koedinger, 2003)	Computer-based learning environment	Computer Science	Tutoring/Guiding	(i) Questions having the form of multiple choice (ii) Succinct explanations of errors	Gradual provision following a 3-step feedback procedure
SQL Tutor (Mitrovic and Martin, 2000)	Computer-based learning environment	Computer Science	Tutoring/Guiding	(i) positive/negative feedback (ii) error flag (iii) hint (iv) all errors (v) partial solution (vi) complete solution	Gradual provision of the first three types of feedback Adaptability supported only for the last three types of feedback
Animalwatch	Computer-based learning environment	Mathematics	Tutoring/Guiding		
(Arroyo et al., 2000)				Hints	Gradual increasing the level of information
(Arroyo et al., 2001)				Hints: a classification of the hints is supported according to their degree of symbolism and their degree of interactivity	Adaptation based on learners' cognitive development and gender
Fiedler and Tsovaltzi (2003)	An algorithm proposed in the context of the DIALOG project	Mathematics	Tutoring/Guiding	Hints (A taxonomy of hints is supported)	Gradual provision from less to more informative hints based on the number and kind of hints produced so far, the number of wrong answers and the category of the learner's answers
Stern et al. (1996)	Computer-based learning environment	Mathematics	Tutoring/Guiding	Hints	Gradual presentation from simple to more specific hints Adaptation based on learners' knowledge level
Adaptive Feedback Framework (AFF)	Theoretical Adaptive Feedback Framework realised in the web-based learning environment COMPASS	Domain Independent	Reflection/Tutoring /Guiding	(i) Correctness-Incorrectness of Response (ii) Correct Response (iii) Performance Feedback (iv) Tutoring Feedback Units associated with various modes of knowledge modules such as a definition, an example, a similar problem and solution of others (v) Explanation of the Response (vi) Belief Prompt-Rethink Write (vii) Error-Task Related Questions	Gradual provision of the different types of feedback (feedback components) following their layered structure (four layers are supported) and based on the category of the learner's answer Adaptation based on learners' knowledge level, preferences and interaction behavior Adaptability supported for all layers of feedback and feedback components

(c) *Reflective feedback* is meant to encourage reflection and usually consists of prompts from the system. It includes two types: belief prompt-rethink write and error-task prompts. The first type can be generic or directed to a particular aspect; this kind of prompt applies to correct actions or answers as well. Error-task related prompts are targeted to specific errors and question the learner's belief about the correctness of their input.

Usually feedback is presented in layers with a typical order being: (1) belief prompt-rethink write, (2) error-task related questions, (3) tutoring feedback and (4) performance feedback.

In exploratory learning, the freedom given to learners leads to situations when feedback is required on several aspects. Unlike previous research on personalised feedback, the focus is on providing personalised prioritisation of feedback. In our search of the literature we did not find any reported research on this topic.

The next section presents an overview of Multi-criteria Decision Making in general and of the Analytic Hierarchy Process in particular, as the latter is used in our proposed approach.

5.2 Multi-criteria Decision Making

Multi-criteria Decision Making (MDM) defines a class of problems where a decision from a predefined set of alternatives needs to be reached by taking into account two or more criteria. Each alternative is evaluated on the set of criteria and the outcomes would provide a means of comparison between the alternatives that will facilitate a selection of one or some alternatives, or a ranking between them. Other purposes are classification of alternatives into groups (clustering) and group ranking (Zopounidis and Doumpos, 2002). Among the possible approaches of decision problems that correspond to this description are: statistical techniques, multi-attribute utility analysis, analytic hierarchy process, knowledge bases, mathematical models, etc.

The Analytic Hierarchy Process (AHP) (Saaty, 1980) is one of the most popular methods in MDM and is widely applied in a diversity of areas such as: logistics (Chan et al., 2006; Partovi, 2006), military (Korkmaz et al., 2008; Crary et al., 2002), manufacturing (Ertay et al., 2006; Shinno et al., 2006) and healthcare (Kwak and Lee, 2002; Lee and Kwak, 1999). In most cases, AHP is used in combination with other methods (Ho, 2008); a recent literature review (Ho, 2008) reports five main categories of tools integrated with AHP: (a) mathematical programming, (b) quality function development, (c) meta-heuristics, (d) SWOT (Strengths, Weaknesses, Opportunities, and Threats) analysis and

(e) data envelopment analysis. Four works related to higher education are reported in areas of IT-based project selection (Kwak and Lee, 1998), teaching method selection (Lam and Zhao, 1998), education requirement selection (Koksal and Egitman, 1998) and faculty course assignment (Ozdemir and Gasimov, 2004).

In the area of user modelling, AHP has been used in combination with fuzzy logic (Grigoriadou et al., 2002) for student diagnosis in an adaptive hypermedia educational system and in combination with Multi-Attribute Utility Theory (MAUT), another method from MDM, in recommender systems (Schmitt et al., 2003), where the evaluation function from MAUT is used to rate how well each alternative fulfills the decision criteria. The rest of this section includes a description of the AHP formalism.

The Analytic Hierarchy Process uses a hierarchical or network structure to represent a decision problem and to establish priorities between alternatives depending on a set of criteria involved in the decision process. It includes three main steps: (a) construction of the hierarchy; (b) analysis of priorities and (c) verification of consistency.

The process (see Figure 5.2) starts with the definition of the hierarchy and is followed by several sets of pairwise comparisons for the criteria involved and the possible alternatives. These result in weights for the criteria and priority vectors for the alternatives; their consistency is verified and they are used to calculate the composite weights of alternatives or final priorities; the consistency of the whole hierarchy is also verified. If the consistency condition is not satisfied, revisions of the pairwise comparisons are necessary for a trustworthy decision.

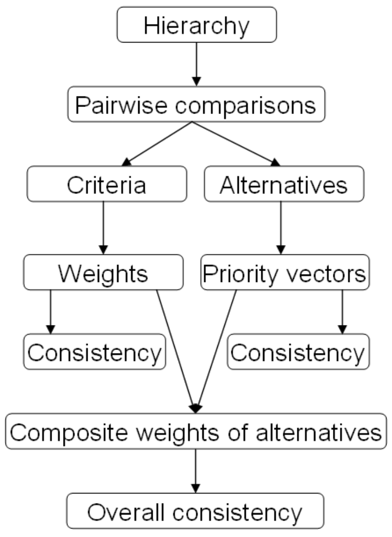


Figure 5.2: AHP Process

Hierarchy. The hierarchy has the general structure represented in Figure 5.3. The highest level represents the *goal*; the second level includes the *criteria* based on which the decision should be taken; the third level includes the possible *alternatives* which will be prioritised with respect to the criteria. The first step includes a decomposition of the decision problem into parts defined by all relevant attributes; these attributes are arranged into hierarchical levels so as to reach the hierarchical structure presented in Figure 5.3.

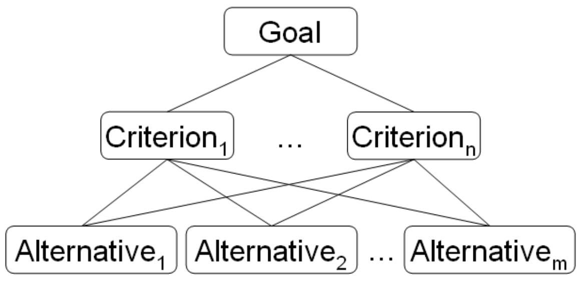


Figure 5.3: Hierarchy in the Analytic Hierarchy Process

Priorities. The analysis of priorities includes pairwise comparisons used to compute weights for the alternatives; the weights will establish an order between the alternatives. This process involves two substeps: (a) decide the priorities between the criteria; (b) decide the priorities between alternatives with respect to each criterion. The priorities take the form of matrices (see (5.1)): one for the first substep (priorities amongst criteria) and n for the second substep (priorities amongst alternatives) (a matrix for each criterion). For both types of matrices the values below the main diagonal are the reversed values from above the main diagonal, i.e. $c_{ji} = 1/c_{ij}, a_{ji} = 1/a_{ij}$, as the comparison result between two objects A and B is reversed when the order changes, i.e. between B and A.

$$\begin{bmatrix} 1 & c_{12} & ... & c_{1n} \\ 1/c_{12} & 1 & ... & c_{2n} \\ ... & ... & ... & ... \\ 1/c_{1n} & 1/c_{2n} & ... & 1 \end{bmatrix} \begin{bmatrix} 1 & a_{12} & ... & a_{1m} \\ 1/a_{12} & 1 & ... & a_{2m} \\ ... & ... & ... & ... \\ 1/a_{1m} & 1/a_{2m} & ... & 1 \end{bmatrix} \tag{5.1}$$

Each pair of criteria c_i and c_j has an associated value that specifies their relative importance. The values of c_{ij} $(1 \leq i, j \leq n)$ and a_{ij} $(1 \leq i, j \leq m)$ are determined using a scale from 1 to 9, where 1 means 'equally important' and 9 means 'extremely more important'. For example, $c_{ij} = 1$ means that the criteria c_i and c_j are equally important, $c_{ij} = 3$ means that c_i is more important than c_j and $c_{ij} = 9$ means that c_i is extremely more important than c_j. The values and meaning for the inverse pairs are: $c_{ji} = 1$ means that c_j and c_i are equally important, $c_{ji} = 1/3$ means that c_j is less important than c_i and $c_{ji} = 1/9$ means that c_j is extremely less important than c_i.

The weight of each criterion is calculated using (5.2) and the criteria weight vector is obtained: $W = (w_1, w_2, \ldots, w_n)$.

$$w_i = \frac{\left(\prod_{j=1}^{n} c_{ij}\right)^{1/n}}{\sum_{i=1}^{n}\left(\prod_{j=1}^{n} c_{ij}\right)^{1/n}} \tag{5.2}$$

For the alternatives, a priority vector is calculated for each matrix (corresponding to a criterion) using the same equation (5.2). Thus priority vectors: $A(Cr_j) = (A_1(Cr_j), A_2(Cr_j), \ldots, A_m(Cr_j))$, $j = \overline{1,n}$ are obtained. Matrix A (5.3) results from combining the n priority vectors.

$$A = \begin{bmatrix} A_1(Cr_1) & A_1(Cr_2) & \cdots & A_1(Cr_n) \\ A_2(Cr_1) & A_2(Cr_2) & \cdots & A_2(Cr_n) \\ \vdots & \vdots & \ddots & \vdots \\ A_m(Cr_1) & A_m(Cr_2) & \cdots & A_m(Cr_n) \end{bmatrix} \tag{5.3}$$

By combining the criteria weights and the priority vectors the final alternatives priorities vector P with respect to all criteria is obtained using (5.4). More specifically, the priority for each alternative is calculated as (5.5).

$$P = A * W \tag{5.4}$$

$$p_i = A_i(Cr_1) * w_1 + A_i(Cr_2) * w_2 + \ldots + A_i(Cr_n) * w_n, i = \overline{1,m} \tag{5.5}$$

Consistency. To verify the consistency of the $n+1$ pairwise comparisons matrices (n alternatives matrices and 1 criteria matrix), an approximation of the maximum eigenvalue for each matrix, denoted as λ_{max} (see Equation 5.6) is used to calculate the consistency index (CI). Equation (5.7) shows how to calculate CI for the criteria matrix and the n alternatives matrices.

$$\begin{gathered} \lambda_{max_j} = (\sum_{i=1}^{m} a_{i1}, \sum_{i=1}^{m} a_{i2}, \ldots, \sum_{i=1}^{m} a_{im}) * \\ (A_1(Cr_j), A_2(Cr_j), \ldots, A_m(Cr_j))^T, j = \overline{1,n} \end{gathered} \tag{5.6}$$

$$\begin{aligned} \text{For criteria: } CI &= \frac{\lambda_{max} - n}{n-1} \\ \text{For alternatives: } CI_j &= \frac{\lambda_{max_j} - m}{m-1},\ j = \overline{1,m} \end{aligned} \tag{5.7}$$

CI and the Random Consistency Index (RCI) are used to calculate the consistency ratio (CR) as in (5.8). The values of the RCI for 1 to 10 criteria are displayed in Table 5.2. Values of the consistency ratio below 0.10 indicate consistency, while greater values indicate the opposite. In the latter case, revision of the pairwise comparisons is necessary.

$$CR = \frac{CI}{RCI} \tag{5.8}$$

Table 5.2: Values of RCI for $n = \overline{1, 10}$.

n	1	2	3	4	5	6	7	8	9	10
RCI	0	0	0.58	0.90	1.12	1.24	1.32	1.41	1.45	1.49

The overall consistency of the hierarchy is a function of the consistency indexes of all pairwise matrices, the RCI for the number of criteria and number of alternatives and the weights of the criteria, as in (5.9).

$$CR = \frac{CI_{criteria} + w_1 * CI_{alt_{Cr_1}} + w_2 * CI_{alt_{Cr_2}} + \ldots + w_n * CI_{alt_{Cr_n}}}{RCI_n + w_1 * RCI_m + w_2 * RCI_m + \ldots + w_n * RCI_m} \tag{5.9}$$

Summarising, the AHP process involves three main steps: definition of the hierarchy, analysis of pairwise comparisons and verification of consistency. These are illustrated for *ShapeBuilder* in the following section.

5.3 Feedback Prioritisation - Iterative Version 1

This section presents the developed approach for feedback prioritisation in the context of *ShapeBuilder*. It includes a description of the AHP formalism for our specific problem and an evaluation through the use of scenarios.

5.3.1 Feedback Prioritisation for *ShapeBuilder*

This section presents how the AHP process translates in the context of *ShapeBuilder*. The hierarchy as in the AHP formalism is illustrated in Figure 5.4. The highest level of the hierarchy represents the goal, which in our case is to obtain feedback priorities. The next level represents the criteria taken into consideration, which refers the type of context: specific or general. The context corresponds to the stage within a task; thus, a learner is either working with a construction and/or expression that is specific, or with a construction and/or expression that is partially or completely general. The lowest level of the hierarchy designates the alternatives, which in our case correspond to giving feedback on the following aspects:

(a) correctness of construction (CC) - the learners' construction should correspond to the construction for the particular task they are working on. For each task there are several ways to build a correct construction which are stored in the knowledge base; therefore, the learners' construction should correspond to one of them;

(b) correctness of expression (CE) - when reduced to its simplest form, the learners' expression should correspond to the expression for the particular task they are working on;

(c) construction-expression correspondence (C-E) - there should be a correspondence between the learners' expressions and their constructions; for example, if they build a construction using a particular strategy, but the expression corresponds to another strategy, the expression does not correspond to the construction.

(d) symmetry of construction (Sym) - this refers to the way the construction is built; symmetry is a characteristic of efficiently built constructions;

(e) generality of construction (CGen) - the learner's construction should be general, i.e. all relevant components of the constructions should be general;

(f) generality of expression (EGen) - the learner's expression should be general, i.e. it should work for any instance of the task;

(g) use of icon variables (IV) - the learners should use icon variables to make their constructions and expressions general.

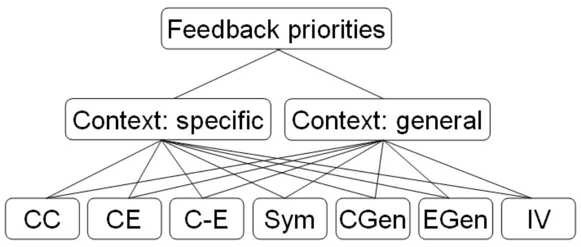


Figure 5.4: AHP hierarchy.

The alternatives presented above were selected based on pedagogical information about the tasks and on the affordances of *ShapeBuilder*. They define the aspects that are relevant for all tasks.

The pairwise comparisons between criteria and between alternatives vary depending on task and learner characteristics:

(a) task difficulty, which could be low, medium or high; this is given by the researcher or teacher;

(b) level of experience (stored for each level of task difficulty), which is measured by the number of tasks the learners were exposed to for each level of difficulty; for example, if a learner has previously solved only one task of medium difficulty his experience with medium difficulty tasks is low; if s/he has solved ten tasks of medium difficulty, his experience with medium difficulty tasks is high.

(c) arithmetics knowledge level, which is given and updated by the teacher.

The learner characteristics presented above were considered because they influence the learner's performance and the level of support they need. The arithmetics knowledge is particularly relevant when learners derive expressions from their constructions. Learners with lower arithmetics ability encounter difficulty at this stage, even if they have a conceptual understanding of the correspondence between their construction and expression; their difficulty lies in their understanding of the arithmetical language rather than the generalisation process.

The learner characteristics were not included as criteria in the hierarchy because of the exponential increase in the necessary pairwise comparisons. The pairwise comparisons, i.e. the c_{ij} and a_{ij} values mentioned in Section 5.2, are provided by experts. This is a very time consuming process and therefore, is limited by the experts availability and willingness to provide the pairwise comparisons. Therefore, we chose to build the mechanism with only one criterion, i.e. the context, and to evaluate it using scenarios that combine various values for the learner characteristics mentioned above. Although this approach is time consuming and requires significant involvement from experts, it has the advantage of closely mimicking the teacher's prioritisation decisions.

To illustrate the application of AHP for prioritising feedback we consider a situation when the task difficulty is high, the level of experience of the learner with highly difficult tasks is low and his/her arithmetics level is good. The pairwise comparisons in this example and in the evaluation were provided by one expert.

The pairwise comparisons and weights of the criteria for the above mentioned situation are displayed in Table 5.3. As in *ShapeBuilder* the learner is given a task of high difficulty only after having at least medium experience with low and medium task difficulty, the specific context is only slightly more important than the general one.

For each of the two criteria, i.e. specific and general context, there is a matrix of pairwise comparisons of the alternatives. The pairwise comparisons and the priority vector when the context is specific is displayed in Table 5.4, while the ones for the general context are displayed in Table 5.5. Both matrices are consistent.

Table 5.3: Criteria pairwise comparisons and weights.

Criteria	Specific	General	Weights
Specific	1	2	0.67
General	1/2	1	0.33
$\lambda_{max} = 2, CI = 0, CR = 0$			

Table 5.4: Alternatives pairwise comparisons and the priority vector with respect to specific context.

Alternatives	CC	CE	C-E	Sym	CGen	EGen	IV	Priority Vector
CC	1	3	3	1	7	7	7	0.33
CE	1/3	1	1/2	1/2	3	3	7	0.13
C-E	1/3	2	1	1/2	5	5	7	0.19
Sym	1	2	2	1	2	3	5	0.21
CGen	1/7	1/3	1/5	1/2	1	1	1/2	0.04
EGen	1/7	1/3	1/5	1/3	1	1	1/3	0.04
IV	1/7	1/7	1/7	1/5	2	3	1	0.05
$\lambda_{max} = 7.52, CI = 0.09, CR = 0.07$								

Table 5.5: Alternatives pairwise comparisons and the priority vector with respect to general context.

Alternatives	CC	CE	C-E	Sym	CGen	EGen	IV	Priority Vector
CC	1	1	1	1	2	2	1	0.16
CE	1	1	1/2	1/2	1/3	1/3	1/3	0.07
C-E	1	2	1	1	1/3	1/3	1/3	0.09
Sym	1	2	1	1	1/2	2	1/3	0.12
CGen	1/2	3	3	2	1	2	1/2	0.18
EGen	1/2	3	3	1/2	1/2	1	1/2	0.12
IV	1	3	3	3	2	2	1	0.26
$\lambda_{max} = 7.62, CI = 0.10, CR = 0.08$								

The final alternatives priorities are displayed in Table 5.6. The most important aspect to give feedback on is the correctness of construction (CC), followed by symmetry (Sym), correspondence between construction and expression (C-E) and use of icon variables (IV); the next aspects to give feedback on are: correctness of expression (CE), construction generality (CGen) and expression generality (EGen).

Therefore, when the task difficulty is high, the learners' experience with this type of tasks is low and their arithmetics level is good, the most important aspect is the correctness of construction and its symmetry; symmetry is quite important as the generalisation of a difficult task is more difficult when the

Table 5.6: Feedback priorities.

Alternatives	CC	CE	C-E	Sym	CGen	EGen	IV
Priorities	0.28	0.11	0.16	0.18	0.09	0.07	0.12
Overall $CR = 0.03$							

construction (and consequently the expression) is not symmetric. The correspondence between construction and expression is used in the development of the expression and is early encouraged so as to develop the expression in parallel to the construction; the use of icon variables is encouraged from the beginning in order to facilitate generalisation at a later stage. Hence, the focus is on the construction, guiding the learners to build it in a correct way, using symmetry and icon variables; the generality is addressed later, when these aspects are taken care of. Also, the importance of the correspondence between construction and expression is stressed as to facilitate the derivation of the expression.

5.3.2 Evaluation

To evaluate the proposed mechanism we use scenario-based validation (Lalioti and Theodoulidis, 1995). We analyse the situation of two learners, Alan and Mike, who have built the same construction, but have different expressions; also, their learner models hold different characteristics. The constructions and expressions of these hypothetical learners are based on data from real learners.

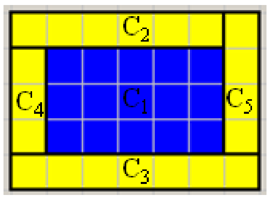


Figure 5.5: Combination of ‘I’ and ‘Spiral’ strategies (C_2 and C_3 are general; C_4 and C_5 are specific)

We will start with the common aspect between the two: the construction, which is displayed in Figure 5.5. It is a combination of two strategies, ‘I’ and ‘Spiral’, and has two general cases (C_2 and C_3) and two specific ones (C_4 and C_5). This is a symmetric situation and consequently, the comparison with the specific and the general strategies will give the same results. Moreover, because the surrounding has two cases from one strategy and two cases from the other one, when compared with these two strategies, the same result is obtained: 3.44. A piece of information that may be important when deciding towards which strategy to guide the learner is related to the order in which the surrounding is done and the complexity of the strategies: in the current example, the learners start with a case from the 'Spiral' strategy, but move to the 'I' strategy for the next two cases; to complete the surrounding, another case of from the 'Spiral' strategy is used. The fact that 2 subsequent cases belong to the same strategy may be a good reason to guide the learner towards that particular strategy; however, on its own, most of the time this cannot be considered a sign of coherence in the learners' thinking. Other actions of the learners and information stored in the learner models are necessary to reach an informed decision.

The expressions defined by Alan and Mike are displayed in Figure 5.6. Although Alan's construction has some generality (through the use of icon variables) the expression is still specific. Mike has some generality in the expression as well as the construction, but the two do not match.

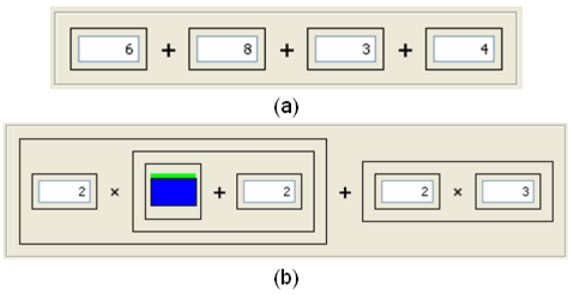


Figure 5.6: (a) Alan's expression (b) Mike's expression

The difficulty of the 'pond tiling' task is medium. On one hand, Alan has low experience with tasks of medium difficulty and he has a low level of arithmetics knowledge. On the other hand, Mike has high experience with medium difficulty tasks and his level of arithmetics is good. Taking these aspects into consideration and the context, i.e. specific and general, the AHP is applied to find in which order the feedback on the different aspects (the seven alternatives presented in Section 5.3.1) should be provided.

For Alan, the pairwise comparisons and the weights for the criteria (the two types of context) are displayed in Table 5.7. As his experience with medium difficulty tasks is low, the specific context criterion is more important that the general context one. The matrices for the alternatives are displayed in Table 5.8 and Table 5.9: the first for the specific criterion and the second for the general one. The final alternatives priorities are displayed in Table 5.10.

The two matrices of pairwise comparisons between alternatives are not consistent according to the standard measurement of consistency, i.e. CR, although the overall hierarchy is consistent (CR = 0.07). The revised alternatives ma-

Table 5.7: Alan: Criteria pairwise comparisons and weights.

Criteria	Specific	General	Weights
Specific	1	3	0.75
General	1/3	1	0.25
$\lambda_{max} = 2.00, CI = 0, CR = 0$			

Table 5.8: Alan: Alternatives pairwise comparisons and the priority vector with respect to specific context.

Alternatives	CC	CE	C-E	Sym	CGen	EGen	IV	Priority Vector
CC	1	5	5	3	9	9	9	0.44
CE	1/5	1	1/2	3	5	5	9	0.17
C-E	1/5	2	1	3	5	5	9	0.21
Sym	1/3	1/3	1/3	1	1/3	3	1/7	0.04
CGen	1/9	1/5	1/5	3	1	1	1/2	0.05
EGen	1/9	1/5	1/5	1/3	1	1	1/5	0.03
IV	1/9	1/9	1/9	7	2	5	1	0.07
$\lambda_{max} = 8.62, CI = 0.27, CR = 0.20$								

Table 5.9: Alan: Alternatives pairwise comparisons and the priority vector with respect to general context.

Alternatives	CC	CE	C-E	Sym	CGen	EGen	IV	Priority Vector
CC	1	1	1	2	2	2	1	0.16
CE	1	1	1	2	1/5	1/7	1/5	0.06
C-E	1	1	1	2	1/5	1/7	1/5	0.06
Sym	1/2	1/2	1/2	1	1/3	1/3	1/7	0.05
CGen	1/2	5	5	3	1	2	1/2	0.20
EGen	1/2	7	7	3	1/2	1	1/2	0.18
IV	1	5	5	7	2	2	1	0.30
$\lambda_{max} = 7.98, CI = 0.16, CR = 0.12$								

Table 5.10: Alan: Feedback priorities.

Alternatives	CC	CE	C-E	Sym	CGen	EGen	IV
Priorities	0.37	0.14	0.17	0.04	0.08	0.07	0.12
Overall $CR = 0.07$							

trices are displayed in Table 5.11 and Table 5.12. The new final priorities are displayed in Table 5.13.

The final order of alternatives is the same in the revised version as in the first (apparently) inconsistent one, although there are small differences in the values. The revised matrices were obtained using Saaty's (Saaty, 1980) suggested method of reconsidering the alternatives for which the corresponding rows in the matrix $[|a_{ij} - (p_i/pj)|]$ have the largest sums. Looking at the original and the new matrices, a change in numbers is observed, but with the preservation of the transitivity between the alternatives. As the idea is to prioritise between

Table 5.11: Alan: Alternatives pairwise comparisons and the priority vector with respect to specific context - revised.

Alternatives	CC	CE	C-E	Sym	CGen	EGen	IV	Priority Vector
CC	1	4	4	5	9	9	9	0.44
CE	1/4	1	1/2	3	5	5	7	0.17
C-E	1/4	2	1	5	5	5	7	0.22
Sym	1/5	1/3	1/5	1	1/2	2	1/5	0.04
CGen	1/9	1/5	1/5	2	1	1	1/2	0.04
EGen	1/9	1/5	1/5	1/2	1	1	1/3	0.03
IV	1/9	1/7	1/7	5	2	3	1	0.06
$\lambda_{max} = 7.81, CI = 0.14, CR = 0.10$								

Table 5.12: Alan: Alternatives pairwise comparisons and the priority vector with respect to general context - revised.

Alternatives	CC	CE	C-E	Sym	CGen	EGen	IV	Priority Vector
CC	1	1	1	2	2	2	1	0.18
CE	1	1	1	2	1/3	1/3	1/3	0.09
C-E	1	1	1	2	1/3	1/3	1/3	0.09
Sym	1/2	1/2	1/2	1	1/2	1/2	1/3	0.07
CGen	1/2	3	3	2	1	2	1/2	0.18
EGen	1/2	3	3	2	1/2	1	1/2	0.15
IV	1	3	3	3	2	2	1	0.25
$\lambda_{max} = 7.55, CI = 0.09, CR = 0.07$								

Table 5.13: Alan: Feedback priorities - revised.

Alternatives	CC	CE	C-E	Sym	CGen	EGen	IV
Priorities	0.37	0.15	0.19	0.05	0.08	0.06	0.11
Overall $CR = 0.04$							

alternatives, transitivity is a logical condition and maybe the only necessary indication of consistency, as the preservation of priorities order seems to indicate.

Coming back to the educational aspect related to these priorities, for Alan, the most important aspect for feedback is the correctness of construction, followed by the correspondence between construction and expression, and the correctness of expression. Thus, as Alan's arithmetics knowledge is low, a correct construction (already achieved, although not symmetric) and its correspondence with the expression (partial correspondence) is used to guide Alan towards a correct expression. The following aspects to give feedback on are: (a) the use of icon variables - thus, Alan will be encouraged to introduce the icon variables in his expression and also to use the icon variable for the the tiles corresponding to cases C_4 and C_5; (b) the generality of the construction, which can be achieved through using icon variables; (c) expression's generality and (d) symmetry. Thus, Alan will be guided to achieve a correct and general construction and expression before introducing symmetry. After this has been achieved, the benefits of symmetry will be presented and Alan will be guided towards the 'Spiral' or 'I' strategy through one of the following options: (i) present the learner with the two options and let him/her choose one of the two (an approach that appears more suitable for advanced learners than novices); (ii) automatically suggest one of the two in a systematic way, e.g. present the one that occurs more/less often with other learners; (iii) inform the teacher about the learners trajectory and the frequency of strategies and let him/her decide between the two. Letting the learner choose is not an feasible option, as Alan has low experience with this type of tasks. In case of an automatic suggestion, Alan would probably be guided towards the 'I' strategy, as this is used more often by other learners than the 'Spiral' strategy. Alternatively, the teacher could be informed about Alan's activity and s/he could choose one of the two.

The pairwise comparisons and weights of the criteria for Mike are displayed in Table 5.14. As Mike's level of experience with medium difficulty tasks is high, the general context criterion is significantly more important than the specific context one. The alternatives matrices are displayed in Table 5.15 for the specific context criterion and Table 5.16 for the general context criterion. The final alternatives priorities for Mike are displayed in Table 5.17. All matrices are consistent.

Table 5.14: Mike: Criteria pairwise comparisons and weights.

Criteria	Specific	General	Weights
Specific	1	1/5	0.17
General	5	1	0.83
$\lambda_{max} = 2.00, CI = 0, CR = 0$			

Table 5.15: Mike: Alternatives pairwise comparisons and the priority vector with respect to specific context.

Alternatives	CC	CE	C-E	Sym	CGen	EGen	IV	Priority Vector
CC	1	3	3	2	5	5	5	0.35
CE	1/3	1	1/2	1	3	3	5	0.15
C-E	1/3	2	1	1/2	3	3	5	0.17
Sym	1/2	1	1/2	1	3	3	5	0.16
CGen	1/5	1/3	1/3	1/3	1	1	1/3	0.05
EGen	1/5	1/3	1/3	1/3	1	1	1/7	0.04
IV	1/5	1/5	1/5	1/5	3	7	1	0.07
$\lambda_{max} = 7.51, CI = 0.09, CR = 0.06$								

Table 5.16: Mike: Alternatives pairwise comparisons and the priority vector with respect to general context.

Alternatives	CC	CE	C-E	Sym	CGen	EGen	IV	Priority Vector
CC	1	1	1	2	1/2	1	1	0.13
CE	1	1	1/2	1/2	1/3	1/3	1/3	0.07
C-E	1	2	1	1/2	1/3	1/3	1/3	0.08
Sym	1/2	2	2	1	1	3	1/3	0.14
CGen	2	3	3	1	1	2	1	0.22
EGen	1	3	3	1/3	1/2	1	1/3	0.12
IV	1	3	3	3	1	3	1	0.24
$\lambda_{max} = 7.59, CI = 0.10, CR = 0.07$								

Table 5.17: Mike: Feedback priorities.

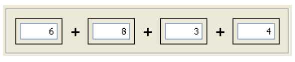
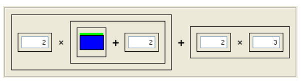
Alternatives	CC	CE	C-E	Sym	CGen	EGen	IV
Priorities	0.17	0.08	0.10	0.15	0.19	0.11	0.21
Overall $CR = 0.03$							

For Mike, the most important aspect for feedback is the use of icon variables, aiming to introduce generality early in the process of solving the task as Mike has high experience with similar tasks (from difficulty point of view). Thus, Mike will be encouraged to use icon variables for the tiles corresponding to cases C_4 and C_5. The second important aspect, construction generality, is done using the icon variables and only after that feedback should address the correctness of construction. Thus, as opposed to Alan, Mike is first guided towards 'thinking in a general way' and then 'fixing' the construction/expression. After generality of construction, the following important aspect is symmetry, as it will facilitate the generalisation and the construction of the corresponding general expression, which is the next aspect for feedback. At this point, any eventual mismatch between the construction and expression would be addressed and through this, Mike would be guided towards a correct expression. Although Mike's expression did not correspond to his expression, this aspect is not addressed immediately; also, the expression shows a way of solving the task that is symmetric, which will probably be reflected in the construction on the following actions.

A summary of the integrated example is displayed in Table 5.18, including the input from the Task LTM and Domain LTM, and the output from the Feedback Priorities component. Thus, a similar diagnosis of the learners models obtained with the CBR mechanism leads to different outputs from the AHP method depending on extra information about the learners.

Although this evaluation was limited to two learners, it provided sufficient evidence that the prioritisation provided by the AHP mechanism is meaningful from pedagogical point of view. At this point, we were faced with a decision of spending more time on evaluating this mechanism or using the time for updating it in accordance with the new version of the system and evaluating the new

Table 5.18: Feedback priorities depending on the learners' model.

		Alan	Mike
Task LTM	task	pond-tiling	pond-tiling
	difficulty	medium	medium
	strategies	'I' 'Spiral'	'I' 'Spiral'
	expression		
Domain LTM	experience	low	high
	arithmetics level	low	good
Feedback priorities		CC, C-E, CE, IV, CGen, EGen, Sym	IV, CGen, CC, Sym EGen, C-E, CE

version. As we felt that the evaluation of this version was encouraging albeit its small scale, we decided to update the mechanism in line with the new system and to evaluate the new version.

The following section presents the second iteration of this mechanism in the context of *eXpresser*. Similarly to this section, the AHP formalism is presented in the context of *eXpresser* and evaluated through the use of scenarios.

5.4 Feedback Prioritisation - Iterative Version 2

This section presents a second version of the feedback prioritisation mechanism tailored for *eXpresser*. Two modifications occurred in line with the changes from *ShapeBuilder* to *eXpresser*: the icon variables were replaced by T-boxes, the symmetry of construction was replaced by a more general property of construction, i.e. being built in a systematic way, and a new criterion was added, i.e. the learning mode - individual or collaborative. The new criterion was added based on the envisaged use of *eXpresser* in a collaborative mode in which two learners with similar approaches are paired to help them both reach a common general solution. A new learner criterion was added, i.e. preferred approach (specific to general or general to specific), as these two different approaches were observed in studies with pupils. In the following the AHP formalism is presented together with its evaluation using scenarios.

5.4.1 Feedback Prioritisation for *eXpresser*

Three scenarios using the 'stepping stones' task are presented to illustrate and validate the AHP process in the context of *eXpresser*. The hierarchy of the AHP formalism is illustrated in Figure 5.7: the goal is to obtain feedback priorities; the criteria is the learning mode, i.e. individual or collaborative, and the stage in a task, i.e. specific and general.

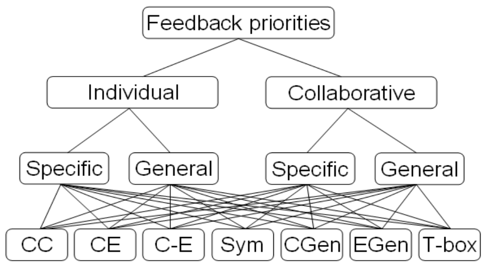


Figure 5.7: AHP hierarchy.

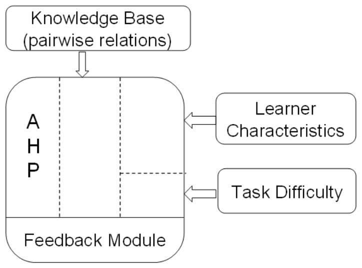


Figure 5.8: Feedback module.

The alternatives are feedback on the following aspects, which are mostly the same as the ones in Section 5.3.1 (except the two changes mentioned previously):

(a) correctness of construction (CC);

(b) correctness of expression (CE);

(c) construction-expression correspondence (C-E);

(d) systematic approach in building the construction (Sys);

(e) generality of construction (CGen);

(f) generality of expression (EGen);

(g) use of T-boxes (T-box).

The pairwise comparisons between criteria and between alternatives vary depending on learner's (dynamic) characteristics:

(a) level of experience (stored for each level of task difficulty),

(b) arithmetics knowledge level and

(c) preferred approach: from specific to general (S-to-G) or from general to specific (G-to-S).

The first two are the same as in Section 5.3.1 and the third has been introduced because the two approaches were observed in classroom trials.

The feedback module (Figure 5.8) integrates this information together with information about task difficulty to retrieve sets of pairwise relations from the Knowledge Base. This generates different instantiations of the AHP process. To illustrate how AHP is going to operate in different situations, three scenarios are considered below (summarised in Table 5.19).

Table 5.19: Scenarios characteristics.

Characteristics	Scenario 1	Scenario 2	Scenario 3
Mode	individual	individual	collaborative
Task difficulty	medium	medium	medium
Experience	low	medium	low&medium
Arithmetics	high	low	high&low
Approach	G-to-S	S-to-G	G-to-S&S-to-G

These particular scenarios were used to illustrate that the AHP mechanism provides adequate prioritisations based on the personal characteristics of learners. In other words, these scenarios show that the AHP approach delivers personalised prioritisations that are relevant for the particular combinations of learner characteristics. Moreover, two learners with the same strategy were chosen to illustrate that the AHP mechanism provides different prioritisation for different learner characteristics even when they used the same strategy. Also, when these two learners work together in a collaborative mode, the output of the AHP mechanism reflects the combination of characteristics of the two learners, thus leading to a prioritisation that is different from the individual mode.

5.4.2 Evaluation

The constructions for the scenarios are displayed in Figure 5.9: *Constructions 1* and *2* are used in Scenario 1 and 2, respectively. In the collaborative scenario, i.e. Scenario 3, the learners who produced these constructions and their corresponding expressions are working together. The pairing for collaboration is based on the similarity of their strategies presented in Section 4.4.2 and the complementarity of approach and/or arithmetic level. A diagnosis of the learners' constructions (see Section 4.4.3) is carried out at the same time with the computation of feedback priorities. Combining these two sources, a decision is taken with regard to necessary and/or relevant feedback. The pairwise comparisons for all scenarios were provided by one expert.

The components of Construction 1 are displayed separately for ease of visualisation; this construction has four patterns: (a) two compact rows of green (lighter colour) tiles and (b) two rows with gaps in between tiles: one green and one red (darker colour). The first two mentioned are the same, and consequently, have the same properties displayed in the property list of the highlighted row in *Construction 1*. The first property, i.e. number of iterations, shows that the pattern depends on the red one because the number of iterations of the green tiles is set to 'the number of red tiles multiplied by 2 plus 1'. *Construction 2* is built in a similar fashion, but the compact rows of green tiles do not depend on the red pattern: the first property (number of iterations) from the property

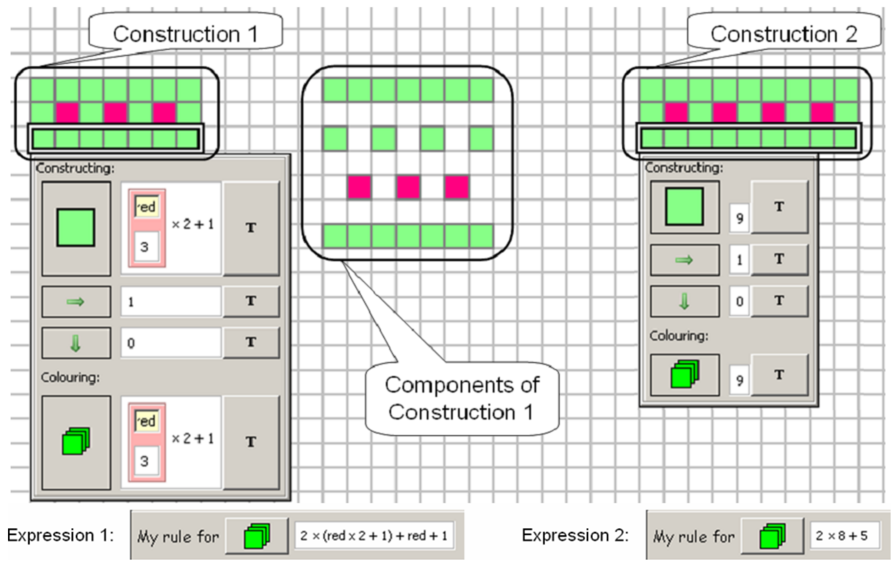


Figure 5.9: The constructions and expressions of two learners using the 'H' strategy in solving the 'stepping stones' task; the components of Construction 1 are also presented separately for ease of visualisation.

list is set to 9. At the bottom of Figure 5.9, two expressions corresponding to the two constructions are displayed. *Expression 1* uses the name red for the number of red tiles, while *Expression 2* is entirely numeric.

In the constructions of Figure 5.9, both learners follow the same strategy in surrounding the stepping stones: two rows of tiles at top and bottom, and one row of tiles in the gaps of the red pattern; also, for both constructions, the row of green tiles with gaps in between (the middle one) does not depend on the red pattern and the expressions do not match their corresponding constructions. There are, however, a few differences: (a) they work with a different number of red tiles, i.e. 3 and 4, respectively; (b) the first learner is very close to a general solution, while the second is still working with the particular case of 4 red tiles; (c) the expression of the first learner (*Expression 1* in Figure 5.9) is already general, while the expression of the second learner (*Expression 2* in Figure 5.3) is numeric.

Scenario 1. Feedback prioritisation is established by taking into consideration: (a) the individual learning mode, (b) the learner's characteristics mentioned in Table 5.19 and (c) *Construction 1* and *Expression 1* from Figure 5.9. The criteria pairwise comparison, the corresponding weights and consistency information are displayed in Table 5.20; the alternatives pairwise comparison with respect to the criteria (specific and general context), the priority vectors and the consistency measures are displayed in Table 5.21 and Table 5.22. The final priorities and the overall consistency are displayed in Table 5.23. From these tables, the numbers assigned by the designer of the AHP component are the criteria and the alternatives pairwise comparisons; the rest are computed using the formulas presented in Section 5.3.

Table 5.20: Criteria pairwise comparison, weights, and consistency.

Criteria	Specific	General	Weights
Specific	1	1/2	0.33
General	2	1	0.67
$\lambda_{max} = 2.00$, $CI = 0$, $CR = 0$			

Table 5.21: Alternatives pairwise comparison, priority vector with respect to the specific context, and consistency.

Alternatives	CC	CE	C-E	Sys	CGen	EGen	T-box	Priority vector
CC	1	2	2	5	2	3	1/2	0.22
CE	1/2	1	1/2	3	1/5	1/2	1/2	0.08
C-E	1/2	2	1	3	1/5	2	1/2	0.11
Sys	1/5	1/3	1/3	1	1/3	1/3	1/3	0.04
CGen	1/2	5	5	3	1	3	1/2	0.22
EGen	1/3	2	1/2	3	1/3	1	1/2	0.09
T-box	2	2	2	3	2	2	1	0.23
$\lambda_{max} = 7.75$, $CI = 0.13$, $CR = 0.10$								

Table 5.22: Alternatives pairwise comparison, priority vector with respect to the general context, and consistency.

Alternatives	CC	CE	C-E	Sys	CGen	EGen	T-box	Priority vector
CC	1	5	5	7	1	5	1/2	0.27
CE	1/5	1	1/2	7	1/5	2	1/3	0.08
C-E	1/5	2	1	3	1/3	3	1/3	0.10
Sys	1/7	1/7	1/3	1	1/3	3	1/3	0.05
CGen	1	5	3	3	1	2	1/2	0.19
EGen	1/5	1/2	1/3	1/3	1/2	1	1/5	0.04
T-box	2	3	3	3	2	5	1	0.27
$\lambda_{max} = 7.82$, $CI = 0.14$, $CR = 0.10$								

Table 5.23: Scenario 1: Feedback priorities and overall consistency.

Alternatives	CC	CE	C-E	Sys	CGen	EGen	T-box
Priorities	0.25	0.08	0.10	0.05	0.20	0.06	0.26
Overall $CR = 0.04$							

As the learner prefers the general-to-specific approach, the top item for feedback is *T-boxes* as they allow general constructions. The next two items to give feedback on are the correctness of construction and its generality. *Construction 1* has two general components and a specific one, which indicates that the learner has already used T-boxes, so no feedback on that is necessary; as the construction is correct, the first feedback to be provided will be on the generality of the construction, and more specifically, on the generality of the only specific component of the construction. From the AHP process, the next priorities are related to the expression: correspondence between construction and expression, correctness of expression and expression generality. The last two items are already in place, so no feedback on them is given. If in the previous step the learner has made the specific component general, the construction would correspond to the expression; if not, feedback would be provided to the learner to make sure the construction (partially general) corresponds to the expression (general).

Scenario 2. In this scenario, the prioritisation is computed for the individual learning mode, taking in consideration the learner's characteristics displayed in Table 5.19, *Construction 2* and its corresponding expression from Figure 5.9. The procedure is applied as in Scenario 1; only the final feedback priorities and the overall consistency are reported in Table 5.24.

Table 5.24: Scenario 2: Feedback priorities and overall consistency.

Alternatives	CC	CE	C-E	Sys	CGen	EGen	T-box
Priorities	0.310	0.108	0.130	0.194	0.092	0.060	0.106
Overall $CR = 0.03$							

As the learner prefers a specific-to-general approach, the feedback addresses generality at the end. The first aspects to give feedback on are: the correctness of construction, its systematic approach and the correspondence between construction and expression. The first two aspects are in place, so the feedback would be given on the correspondence between expression and construction. If the learner addresses this aspect and the new expression is $2*9+5$, the feedback on the following item, i.e. correctness of expression, becomes unnecessary. If the learner does not correct the expression accordingly, the feedback would address the correctness of expression, pointing out that the construction is correct and that the expression should correspond to the construction. So, feedback at this point includes the two interrelated aspects: the correctness of expression and the correspondence between construction and expression. Only after establishing the correctness of construction and expression for the specific case of 4 red tiles, the feedback will address the generality of the construction: the use of T-boxes, the generality of construction, and, finally, the generality of expression.

Scenario 3. In the collaborative mode, the two learners are working together towards finding a general solution. The first leaner has a construction with 3 red tiles, while the second has a construction with 4 red tiles. Consequently, a specific approach on one side will lead to an inadequate construction on the other, which enforces the learners to work with the general. The procedure is applied as in Scenario 1, i.e. the pairwise comparisons, the weights for the criteria and the priority vectors for each criterion are defined for this collaborative situation. The feedback priorities are displayed in Table 5.25.

Table 5.25: Scenario 3: Feedback priorities and overall consistency.

Alternatives	CC	CE	C-E	Sys	CGen	EGen	T-box
Priorities	0.15	0.08	0.11	0.24	0.17	0.05	0.19
Overall $CR = 0.03$							

As the learners are 'forced' to work with the general, the first aspect to give feedback on is the systematic approach of the construction as, otherwise, it would be difficult to make it general – as both learners have systematic (and symmetric) constructions, this is not necessary. The next aspect to give feedback on is the use of T-boxes; ideally, this feedback from the system would be replaced by the feedback of learner one, who has a partially general construction, to learner two, who has a specific construction. The next two aspects to be addressed are the generality and the correctness of the construction. For the same reason mentioned previously, the construction will be correct only when it is general, so generality is addressed first and correctness afterwards. The expression is dealt with at the end, starting from the correspondence with the construction, addressing its correctness and finally, its generality.

The priorities delivered by the AHP process were validated by two experts in the field of mathematical education who were aware of the way learners interacted with *eXpresser*. Both of them agreed on the prioritisation for the two individual situations, but there was one disagreement on the collaborative scenario. One expert agreed with the prioritisation delivered by the AHP process, while the other argued for the following order: T-box, CGen, CC, Sys, C-E, CE and EGen. This order differs from the output of the AHP process by the fact that the systematic approach is moved from the first place to the fourth. The second expert's argument for this was that they could build a construction that is correct and not necessarily systematic. On the other hand, the other expert argued that a systematic approach is important from the very beginning to facilitate the generality of construction (and then, the expression) because one of the learners prefers the specific-to-general approach and also has a low arithmetics ability; therefore, even if the other learner would be able to reach a general construction (though non-systematic) and to find a corresponding expression, for the other learner this would be difficult and hardly beneficial.

Because the AHP mechanism is based on the pairwise comparisons, in case of disagreement, the pairwise comparisons can be fine tuned and checked for consistency, thus allowing to change the prioritisation output. Providing the teachers/researchers with an easy interface for making such changes is a direction of future research that could improve our proposed mechanism.

5.5 Discussion

For both versions of the feedback prioritisation mechanism one expert provided the pairwise comparisons, but the output of the prioritisation mechanism was checked against two other experts only in the second version. For this latter version two individual and one collaborative scenario were used. The feedback priorities for the individual mode were confirmed by both experts, whilst the priority given to the *systematic* approach in the collaborative mode was considered by one of the experts as too high. One possible explanation for the diversity of the experts' opinion could be the added complexity of the collaborative mode, which is an issue that requires further investigation. The scale of the evaluation was small due to the time and effort needed from experts to provide pairwise comparisons and to evaluate the outcomes of the mechanism. Thus, although the mechanism was proven to deliver meaningful prioritisations for the three scenarios considered, more evaluation studies should be conducted using different combinations of learner characteristics before we can claim that the AHP approach works in general.

As already mentioned above, a considerable limitation of the AHP-based approach is that in some applications a large number of pairwise comparisons might be needed. To alleviate this situation in *eXpresser* we decided to use a small number of learner characteristics combinations. If a combination of learner characteristics has no corresponding pairwise comparisons, the closest combination could be used instead. For example, if the current combination is individual mode, low task difficulty, medium experience, medium arithmetics level and a general-to-specific preferred approach, the closest combination would be individual mode, low task difficulty, medium experience, high arithmetics level and a general-to-specific preferred approach. This, however, may not lead to the best prioritisation and may potentially confuse the learner. Further investigation is needed to evaluate the trade off between keeping the number of pairwise comparisons low and delivering relevant prioritisations. To address this limitation future work will look into modelling the AHP process using neural networks.

5.6 Summary and Contribution of the Chapter

In this chapter a brief overview of personalised feedback was presented, together with a proposed approach to feedback prioritisation based on a method from Multi-criteria Decision Making called the Analytic Hierarchy Process. This method was presented in detail and examples on how it is used were given for both versions of the feedback prioritisation mechanism. The second version was evaluated by two experts who agreed on the output of the proposed mechanism with a small exception.

The contribution of this chapter is the development of a mechanism for feedback prioritisation based on the Analytic Hierarchy Process. Although designed for a particular educational environment, this approach could be applied in other educational systems. The criteria and alternatives, however, would need to be replaced with relevant aspects for the particular environment and pairwise comparisons would need to be provided.

Feedback prioritisation is an issue teachers deal with in everyday teaching, when learners need feedback on several aspects. Let us take the example of a very simple task in the domain of algebra, where a learner is given a task in which the first step is to define an algebraic expression. When a learner has two mistakes in his expression, the teacher decides which one to address first. In computer-based environments, however, this issues does not seem to be considered at all.

In the context of structured learning environments, the material is broken into steps that address only one aspect. Therefore, in such environments, feedback prioritisation is avoided altogether by the way the material is structured. In ELEs, however, there is less structure and the learners are allowed more freedom, which leads to situations similar to the ones in classroom teaching, where the teacher needs to assess what the learner has done and which aspect to address first. Previous research in the area of ELEs, however, does not seem to address this issue either.

In classroom situations, teachers assess what a learner has done and what aspects s/he needs feedback on very quickly based on their domain and pedagogical expertise (Marzano et al., 2005). In our research, the diagnosis is done by the learner modelling mechanism presented in Chapter 4, while the prioritisation is based on an Analytic Hierarchy Process method which takes into consideration domain and task knowledge, and pedagogical expertise. In other words, our approach emulates teachers' behaviour to address the issues they encounter in classrooms and embed it into our exploratory learning environment. Therefore, the main contribution of the research presented in this chapter is addressing an important issue, i.e. feedback prioritisation, that has not been addressed before, and attempting the developing of a prioritisation mechanism for our exploratory learning environment.

This AHP mechanism can not be easily applied to other exploratory learning environments, as it has been developed using domain-specific information. Thus, to be applied to other environments, the corresponding domain-specific information needs to be identified, i.e. the criteria and the alternatives. Although this approach seems to provide meaningful prioritisations that mimic the teacher's decisions, currently the effort needed for developing this mechanism outweighs the benefits.

Chapter 6

Grouping for Collaborative Learning Activities

In exploratory learning the focus is more on the knowledge construction process rather than knowledge itself. Students learning emerges during the construction and (or) exploration of models by varying their parameters and observing the effects of these variations (Klahr, 2000). Exploratory learning is particularly useful for problems that are not well structured and with no clear boundaries between correct and incorrect approaches to solve a task (Lynch et al., 2006). Moreover, some problems have several valid solutions but none of them can be considered better than the others, and a full understanding of the knowledge domain often requires awareness of the various ways the problems could be solved. In classroom situations, this understanding is developed during collaborative activities in which learners discuss similarities and differences between their approaches (Lynch et al., 2006).

Successful collaboration for learning activities, however, depends on forming groups in which the activity is relevant for all members of the group. This chapter presents a mechanism inspired from Group Technology (Selim et al., 1998) for grouping learners for collaborative activities that allows to form groups of learners based on the similarity of their approaches to the same task. The next section introduces the problem of grouping in the context of exploratory learning and gives an overview of general criteria considered for collaboration in learning environments. The following section gives a brief overview of Group Technology, while the subsequent one presents the proposed approach for grouping in the context of *eXpresser*. This is followed by an evaluation using classroom scenarios, a discussion, conclusions and an outline of the contribution of the chapter.

6.1 Grouping for Collaborative Learning Activities

Although collaborative learning has been proved successful in classroom situations (Brown and Palincsar, 1989; Slavin, 2003), in computer-supported learning environments it does not seem to lead to the same learning benefits. One contributing factor is the way the collaborative groups are formed, as forming efficient groups is very important to ensure an educational benefit from the group interaction (Daradoumis et al., 2002).

For problems that could be solved in several ways, the aim of collaborative activities is to discuss similarities and differences between various approaches followed by students when working individually. Often, establishing similarities and differences between various approaches involves translating between multiple representations. In simulation-based learning environments, the use of multiple representations is considered beneficial as it leads learners to deeper knowledge acquisition of a domain, that in turn, could lead to knowledge transfer in other learning situations (van der Meij and de Jong, 2006). Also, "having to make the mental transference between representations ... forces reflection beyond the boundaries and details of the first representation and an anticipation of correspondences in the second. The deeper level of cognitive processing can reveal glitches that might otherwise have been missed" (Petre et al., 1998, p. 474).

Multiple External Representations (MERs) (as opposed to mental representations) have several functions: to complement, constrain and construct (Ainsworth, 1999). "The first function is to use representations that contain complementary information or support complementary cognitive processes. In the second, one representation is used to constrain possible (mis)interpretations in the use of another. Finally, MERs can be used to encourage learners to construct a deeper understanding of a situation" (Ainsworth, 1999, p. 134).

One of the contributing factors to successful collaboration is the formation of groups in a way that each group member will benefit from the collaborative interaction. The criteria for successful grouping, however, are still not well established, not even for classroom collaborative activities. The tendency is to group students based on their achievement (Macintyre and Ireson, 2002) and form heterogeneous or homogeneous groups with the aim to reduce heterogeneity of learning or of social behaviour (Gregory, 1984). It has been shown that low-achieving students learn more in heterogeneous groups than in homogeneous ones, and that high-achieving students benefit equally from heterogeneous and homogeneous groups (Webb et al., 1997). Besides these findings, there is little known about the influence of group formation on the collaborative processes and performance (Webb et al., 1997; Leornard, 2001).

In computer-supported learning, the criteria used for group formation are learners' characteristics related to their performance and social characteristics (e.g. Gogoulou et al. (2007); Graf and Bekele (2006)) and various approaches have been proposed to ensure the formation of optimal groups. For example, an algorithmic approach has been used in Gogoulou et al. (2007) to form homogeneous and heterogeneous groups, and a Genetic Algorithms approach was used to form mixed groups by considering the following learners' characteristics: learner's personality, competence level, learning style, an indicator for collaborative behaviour and an indicator for acting as evaluator in peer-assessment. An Ant Colony Optimisation approach is used in Graf and Bekele (2006) to form heterogeneous groups based on personality traits and performance. Fuzzy C-Means clustering for formation of homogeneous and heterogeneous groups is compared to other low complexity algorithms in Christodoulopoulos and Papanikolaou (2007). A framework for collaborative group formation was proposed in Ounnas et al. (2009) using a constraint satisfaction problem formulation, and Logic Programming and Semantic Web Technologies. A high-level description of the processes involved in group formation was proposed for virtual learning environments where learners do not know each other and several parameters that influence group collaboration were identified : (a) individual and group learning, and social goals; (b) relationships among group members; (c) the interaction process; and (d) members' specific characteristics (Daradoumis et al., 2002). To ensure formation of optimal groups, dynamic grouping supported by wireless handhelds has been proposed for classroom use, allowing reconfigurations of groups to find optimal ones (Zurita et al., 2005). Consequently, current research indicates that group formation is a complex problem with no straightforward answer.

Unlike previous work on group formation, our research is in the field of exploratory learning environments, where a problem can be solved using different strategies. The goal of the collaborative activities is to discuss similarities and differences between strategies adopted by learners to solve the task. Therefore, the grouping mechanism needs to take into consideration the individual approaches and the similarities/differences between the approaches followed by learners. To this end, modelling individual learners' approaches is required, together with a systematic way for defining similarities between approaches.

To address the challenging task of modelling individual behaviour, a case-based reasoning approach was presented in Chapter 4. Simple and composite cases were defined, where composite cases stand for the various approaches to solve a task, called strategies, and are defined as a series of simple cases linked through temporal and dependency relations. Each task has a Task Model consisting of composite cases; while solving a task, learners' behaviour is compared to these cases to identify the ones used by each learner, and this information is then stored in Learner Models. This formulation allows the recognition of strategies followed by learners *during* the task, rather then at the end, allowing thus to form groups any time the teacher wants to. Moreover, it provides a systematic way of defining similarities between strategies based on the existence of common cases. Lastly, it offers teachers flexibility to set their own preferences about the operation of the similarity detection mechanism by biasing group formation towards their personal grouping preferences, e.g. the teacher may want to group together students whose strategies have at least 3 cases in common or students who have largely followed different exploration strategies with just 1 case in common.

To take into consideration the similarity between different approaches followed by learners, we use an approach inspired from Group Technology (Selim et al., 1998) by defining learners and strategies as property vectors, based on the information from individual Learner Models and from the Task Model, calculating similarities between them and deriving an incidence matrix on which clustering is performed. This procedure leads to formation of homogeneous groups; however, heterogeneous groups can also be formed by choosing one or more learners from each or some of the homogeneous groups.

Unlike previous research, we propose characteristics of individual approaches and similarities between approaches as criteria for group formation, as these aspects are relevant for the way learning activities are defined using *eXpresser*. Although the goals of a task have been considered in group formation, we are not aware of any research that considers our proposed criteria. We are aware of only one approach based on Group Technology that has been previously used for group formation for educational purposes. This work was developed for distance learning environments with the aim to group learners and learning objects so that students are given the opportunity to learn skills or knowledge that they do not master already (Pollalis and Mavrommatis, 2009). Also, characteristics of solutions have been previously used to support collaboration (de los Angeles Constantino-Gonzalez et al., 2003); more specifically, differences between problem solutions are detected to help students recognise and resolve conflicts between their problem solutions. In contrast to these works, our approach is developed for an exploratory environment, where users are left free to follow any strategy they wish to solve a problem, making it difficult during task execution to distinguish between an approach that could lead to an appropriate solution and an incorrect one. Moreover, it uses similarities rather than differences between approaches to solve the task.

6.2 Group Technology

In order to incorporate our desired criteria in the grouping mechanism, we looked for a method that can perform grouping based on the defined criteria, as well as allowing flexibility about the definition of one of the criteria. Thus, we wanted to take into consideration the strategies used by the learners and the similarities between the various strategies used for the same task. Moreover, we are interested in a flexible way of defining similarities between strategies, as this has an impact on the difficulty of the collaborative task and also is viewed differently by different teachers (this is discussed in more detail in Section 6.4). Therefore, a mechanism that can handle this flexibility was required. In our search of the literature for different grouping approaches we came across a method called Group Technology which seemed promising for solving our group formation problem.

Group Technology (Selim et al., 1998) designates a method for group formation of machines and parts in cellular manufacturing systems. The idea is to optimise production of families of parts by creating machine cells. Thus, clusters of machines can be located in close proximity and be responsible for a particular family of parts to minimise production/transfer time. Therefore, two types of groupings are considered: grouping parts into families and grouping machines into cells. This is referred to as the cell formation problem. Several methods have been used to solve it, such as descriptive procedures, cluster analysis, graph partitioning, artificial intelligence and mathematical programming (Selim et al., 1998; Joines et al., 1996). We are interested in the clustering analysis approach, which in turn includes several techniques: similarity coefficient, set theoretic, evaluative and other analytic methods (King and Nakornchai, 1982). Our approach is using array-based clustering and the similarity coefficient technique.

In array-based clustering, a part-machine incidence matrix is used whose entries are either zero or one. If the entry in row i and column j is one, part j needs to visit machine i; if it is zero, no visit is needed. The array-based techniques lead to clusters of parts and machines by rearranging the order of rows and columns to form diagonal blocks of one in the part-machine matrix.

The similarity coefficient techniques involve grouping parts and machines into cells based on similarity measurements between machines and parts, and between machines. These similarity coefficients are used in agglomerative clustering techniques, of which the most popular is single linkage clustering (Selim et al., 1998).

In our approach, similarity coefficients are used to form the incidence matrix and array-based clustering is then applied to obtain the cells. For us, the parts correspond to learners, the machines correspond to exploratory strategies employed to perform a task, and the goal is to group students according to the strategy (or strategies) they used, taking into account that some strategies are similar to each other while others are different. Consequently, we have developed a mechanism based on Group Technology under the assumption that this approach is appropriate for our problem and that it can be applied in this new context, i.e. learning environments.

6.3 Grouping for Collaboration using *eXpresser*

In the context of *eXpresser* multiple representations are used at two levels: (a) at strategy level, where the same construction can be built using different structures and (b) at the T-boxes level which can be represented as numbers only, name only or both. As mentioned in Chapter 4, through their multiple representation, the T-boxes have the role of scaffolding the route from numbers to variables. The aim of collaborative activities is to reflect on the equivalence of seemingly different constructions and expressions and observe the similarity at a higher structural level as the essence of generalisation. These collaborative activities would benefit learners by raising their awareness of the several ways to approach the same task, and their effort to establish the equivalence of representations for both visual patterns and variables expressed with T-boxes will lead to deeper understanding (van der Meij and de Jong, 2006; Petre et al., 1998; Ainsworth, 1999).

Fig. 6.1 illustrates two seemingly different constructions and expressions for the 'pond tiling' task. The strategy used to solve the task is the same, i.e. the 'H' strategy, but the visual representations of the construction and the representation of the variables involved in the task (the T-boxes) seem different. The learners' discussion aims to establish the equivalence of these representations by recognising what is different, e.g. the dimensions of the pond, the names used for variables, and what is the same and captures the essence of generalisation, e.g. the structure used (the 'H' strategy), the expressions.

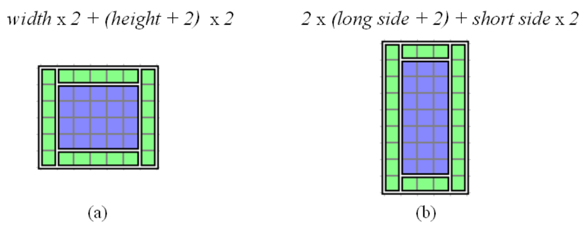

width x 2 + (height + 2) x 2
2 x (long side + 2) + short side x 2
(a)
(b)

Figure 6.1: Two approaches using the 'H' strategy.

Translating between representations, however, is found difficult by students (Ainsworth, 1999; Schoenfeld et al., 2002; Yerushalmy, 1991). Therefore, our approach is to group students using the same or similar strategies, which ensures the presence of structural similarity between their strategies, to facilitate the translation between their representations of variables and expressions. To this end, the grouping procedure includes the following phases (see also Fig. 6.2):

Phase 1. Represent all strategies stored in the Task Model as binary vectors that define similarities between them.

Phase 2. Retrieve learner strategies from the Learner Models and represent learners as vectors whose elements depict the existence of a relation between a learner' strategy and a strategy stored in the set of task strategies.

Phase 3. Define resemblance coefficients and calculate them.

Phase 4. Derive the Strategies-Learners Matrix (SLM) from the results of previous step.

Phase 5. Perform clustering on SLM.

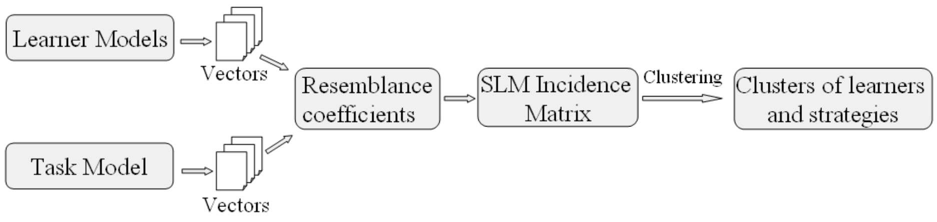


Figure 6.2: The procedure for group formation.

Definition 6.1 *Let S be the set of strategies of a task: $S = \{s_j\}$, $j = 1, 2..., n$.*

Every strategy can be represented as a n-dimensional vector of 0s and 1s: $s_j = (s_j^1, s_j^2, ..., s_j^n)$ where:

$$s_j^i = \begin{cases} 1 & \text{if } s_j \text{ is similar to strategy } s_i \\ 0 & \text{if } s_j \text{ is not similar to strategy } s_i \end{cases}$$

For example, the vectors for the five strategies of the 'pond-tiling' task illustrated in Fig. 6.3 are displayed in Table 6.1.

Each task has multiple solutions corresponding to different visual representations. Some of these solutions are similar to each other while others are different. For example, in the 'pond-tiling' task the 'H' strategy in Fig. 6.1 and the '+4' strategy in Fig. 6.3c share similar characteristics because they have the same horizontal bars, while the 'I' strategy in Fig. 6.3b and '+4' strategy in Fig. 6.3c share similar characteristics for having the same vertical bars. The constructions

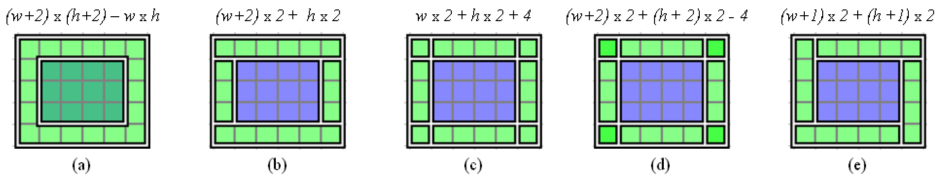


Figure 6.3: 'Pond tiling' task constructions and associated rules: (a) the 'Area' strategy; (b) the 'I' strategy; (c) the '+4' strategy; (d) the '-4' strategy; (e) the 'Spiral' strategy.

are illustrated with same pond; however, learners build constructions of various dimensions. Therefore, in this work the notion of similarity between different strategies refers to structural similarity rather than the exact dimensions of the construction.

Table 6.1: Vectors for the strategies of 'pond tiling' task

	'Area'	'I'	'+4'	'-4'	'Spiral'
'Area'	1	0	0	0	0
'I'	0	1	1	1	0
'+4'	0	1	1	0	0
'-4'	0	1	0	1	0
'Spiral'	0	0	0	0	1

In Table 6.1, the vector (0 1 1 1 0) for the 'I' strategy (displayed in Fig. 6.3b) means that this strategy is similar to itself, to the '+4' strategy (Fig. 6.3c) and to the '-4' strategy (Fig. 6.3d); the vector (0 1 1 0 0) means that the '+4' strategy is similar to itself and to the 'I' strategy (Fig. 6.3b). These similarities can be automatically deducted from the existence of structurally similar components; alternatively they can be defined by teachers.

Definition 6.2 *Let $L = \{\lambda_k\}$, $k = 1, 2, ..., m$ be the set of learners.*

A learner can be represented as a vector of 0s and 1s: $\lambda_k = (\lambda_k^1, \lambda_k^2, ..., \lambda_k^n)$, where:

$$\lambda_k^i = \begin{cases} 1 & \text{if learner } \lambda_k \text{ used } s_i \text{ strategy} \\ 0 & \text{if learner } \lambda_k \text{ did not use } s_i \text{ strategy} \end{cases}$$

For example, learner A that has used the 'I' strategy is represented as (0 1 0 0 0) and learner B that has used the 'Spiral' strategy is represented as (0 0 0 0 1). Sometimes learners use combinations of different strategies; for example, learner C who has used the 'I' and '+4' strategies would be represented as (0 1 1 0 0). This vector formulation is based on the information stored in the Learner Models; thus, learner A and B have in their Learner Models that their constructions are most similar to strategies 'I' and 'Spiral', respectively, while the Learner Model for learner C indicates that the 'I' and '+4' strategies are most similar.

Definition 6.3 *For each learner vector λ_k and each strategy vector s_j, the following are defined (see also Fig. 6.4):*

1. *a is the number of matching 1s, i.e. the number of strategies contained in both vectors;*
2. *b is the number of 1s in λ_k and 0s in s_j, i.e. the number of strategies followed by the learner which are contained in λ_k but not included in s_j;*
3. *c is the number of 0s in λ_k and 1s in s_j, i.e. the number of strategies that the learner did not follow but are included in s_j;*
4. *d is the number of matching 0s, i.e. the number of strategies that are not contained in neither of the two vectors.*

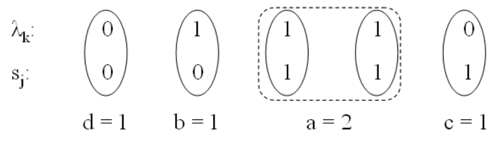


Figure 6.4: Example for Definition 6.3.

Next we use two resemblance coefficients: one for the similarity between learners and strategies, and one for the relevance of each strategy for a particular learner.

Definition 6.4 *The similarity coefficient (SC) between a learner λ_k and a strategy s_j is defined as: $SC(\lambda_k, s_j) = \frac{a}{a+b+c}$, for each learner $\lambda_k \in L$, $k = 1, 2, ..., m$ and each strategy $s_j \in S$, $j = 1, 2, ..., n$.*

This was first defined for use in Group Technology by McAuley (1972) and is in fact a Jaccard similarity coefficient - a well known measure of similarity. A study that compared 20 similarity coefficients for the cell formation problem found it to be the most stable (Yin and Yasuda, 2005), i.e. in several trials the results are within a small variation range around the average result, rather than within a wide range spanning from bad to very good results (that can still give a relatively good average).

Definition 6.5 *The Relevance Coefficient (RC) of a strategy s_j for learner λ_k is defined as: $RC(\lambda_k, s_j) = \frac{a}{a+b}$, for each learner $\lambda_k \in L$, $k = 1, 2, ..., m$ and each strategy $s_j \in S$, $j = 1, 2, ..., n$.*

The strategies-learners matrix is defined as:
$SLM = \{c_{ij}\}$, $i \in [1, n]$, $j \in [1, m]$,

$$c_{ij} = \begin{cases} 1 & \text{if } RC \geq \theta^{RC} \text{ and } SC \geq \theta^{SC} \\ 0 & \text{otherwise} \end{cases}$$

where $\theta^{RC}, \theta^{SC} \in (0, 1]$.

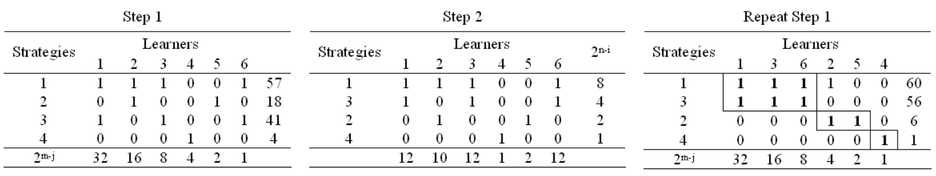
A minimum density of the matrix is necessary to obtain meaningful results. More specifically, each column should have at least a '1', i.e. each learner should follow at least one strategy. Therefore, the minimum density is the number of learners: m. Consequently, to fulfill the matrix density constraint, the values of θ^{RC} and θ^{SC} could be defined dynamically for each class. To avoid unnecessary computation, however, the following were established: (a) the value of θ^{RC} should not be lower than 0.5; this was decided because the relevance coefficient reflects the strategies followed by the learner and, consequently, should have an important role to ensure that the learner is placed in a group of learners that use at least one of the strategies followed by him/her; (b) calculate values dynamically only if the density constraint is not satisfied using the value of 0.5 for both thresholds. Therefore, the grouping starts with the value of 0.5 for both thresholds and if the matrix density constraint is not satisfied, the value of θ^{SC} is gradually decreased until the constraint is satisfied.

To illustrate the next phase of the procedure, i.e. the clustering, let us consider the matrix displayed in Step 1 of Fig. 6.5. Rank Order Clustering (ROC), one the most frequently used methods in array-based clustering (Joines et al., 1996), is applied, which involves organising columns and rows in the order of decreasing binary weights. The following procedure is applied which is illustrated in Fig. 6.5:

Step 1. Assign value 2^{m-j} to column j. Evaluate each row (using $Row_i = \sum_{j=1}^{m} c_{ij} 2^{m-j}$) and order rows in decreasing order. If there is no change compared to previous order, stop. Else, go to step 2.

Step 2. Assign value 2^{n-i} to row i. Evaluate each column ($Column_j = \sum_{i=1}^{n} c_{ij} 2^{n-i}$) and order columns in decreasing order. If there is no change compared to previous order, stop. Else, go to step 1.

In this example, the following clusters were formed: learners 1, 3 and 6 with strategies 1 and 3; learners 2 and 5 with strategy 2, and learner 6 with strategy 4. In this particular example the blocks of 1s are clear cut; however, that is rarely the case, showing that clusters are not independent. Also, one strategy may be used by many learners, forming a big cluster; this is known as the "bottleneck machine problem", when a large number of components need to



Step 1

Strategies	Learners 1	2	3	4	5	6	
1	1	1	1	0	0	1	57
2	0	1	0	0	1	0	18
3	1	0	1	0	0	1	41
4	0	0	0	1	0	0	4
2^{m-j}	32	16	8	4	2	1	

Step 2

Strategies	Learners 1	2	3	4	5	6	2^{n-i}
1	1	1	1	0	0	1	8
3	1	0	1	0	0	1	4
2	0	1	0	0	1	0	2
4	0	0	0	1	0	0	1
	12	10	12	1	2	12	

Repeat Step 1

Strategies	Learners 1	3	6	2	5	4	
1	**1**	**1**	**1**	1	0	0	60
3	**1**	**1**	**1**	0	0	0	56
2	0	0	0	**1**	**1**	0	6
4	0	0	0	0	0	**1**	1
2^{m-j}	32	16	8	4	2	1	

Figure 6.5: Steps of Rank Order Clustering example.

be processed by one machine. In the context of forming groups for collaboration using *eXpresser*, these constraints are not considered critical limitations for the formulation of strategies-learners clusters: if clusters are not independent, it means that some learners are using other strategies besides the ones of that cluster; if many learners are using the same strategy, forming a large cluster, it can be broken down in several subgroups for the purpose of the collaborative task.

6.4 Evaluation

We illustrate the approach presented in the previous sections using data from a classroom session where 18 students used *eXpresser* to solve the 'stepping stones' task. Out of the 18 learners, 6 used the 'C' strategy (C) illustrated in Fig. 6.6b, 4 used the 'HParallel' strategy (H) displayed in Fig. 6.6c, 2 used the 'VParallel' strategy (V) illustrated in Fig. 6.6d, 1 used the 'Squares' strategy (S) displayed in Fig. 6.6d, 1 used a combination of 'HParallel' and 'VParallel' strategies (H&V) and the remaining 4 students were either off-task or used non-systematic approaches such as building the construction using individual tiles - see Table 6.2.

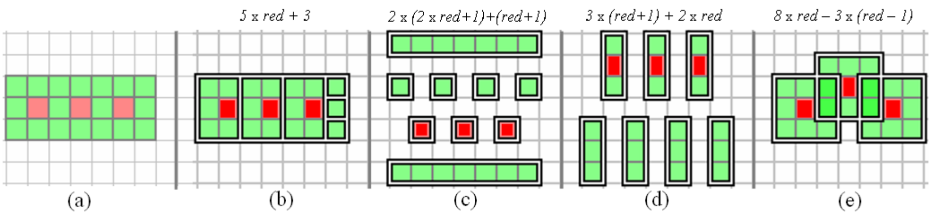


Figure 6.6: 'Stepping stones' task constructions and associated rules: (a) the task construction regardless of structure; (b) the 'C' strategy; (c) the 'HParallel' strategy; (d) the 'VParallel' strategy; (e) the 'Squares' strategy

A subset of the vectors for strategies and learners is displayed in Table 6.3. For learners that used the same strategy, only one example is provided; for example, learners λ_1 to λ_6 have the same vectors and thus only learner λ_1 is displayed. The learners that did not follow a systematic approach are excluded.

Table 6.2: Distribution of strategies used by learners.

Strategies	C	H	V	S	H&V	Other
Number of Learners	6	4	2	1	1	4

As shown in Table 6.2, four learners used non-systematic approaches to solve the task denoted by 'Other'. These could also be represented by a distinctive vector which will result in a cluster formed by these learners; however, they are already classified as a distinctive group and, therefore, including them in the grouping mechanism will only lead to unnecessary computations.

Table 6.4 displays the values of RC and SC for each strategy and learner. Using $\theta^{RC} = 0.5$ and $\theta^{SC} = 0.5$, the initial matrix in Table 6.5 is obtained; applying ROC to it leads to the final matrix in Table 6.5 and to the following groups:

(1) Group 1 includes learners $\lambda_i, i = 1, 2, ..., 6$ and λ_{13} that adopted the 'C' and 'Squares' strategies;

(2) Group 2 includes learners $\lambda_i, i = 7, 8, ..., 10$ and λ_{14} that adopted the 'HParallel' strategy;

(3) Group 3 includes learners $\lambda_i, i = 11, 12$ that adopted the 'VParallel' strategy.

The advantages of using this method, as opposed to clustering based only on the strategies used, is that the similarities between different strategies could be modified by the teacher. They could vary from being very strict (a strategy is similar only to itself) to being very relaxed (a strategy is similar to other strategies when there is at least one part that is similar). Given the way classes are formed in the UK, based on achievement levels, a relaxed definition of similarity would be more appropriate for high achieving classes that need more challenges, while a strict definition of similarity would be more appropriate for low achieving classes. Our proposed approach, thus, gives the necessary flexibility to teachers to define the similarity depending on the characteristics of

Table 6.3: Strategies and learners vectors.

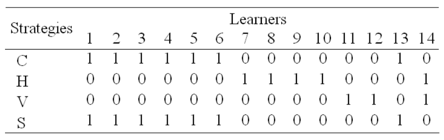
Strategies	C	H	V	S	Learners	λ_1	λ_7	λ_{11}	λ_{13}	λ_{14}
C	1	0	0	1	C	1	0	0	0	0
H	0	1	0	0	H	0	1	0	0	1
V	1	0	0	0	V	0	0	1	0	1
S	1	0	0	1	S	0	0	0	1	0

Table 6.4: Similarity between strategies and learners and relevance of strategies for each learner.

Strategies		λ_1	λ_2	λ_3	λ_4	λ_5	λ_6	λ_7	λ_8	λ_9	λ_{10}	λ_{11}	λ_{12}	λ_{13}	λ_{14}
C	RC	1	1	1	1	1	1	0	0	0	0	0	0	1	0
	SC	0.5	0.5	0.5	0.5	0.5	0.5	0	0	0	0	0	0	0.5	0
HParallel	RC	0	0	0	0	0	0	1	1	1	1	0	0	0	0.5
	SC	0	0	0	0	0	0	1	1	1	1	0	0	0	0.5
VParallel	RC	0	0	0	0	0	0	0	0	0	0	1	1	0	0.5
	SC	0	0	0	0	0	0	0	0	0	0	1	1	0	0.5
Squares	RC	1	1	1	1	1	1	0	0	0	0	0	0	1	0
	SC	0.5	0.5	0.5	0.5	0.5	0.5	0	0	0	0	0	0	0.5	0

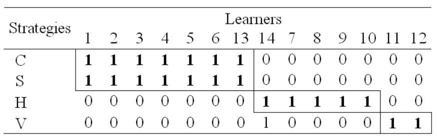
Table 6.5: Initial and final matrix.

Initial matrix



Strategies	Learners													
	1	2	3	4	5	6	7	8	9	10	11	12	13	14
C	1	1	1	1	1	1	0	0	0	0	0	0	1	0
H	0	0	0	0	0	0	1	1	1	1	0	0	0	1
V	0	0	0	0	0	0	0	0	0	0	1	1	0	1
S	1	1	1	1	1	1	0	0	0	0	0	0	1	0

Final matrix (after ROC)



Strategies	Learners													
	1	2	3	4	5	6	13	14	7	8	9	10	11	12
C	**1**	**1**	**1**	**1**	**1**	**1**	**1**	0	0	0	0	0	0	0
S	**1**	**1**	**1**	**1**	**1**	**1**	**1**	0	0	0	0	0	0	0
H	0	0	0	0	0	0	0	**1**	**1**	**1**	**1**	**1**	0	0
V	0	0	0	0	0	0	0	1	0	0	0	0	**1**	**1**

the class. More specifically, this can be done by considering this aspect when defining the vectors for strategies using Definition 6.1 given in Section 6.3.

For example, a teacher may consider the 'Squares' strategy to be similar to the 'VParallel' rather than the 'C' strategy. Consequently, the strategies vectors would be: (a) 'C' strategy (1 0 0 0); (b) 'HParallel' strategy (0 1 0 0); (c) 'VParallel' strategy (0 0 1 1); (d) 'Squares' strategy (0 0 1 1). Using these vectors a new SLM matrix is obtained and the clustering procedure outputs the following groups:

(1) Group 1 includes learners $\lambda_i, i = 1, 2, ..., 6$ that used the 'C' strategy;

(2) Group 2 includes learners $\lambda_i, i = 7, 8, ..., 10$ and λ_{14} that used the 'HParallel' strategy;

(3) Group 3 includes learners $\lambda_i, i = 11, 12$ and λ_{14} that used the 'VParallel' and 'Squares' strategies.

The mechanism we developed provides the teachers with groups based on the strategies followed by learners, i.e. the clusters formed as explained above. Using this information, teachers decide the size of groups and how the learners are distributed. Currently, our approach does not include social, cultural or personality factors, which are handled by the teacher. Future work, however, will look at integrating these factors and at automating the group formation in a flexible manner that, for example, will allow teachers to enter constrains such as ‘learner X should never be grouped with learner Y’.

6.5 Discussion

The research presented in this chapter had to address the challenging task of developing a grouping mechanism that takes into account not only the information about the strategy or strategies followed by each learner, but also the similarity between the different strategies of a task. To this end, we were inspired by Group Technology approaches as they offer a way of formulating the problem according to the desired criteria and of performing the grouping using simple clustering methods.

While the Group Technology inspired mechanism performs well, it has the limitation of using a ‘black and white’ approach rather than a continuous measurement scale. Thus, a strategy is either similar to another one or it is not similar at all. A grading scale, however, could be defined to reflect different degrees of similarity. In the same way, the similarity of a learner’s strategy to all stored strategies is defined as either similar or dissimilar. It would be useful in the future to extend this mechanism to exploit information stored in the Learner Model about the most similar strategies, including similarity values for each one. For example, if a learner’s strategy is similar to ‘HParallel’ and ‘VParallel’ strategies, with the values of 2.37 and 3.14, respectively, this information could be used instead of the ‘black and white’ approach.

Currently the approach does not distinguish between the types of strategies described in Section 4.3.3: complete, mixed, non-systematic and partial, as we consider that mixing these types would be beneficial for learning. Consequently, the groups include learners using these different types. Often, the learners who follow complete strategies have used the other types before, and in practice teachers usually invite these learners to act as tutors for their peers. Research shows that peer tutors usually benefit by taking up this role because it helps them to reflect on their own knowledge and use it as a basis for constructing new knowledge - a process referred to as knowledge-building (Roscoe and Chi, 2007). Three properties of peer tutoring have been related to tutor learning: structuring, taking responsibility and reflecting (Biswas et al., 2005). Giving explanations, asking and answering questions helps peer tutors in structuring their own knowledge; taking responsibility for their tutee’s learning motivates peer tutors to gain a better understanding of the material; peer tutors’ reflection on how their explanations were understood and used helps them in evaluating their own understanding of the domain.

Research also shows that tutee learning is maximised when the tutee reaches an impasse and is prompted to find the right way to continue and explain it, and is given an explanation only if they failed to do so (Vanlehn et al., 2003). Therefore, learners with partial constructions that do not know how to continue would benefit from the explanations of a peer that has completed the same strategy; the ones with mixed strategies would benefit from discussing the similarities and differences between their approaches; the ones with non-systematic strategies would learn about the benefit of working with systematic approaches.

6.6 Summary and Contribution of the Chapter

In this chapter we presented an approach for group formation that exploits ideas from Group Technology and takes into consideration aspects that are relevant for exploratory learning environments in general, and *eXpresser* in particular. The criteria used in the grouping process are the strategies followed by learners in solving a task and the similarities between different strategies of a particular task.

Resemblance coefficients were used to define: (a) the similarity between learners and strategies and (b) the relevance of each strategy for a particular learner. The approach outputs homogeneous groups; however, heterogeneous groups can be formed by choosing one or more learners from each or some of the homogeneous groups.

The mechanism we propose has an important advantage compared with simple clustering, which is the flexibility given to teachers in defining similarities between strategies. When similarity is defined in a strict way, i.e. a strategy is similar only to itself, our mechanism gives the same output as a simple clustering method. The latter, however, does not allow a more relaxed definition of similarity, i.e. a strategy is similar to other strategies when there is at least one part that is similar or when there is some conceptual similarity, while our approach supports such definitions.

Similarly to the learner modelling and the feedback prioritisation mechanisms, the proposed approach for grouping can be applied for other learning environments where it is important to consider the above mentioned criteria, i.e. ill-defined domains where a problem has multiple equally valid solutions.

In summary, the contribution of this chapter is twofold: (a) identifying relevant criteria for exploratory learning in general and *eXpresser* in particular and (b) defining a mechanism that outputs groups of learners based on the above mentioned criteria.

The evaluation showed that our proposed approach works in the context of exploratory learning and that it provides meaningful grouping, i.e. learners with similar strategies are placed in the same cluster. This evaluation is based on judging the numerical outputs of the mechanism against the requirement to cluster learners based on the similarity between their strategies. A more powerful evaluation, however, would be to present teachers with the outcomes of the mechanism and ask for them to evaluate them.

Chapter 7

Task Knowledge Base Adaptation

In Chapter 4 we proposed a learner modelling mechanism for monitoring learners' actions when constructing/exploring models by modelling sequences of actions reflecting different strategies in solving a task. An important problem, however, remains: only a limited number of strategies are known in advance and can be introduced by the designer/teacher. In addition, even if all strategies were known, introducing them in the knowledge base would take considerable time and effort. Moreover, the knowledge about a task evolves over time - students may discover different ways of approaching the task, rendering the knowledge base suboptimal for generating proper feedback, despite the initially good coverage. To address this, we employ a mechanism for adapting the knowledge base in the context of the second version of the ELE for mathematical generalisation, i.e. *eXpresser*.

The task knowledge base adaptation involves a mechanism for acquiring inefficient cases, i.e. cases which include actions that make it difficult for students to create a generalisable model, and a mechanism for acquiring new strategies. The former could be potentially useful to enable targeted feedback about the inefficiency of certain parts of a construction, or certain actions of the student; this approach could also lead gradually to creating a library of inefficient constructions produced by students that could be analysed further by a researcher/teacher. Without the later a new valid strategy will not be recognised as such, and, consequently, the learner modelling module will diagnose the learner to be still far from a valid solution and any potential feedback will be confusing as it will guide the learner towards the most similar strategy stored in the knowledge base.

The next section gives a brief overview of adaptive modelling and positions our approach within the literature. Section 7.2 presents the proposed approach for knowledge base adaptation in the context of *eXpresser*, Section 7.3 describes the evaluation studies, and Section 7.4 discusses the results and presents directions for future work. Finally, Section 7.5 summarises the chapter and gives an overview of the contribution.

7.1 Adaptive Modelling

Adaptive systems refer to systems that change over time to respond to new situations. There are three levels of adaptation depending on the complexity and difficulty of the adaptation process, with the first level being the least difficult and the third being the most complex and difficult (Anguita, 2001):

1. Adaptation to a changing environment;
2. Adaptation to a similar setting without explicitly being ported to it;
3. Adaptation to a new/unknown application.

As our adaptive modelling mechanism involves adaptivity at the first level, we will review the literature on this aspect in more detail. At this level, the system adapts itself to a drift in the environment by recognising the changes and reacting accordingly (Anguita, 2001).

The most frequently used tools for adaptation are: artificial neural networks (ANN), fuzzy systems, evolutionary computation and machine leaning.

In the filed of artificial neural networks the term *catastrophic interference* (Sharkey and Sharkey, 1995) and *Stability-Plasticity Dilemma* (Grossberg, 1987) have been often used for the problem of adapting to a changing environment. The former refers to the problem of learning and retaining information that arrives in sequential episodes over time, as training on a new set of items may drastically disrupt performance on previously learned items (McCloskey and Cohen, 1989). The latter refers to the dilemma of a system having to be adaptive enough to allow learning new things while not diluting or forgetting previously learned patterns (Grossberg, 1987). Therefore, essentially, both terms refer to the same problem.

Research in this area spans from theoretical works (Lecerf, 1999; Hamker, 2001; Robins, 2004; Abraham and Robins, 2005; Fernando, 2010) to applied research in areas such as identifying unique orthographic word forms (Glotin et al., 2010), colour image segmentation (Yeo et al., 2005) and fault diagnosis (Xu et al., 2009).

Fuzzy systems have been used in dynamic contexts and substantial work has been done in the area of adaptive or evolving rule-based models (Angelov and Buswell, 2002; Angelov, 2003; Xydeas et al., 2006; de Barros and Dexter, 2007; Angelov et al., 2008). Evolutionary computation uses adaptive (e.g. Igel and Kreutz (2003); Mallipeddi et al. (2010)) and self-adaptive techniques (e.g. Fister et al. (2010); Montalvo et al. (2010)). Machine learning is found at the core of adaptation problems, as it addresses the problem of adaptivity through learning (Anguita, 2001). The research includes symbolic methods such as decision trees (Tong et al., 2010) and rules (Yang and Shao, 2007), sub-symbolic methods such as neural networks (Gadkar et al., 2005) and Bayesian learning (Hirayama et al., 2006), and statistical methods such as regression analysis (Rodríguez-Serrano et al., 2010), cluster analysis (Zhang et al., 2009) and discriminant analysis (Dogantekin et al., 2010).

Case-based Reasoning is associated with the second level of adaptivity (Anguita, 2001), i.e. adapting to a similar setting, due to the principle of *reasoning by analogy* that is used, which involves reusing and adapting solutions of known problems to solve similar problems. In our research, however, the knowledge base adaptation is not done through the reuse and adapt processes in CBR, but using an algorithmic approach that exploits knowledge of the domain, and more specifically, of the tasks involved. Therefore, our approach involves adaptivity only at the first level, and is presented in detail in the following section.

7.2 Adaptive Modelling for *eXpresser*

Our approach for adapting the task knowledge base of *eXpresser* includes acquiring inefficient simple cases and acquiring new strategies. Some examples from the 'stepping stones' task are displayed in Fig. 7.1 for the two situations; the constructions are broken down into individual components used by the students for ease of visualisation. These examples, with the adaptation rationale and mechanism are discussed below.

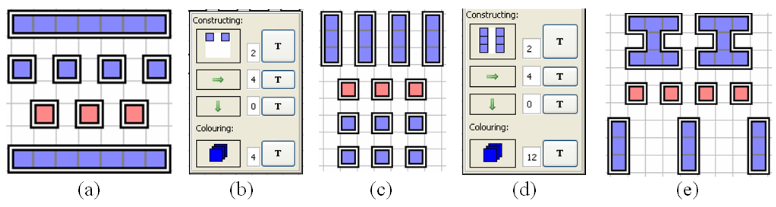


Figure 7.1: (a)-(b) 'HParallel' strategy with an inefficient component (blue middle row) and its property list; (c)-(d) 'VParallel' strategy with an inefficient component (blue vertical bars) and its property list; (e) a new strategy

Acquiring inefficient simple cases. The goal of this mechanism is to identify parts of strategies constructed in inefficient ways and store them in a set of 'inefficient constructions', i.e. constructions that pose difficulties for learners in their process of generalisation. The set could be used for automatic generation of feedback or analysed by a researcher/teacher. The results could inform the design of better interventions, could be presented as a lesson learned to the scientific community of mathematics teachers/researchers, or even discussed in class (e.g. if an inefficient construction is often chosen by pupils in that class).

The construction in Fig. 7.1a illustrates an inefficient case within the 'HParallel' strategy (see Figure 4.15c in Section 4.4.2) of the 'stepping stones' task: the middle bar of blue tiles is constructed as a group of two tiles repeated twice - see its property list in Fig. 7.1b. The efficient way to construct this component is one tile repeated 4 times or, to make it general, one tile repeated the number of red tiles plus 1. Both approaches, i.e. the efficient and the inefficient, lead to the same visual output, i.e. there is no difference in the way the construction looks like, making the situation even more confusing. The difficulty lies in relating the values of the blue middle row (C_i) to the ones of the red middle row (C_j). If the learner relates value 2 of iterations of C_i to value 3 of iterations of C_j, i.e. value 2 is obtained by using the iterations of red tiles (3) minus 1, this would work only for a 'stepping stones' task defined for 3 red tiles. In other words, this will not lead to a general model. Another example of an inefficient case is given in Fig. 7.1c, with its property list presented in Fig. 7.1d. These inefficient cases are not related to mathematical misconceptions, but seem to be a consequence of the system's affordances combined with little experience of generalisation tasks, and come from learners' wish to make there patterns bigger without considering the generality of their approach. However, they are pedagogically important, offering learners the possibility to reflect on their actions in relation to generalisation.

Algorithms 1, 2 and 3 illustrate how inefficient simple cases are identified and stored. First, the most similar strategy is found. If there is no exact match, but the similarity is above a certain threshold θ_1, the process continues with the identification of the inefficient cases; for each of these cases, several checks are performed (Alg. 2). Upon satisfactory results and if the cases are not already in the set of inefficient cases, they are then stored (Alg. 3). What is stored is actually a modification of the most similar (efficient) case, in which only the numerical values of *iterations*, *move-right* and/or *move-down* are updated together with the value and dependency relations. These are the only modifications because, on one hand, they inform the way in which the pattern was built, including its non-generalisable relations, and, on the other hand, it is important to preserve the values of $PartOfS$ attributes, so the researcher/teacher knows in which strategies these can occur. The colouring attributes and the relation between cases are not important for this purpose and, therefore, they are not modified. This has also the advantage of being computationally cheaper.

Algorithm 1 Verification($StrategiesCaseBase$, $InputStrategy$)

```
Find most similar strategy to $InputStrategy$ from $StrategiesCaseBase$
$StoredStrategy \leftarrow$ most similar strategy;
if $similarity > \theta_1$ then
    Find cases of $InputStrategy$ that are not an exact match to any case of $StoredStrategy$
    for each case that is not an exact match do
        $InputCase \leftarrow$ the case that is not an exact match
        Compare $InputCase$ to all cases of the set of inefficient cases;
        if no exact match then
            Find the most similar case to $InputCase$ from the cases of $StoredStrategy$
            $StoredCase \leftarrow$ the most similar case
            if Conditions($StoredCase$, $InputCase$) returns true then {see Alg. 2}
                InefficientCaseAcquisition($StoredCase$, $InputCase$) {see Alg. 3}
            end if
        end if
    end for
end if
```

Algorithm 2 Conditions($C1$, $C2$)

```
if ($MoveRight[C1] \neq 0$ and $Iterations[C1]*MoveRight[C1] = Iterations[C2]*MoveRight[C2]$)
or
($MoveDown[C1] \neq 0$ and $Iterations[C1]*MoveDown[C1] = Iterations[C2]*MoveDown[C2]$)
then
    return true
else
    return false
end if
```

Algorithm 3 InefficientCaseAcquisition($StoredCase$, $InputCase$)

```
$NewCase \leftarrow StoredCase$
for $i = 4$ to $v - 1$ do {attributes from iterations to move-down}
    if value of attribute $i$ of $NewCase$ is different from that of $InputCase$ then
        replace value of attribute $i$ of $NewCase$ with the one of $InputCase$
    end if
end for
for all relations between attributes do {value an dependency relations}
    replace relations of $NewCase$ with the ones of $InputCase$
end for
add $NewCase$ to the set of inefficient cases
```

New strategy acquisition. The goal is to identify new strategies and store them for future use. New strategies could be added by the teacher or could be recognised from the learners' constructions. In the later case, after some verification checks (see Algorithm 5), the decision of storing a new strategy is left with the teacher. This serves as another validation step for the detected new strategy.

Fig. 7.1e illustrates the so-called 'I' strategy, due to its resemblance to letter I. When compared to all strategies, it is rightly most similar to the 'VParallel' one (see Figure 4.15d in Section 4.4.2), as some parts correspond to it. However, the similarity is low, suggesting it may be a new strategy. Without the adaptation mechanism, the learner modelling module will infer the learner is using the 'VParallel' strategy, but is still far from completion. This imprecise information is potentially damaging as it could, for example, lead to inappropriate system actions, e.g. providing confusing feedback by guiding the learner towards the 'VParallel' strategy. Conversely, the adaptation mechanism will prevent generating such confusing feedback, and storing the new strategy will enable appropriate feedback in the future - automatically or with input from the teacher/researcher.

Algorithms 4, 5 and 6 illustrate how an input strategy is identified and stored as a new strategy (composite case). If the similarity between the input strategy and the most similar strategy from the case-base is below a certain threshold θ_2 (Alg. 4), some validation checks are performed (Alg. 5) and upon satisfaction, the new strategy is stored in the case-base (Alg. 6). If the input strategy has been introduced by a teacher and the similarity is below θ_2, the teacher can still store the new strategy, even if very similar to an existing one.

Algorithm 4 NewStrategyVerification($StrategiesCaseBase$, $InputStrategy$)

```
Find most similar strategy to $InputStrategy$ from the $StrategiesCaseBase$
if $similarity < \theta_2$ then
  if ValidSolution($InputStrategy$) returns true then {see Alg. 5}
    NewStrategyAcquisition($InputStrategy$) {see Alg. 6}
  end if
end if
```

Algorithm 5 ValidSolution($InputStrategy$)

```
if SolutionCheck($InputStrategy$) returns true then {checks if $InputStrategy$ 'looks like' a solution}
  if the number of cases of $InputStrategy < \theta_3$ then
    if $InputStrategy$ has relations between attributes then
      RelationVerification($InputStrategy$) {verifies that the numeric relation corresponds to the task rule solution}
      if successful verification then
        return true
      end if
    end if
  end if
end if
```

Algorithm 6 NewStrategyAcquisition($NewStrategy$)

```
add $NewStrategy$ to the strategies case-base
adjust values of $PartOfS$
```

In Algorithm 5 the SolutionCheck($InputStrategy$) function verifies whether $InputStrategy$ 'looks like' a solution by examining if the mask of $InputStrategy$ corresponds to the mask of the task. The following check takes into consideration the number of simple cases in the $InputStrategy$. Good solutions are characterised by a relatively small number of simple cases; therefore, we propose for the value of θ_3 the maximum number of cases among all stored strategies for the corresponding task, plus a margin error (such as 3). If this check is satisfied, the RelationVerification($InputStrategy$) function derives a rule from the value relations of the cases and checks its correspondence to the rule solution of the task. For example, in the construction of Fig. 7.1c, the rule derived is $3*(\frac{red}{2}+1)+7*\frac{red}{2}$ which corresponds to the solution $5*red+3$. If all checks are satisfied, the new strategy is stored in the case-base and the $PartOfS$ values are adjusted.

7.3 Evaluation

The validation of our proposed mechanisms includes:

(a) identifying the boundaries of how far a pattern can be modified and still be recognised as similar to its original;

(b) correct identification of inefficient cases within these boundaries and

(c) correct identification of new strategies.

This low-level testing of the system shows how the adaptation of the knowledge-base and the learner modelling module function together to improve the performance of the system.

To this end, experiments have been conducted using real as well as artificial data. The real data comes from small trials (1 to 4 pupils) as well as classroom use of *eXpresser*; the classroom data comes from two sessions that took place in July 2008 in a secondary school in London (the same as the ones mentioned in Section 4.4.3). The artificial data was obtained by varying parameters of the constructions produced by learners in the real settings mentioned above. More specifically, to obtain artificial data based on a learner's construction, the values of the attributes of individual cases were changed, but the overall structure of the construction is maintained; in other words, the artificial data will reflect a different instance of the task than the one used by the learner, but the approach used by the learner is preserved. For example, artificial data based on a learner's construction using the 'H strategy' for the pond-tiling task and working with a pond of width 5 and height 7, would be a construction using a pond of width 6 and height 4 following the same strategy (i.e. 'H strategy'). For partial and mixed constructions, the same principles apply, i.e. the constructions are proportionally adjusted to reflect the same structure.

The use of artificial data was necessary to provide a more realistic evaluation of the algorithms, as the amount of real data available was small. Overall, 55% is real data and 45% is artificial data. More details about the distribution is given below, for each experiment.

First, a preliminary experiment using classroom data was conducted to identify possible values for the threshold θ_1 in Algorithm 1 and threshold θ_2 in Algorithm 4. Since our main aim was to test the adaptive modelling mechanism we decided not to seek optimal values for these thresholds, but only to find a good enough value for each one. Two possibilities were quickly identified - for θ_1: the minimum *overall similarity* (4.50) minus an error margin (0.50) or value 1.00 for the *numerical similarity*; for θ_2: the maximum *overall similarity* (3.20) plus an error margin (0.30) or value 1 for the *numeric similarity*.

Experiment 1: identifying the boundaries of how far a pattern can be inefficiently modified and still be recognised as similar to its original efficient form. As mentioned previously, we consider changes in a pattern that can lead to the same visual output as the original one but use different building-blocks. More specifically, these building-blocks are groups of two or more of the original efficient building-block. This experiment looks for the limits of changes that a pattern can undergo without losing its structure. We used 34 artificial inefficient cases from the 'pond tiling' and 'stepping stones' tasks.

From the 34 cases, 47% were from the 'stepping stones' task and 53% were from the 'pond tiling' task. Using these cases, the following boundaries were identified:

(i) groups of less than 4 building-blocks;

(ii) groups of 2 building-blocks repeated less than 6 times and

(iii) groups of 3 building-blocks iterated less than 4 times.

To illustrate these bounds the 'C' strategy of the 'stepping stones' task is used in Fig. 7.2. The building-block of this strategy is a 'C shape' pattern. The 1st column illustrates an inefficient pattern that is still recognisable as similar to the efficient form displayed in the 3rd column and the 2nd column displays an inefficient pattern that is not recognisable anymore as similar to the efficient form. The three rows correspond to the three bounds mentioned above. The iterations of the efficient pattern from the 3rd column refer to the number of iteration needed to obtain the same shapes as the ones from the 1st and the 2nd column. For example, in the 1st row, the '3/4 iterations' denotes that 3 iterations are needed to obtain the same shape as the one in the 1st column and that 4 iterations are needed to obtain the same shape as the one in the 2nd column; the property list displays the first value.

Due to the awareness of the sub-patterns in the examples illustrated above, to the human eye the differences between the recognisable and not recognisable inefficient patterns seem very small. The metrics, however, do not incorporate this knowledge and, therefore, are limited to the bounds illustrated above. An interesting direction for future work would be to investigate the effects of incorporating such knowledge in the metrics.

Experiment 2: correct identification of inefficient cases within the previously identified boundaries. From the 34 inefficient cases of *Experiment 1*, 13 were outside the identified boundaries and 21 were within. From the 21 cases within the boundaries, 62% were from the 'stepping stones' task and 38% were from the 'pond tiling' task. Using the previously identified values for θ_1, we obtained the following results: out of these 21 cases, 52.48% had the overall similarity

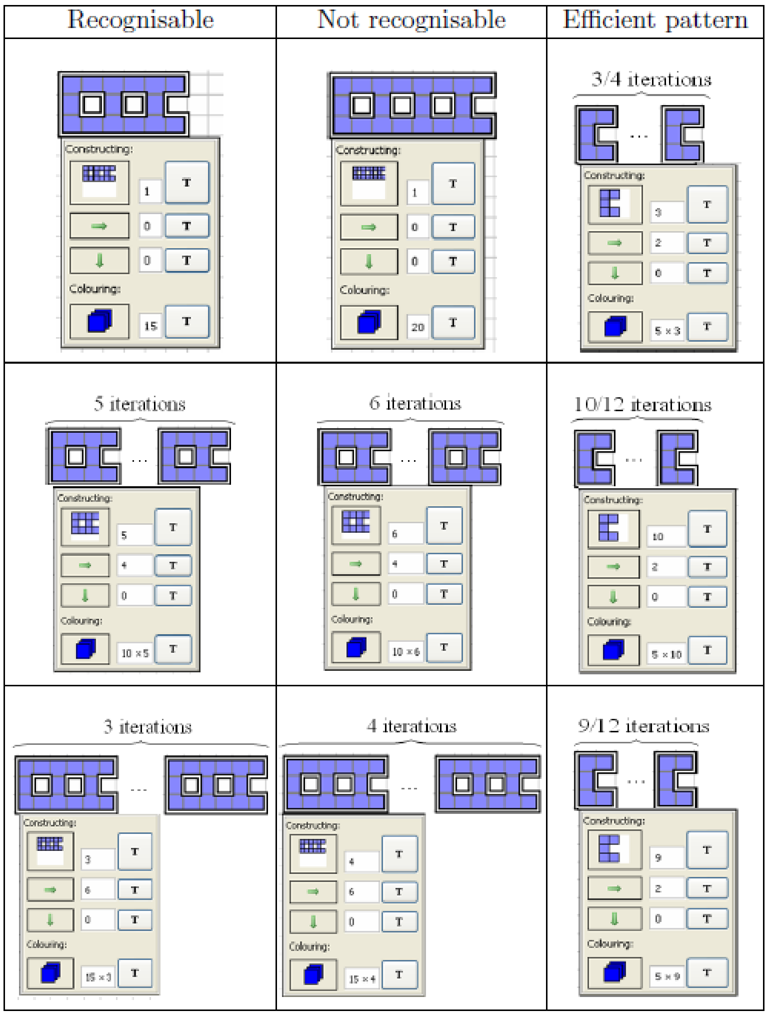


Figure 7.2: Bounds for recognition of inefficient patterns

greater than 4.00 and 100% had the numeric similarity above 1.00. These results indicate that a small modification of a pattern can drastically affect the identification of the strategy the learner is following; hence almost half the cases have an overall similarity less than 4.00. The results obtained using the numeric similarity are much better, which was expected given that the modifications are just numerical.

Experiment 3: correct identification of strategies. The data for this experiment included 10 new strategies: 7 observed in trials with pupils and 3 artificial. Out of the 10 new strategies, 4 were from the 'pond tiling' task; all of them were observed in trials with pupils. The remaining 6 new strategies were from the 'stepping stones' task, with 3 of them observed and 3 artificial. The knowledge base for the two tasks included originally 4 strategies for the 'stepping stones' task and 2 strategies for the 'pond tiling' task. Using the previously identified values for θ_2, we obtained the following results: out of the 10 new strategies, 100% had the overall similarity below 3.50 and 70% had the numeric similarity below 1.00. As opposed to Experiment 2, the overall similarity performs better, being consistent with the fact that the overall similarity reflects better the resemblance with the stored strategies than the numeric similarity alone. Given the range that the overall similarity has, i.e. from 0 to 10, values below 3.50 indicate a very low similarity and therefore, rightly suggest that the learner's construction is considerably different from the ones in the knowledge base.

7.4 Discussion

A summary of results is displayed in Table 7.1. They show that the algorithms are capable of recognising new relevant data with an accuracy of 100% using two of the previously identified values for the thresholds.

Table 7.1: Summary of results

Experiment	Threshold		Accuracy
Identification of inefficient patterns	Overall similarity	4	52.48%
	Numeric similarity	1	100%
Identification of new strategies	Overall similarity	3.50	100%
	Numeric similarity	1	70%

As already mentioned in Section 4.4.2, the numeric similarity ($\overline{F_1}$) reflects the structure of a construction. Therefore, using the numeric similarity for the thresholds θ_1 and θ_2 structural modification is identified - performing very well in *Experiment 2*, where all changes are numerical, but not as well in *Experiment 3*, where the changes are not exclusively numerical. As a strategy is not defined only by patterns and their structure, but by relations as well, the overall similarity performs better in *Experiment 3*.

Automatic detection of inefficient cases and new strategies is potentially useful for improving the feedback given to learners by automatically incorporating related information or by informing the teacher and allowing them to author feedback they consider appropriate.

The algorithmic approach presented here has the advantage of being closely tied up with the strategy identification mechanism, thus, using the same similarity metrics to identify inefficient patterns and new strategies. This advantage for our particular environment, i.e. *eXpresser*, however, could be seen as a disadvantage in terms of the generality of the approach for other learning environments.

There are two aspects of the algorithm approach, each with different implications on the applicability of the proposed mechanism to other environments. One aspect is the user modelling mechanism and the approach taken to diagnose the learner and the other is the identification of the new relevant data. As the algorithmic approach developed here is tailored for *eXpresser* and is intertwined with the user modelling mechanism, the approach is not applicable to other learning environments as such. The high-level approach, however, could be applied to other systems provided that task or domain-related information is available about how to test the validity of a new approach.

7.5 Summary and Contribution of the Chapter

In this chapter we presented an approach for adaptive modelling of the tasks knowledge base. This work was motivated by the fact that due to the open nature of the environment not all strategies are known in advance and that learners use the system in inefficient ways that lead to difficulties in solving the given tasks. To overcome these problems, we developed an adaptive modelling mechanism to expand an initially small knowledge base, by identifying inefficient cases (i.e. cases that pose additional difficulty to the user's learning process) and new strategies.

The proposed mechanism builds on the learner modelling mechanism presented in Chapter 4 in general, and the similarity metrics in particular (presented in Section 4.4.2). For both inefficient patterns and new strategies, the principle is the same: they are compared with data from the knowledge base and if they are not already stored, some checks are performed and upon successful verification, they are added to the knowledge base. The checks are related to specific information about a particular task which are available from the Task

Model. With this mechanism, new data can be added to the knowledge base without affecting the recognition of existing data.

Our adaptive modelling mechanism ensures that the learner diagnosis will be accurate even when the researcher or teacher authors only one or two strategies for a new task. Also, it ensures that the learner diagnosis will be accurate when learners' behaviour changes over time even if initially there is a large knowledge base.

The evaluation of the proposed approach showed that it is capable of recognising new inefficient patters within certain boundaries and of recognising new strategies. The boundaries for recognising inefficient patterns are related to the similarity metrics' ability to identify how much they have been modified from their original initial form. When looking at the modifications that learners tend to make, we notice that they take the form of using repetitions of the basic building-block, which modify the structure of the pattern. The similarity metrics, however, were defined to recognise *structural* similarity. Therefore, to improve the metrics' ability to recognise modifications of efficient patterns, they should be enhanced with the capacity to recognise sub-patterns.

The adaptive mechanism that we developed was tailored for *eXpresser* and the domain of mathematical generalisation. We believe, however, that the high level idea can be used in other exploratory learning environments and for domains where several approaches are possible for the same problem. Therefore, the main contribution of this chapter is addressing one of the main issues of exploratory learning environments, i.e. an optimal coverage of the knowledge base when not everything can or is efficient to be modelled and stored.

Chapter 8

Conclusions

The research presented in this thesis belongs to the broad area of modelling human behaviour with the purpose to give computers the means to respond intelligently. In particular, it is relevant to educational systems that want to adapt to their users and their specific needs and, more specifically, to a particular type of educational systems called Exploratory Learning Environments (ELEs). These environments are different from tutoring systems in relation to two main features: (a) the freedom given to learners and (b) the characteristics of the domain, e.g. less structure, several valid approaches for the same problem, understanding of the domain involves being aware of several perspectives on the same issue.

We focused on a learner modelling mechanism in an ELE for the domain of mathematical generalisation that infers a learner's approach to a task, where each task could be approached in several ways. We have developed a learner modelling mechanism based on a modified version of Case-based Reasoning that can deal with problems that have multiple solutions. Our aim was to be able to diagnose the learner *during* as well as at the end of a task, and to use the inferred information for several educational purposes. The main contribution of this thesis is the development of the learner modelling mechanism, while several secondary contributions were made in relation to the following issues: feedback prioritisation, grouping for collaboration and maintaining an optimal coverage of the knowledge base.

This chapter presents a summary of the research and findings of the thesis. It also outlines the contribution of this work, its limitations and possible extensions.

8.1 Summary of Research and Findings

Personalised support in exploratory learning is acknowledged as an essential issue to enable learners to benefit from the advantages of learning by exploration. Achieving personalised support is, however, a very challenging task, as pointed out in Chapter 1. Several attempts have been presented in Chapter 2 with most approaches focusing on modelling the learners' knowledge of the domain (Veermans, 2003; Stathacopoulou et al., 2005; Andaloro and Bellomonte, 1998), one focusing on modelling skills related to exploratory learning (Ting and Phon-Amnuaisuk, 2009), one focusing on modelling meta-skills related to exploratory learning (Conati and Merten, 2007) and, finally, one focusing on modelling knowledge and effective exploration at the same time (Bunt, 2001; Bunt and Conati, 2003).

Unlike previous research, the work presented in this thesis focuses on problems with multiple solutions in an environment where learners build their own models. The aim is to enable learner diagnosis during as well as at the end of an exploratory learning task. From the reviewed literature in only two cases (Stathacopoulou et al., 2005; Andaloro and Bellomonte, 1998) learners construct as well as explore models, while the rest allow only exploration of models. Unlike the approach in Stathacopoulou et al. (2005), our approach is based on strategies rather than concepts of the domain. Although conceptually our research is close to the approach in Andaloro and Bellomonte (1998) in that the information stored in the knowledge base is founded on the models that are common to different students, in addition our approach has a mechanism that enriches the knowledge base with new data when new relevant information is encountered.

The research presented in this thesis could be split into three categories:

1. Modelling exploratory user behaviour, which was presented in Chapter 4;
2. Exploiting the information in the learner models for pedagogical purposes, i.e. feedback prioritisation and grouping for collaboration, which were presented in Chapters 5 and 6, respectively;
3. Maintaining an optimal knowledge base through adaptive modelling, which was presented in Chapter 7.

To model users' exploratory behaviour, a Case-based Reasoning approach was used, where each task has a knowledge base of strategies corresponding to possible solutions for the task. The strategies or composite cases are composed of simple cases which correspond to shapes in *ShapeBuilder* and patterns in *eXpresser*. To identify the strategies followed by learners, their constructions are compared to the strategies in the knowledge base of the corresponding task using an aggregated measure of similarity, and the strategies identified as most similar are stored in individual learner models. This high-level idea was developed using the initial version of the learning environment, i.e. *ShapeBuilder*, and adjusted for the next version, i.e. *eXpresser*. Both versions successfully identified situations of pedagogical importance: complete specific and general strategies, mixed strategies, non-systematic strategies and partial strategies. In addition, for the second version a mechanism for identifying off-task behaviour was also developed. These situations or scenarios were identified using two sources of information: behaviours observed in trials with pupils and pedagogical information about the importance of detecting certain behaviours.

The modelling mechanism has been developed in an iterative and incremental manner, in parallel to other components of the exploratory environment. This has influenced our approach, directing our efforts towards ensuring the flexibility and extendibility of the modelling mechanism, as "user models cannot and should not be separated from the software systems that use them" (Chin, 2001, p. 183). The case-based representation that we used allowed us to adjust the representation to the requirements of the subsequent version with minimal modifications, even if the interface of the system had considerably changed.

The information in the learner models was subsequently used for feedback prioritisation and grouping for collaboration. An approach based on the Analytic Hierarchy Model, a method from Multi-criteria Decision Making, was used for feedback prioritisation, while the mechanism for grouping for collaboration was based on Grouping Technology, a method for group formation of machines and parts in cellular manufacturing systems. The feedback prioritisation mechanism was tested using two experts who agreed with the outcome of the proposed mechanism, with one small exception. The grouping mechanism was tested using classroom data and showed that meaningful groups are created while also allowing flexibility in establishing the degree of similarity between strategies considered as a criterion for grouping.

To ensure optimal coverage of the knowledge base, an adaptive modelling mechanism was developed that recognised new relevant data and stored it. These data refer to inefficient cases, i.e. cases that make the generalisation task more difficult for the learner, and to new strategies. Three experiments were conducted and showed that both inefficient cases and new strategies were correctly identified by our adaptive mechanism.

8.2 Thesis Contributions

This section presents an overview of the thesis contributions. For each chapter describing research work, the contribution is outlined together with the research areas to which it has contributed.

The main contribution of this thesis is the development of a learner modelling mechanism in the context of an exploratory learning environment for mathematical generalisation, which was described in Chapter 4. Our proposed approach aimed to provide diagnosis of what a learner is doing during as well as at the end of a task, and thus, enable the system to adapt its response to each learner. Thus, this work contributes to the fields of *User Modelling and User-Adaptive Systems*, *Artificial Intelligence in Education* and *Intelligent Systems.*

Particularly, the proposed approach brings a new way of modelling learner behaviour in exploratory learning environments. Thus, we focused on learners' approaches to a particular task, that we call strategies and on identifying what strategy a learner is using. In this way, we addressed on of the key questions in the area of modelling exploratory behaviour, i.e. what to model? Modelling learner's strategies, rather than concepts for example, gives the advantage of having a more holistic view of the learner's perspective of a particular task. In other words, a strategy contains more information than a probability attached to a concept. Also, this is more appropriate for ill-defined domains (Lynch et al., 2006) for which exploratory learning is more suitable than tutoring, as they are often characterised by complex problems, in which a concept cannot be explored in separation from other ones because the essence lies in the relation between concepts.

Although we developed the learner modelling mechanism for our particular exploratory learning environment (ELE) and a specific domain, i.e. mathematical generalisation, we believe that the high level idea can be applied to other ELEs and domains where a problem can have several equally valid solutions. Moreover, the knowledge representation could potentially extend beyond educational settings - for example, to manufacturing problems such as block assembly or to design problems with reusable parts.

Using the modelling mechanism, we addressed two pedagogical issues: personalised feedback prioritisation and grouping for collaboration. The issue of feedback prioritisation emerged from our choice to model strategies, as these included information on several aspects relevant to the tasks. We did not encounter any previous research on this issue in computer-based learning environments of any type. As our approach aims to deliver personalised feedback priorities, it contributes to the field of *Artificial Intelligence in Education.*

The approach we developed for grouping learners for the purpose of collaborative activities also contributes to the field of *Artificial Intelligence in Education* because it aims to deliver groups in such a way as to ensure that each learner will be in a group that discusses relevant aspects with respect to their individual approaches.

Our approach to grouping for collaboration is different from previous research in that the criteria used is driven by the aims of the collaborative activities in the context of exploratory learning. These aims are to discuss similarities and differences between approaches that learners developed on their own prior to the collaborative activities. Therefore, the relevant criteria are the individual approaches and the similarities between them.

Similarly to the learner modelling mechanism, the feedback prioritisation and the grouping for collaboration mechanisms could be applied to other ELEs. Feedback prioritisation is likely to be an issue for any ELE that aims to give more freedom to the learners rather than structure their interactions to avoid the prioritisation problem. Also, as collaborative activities for less structured domains (and not only) often involve discussing similarities and differences between various approaches, we believe the criteria and the mechanism we propose for grouping learners is relevant for other ELEs (and potentially, other types of learning environments as well).

To ensure optimal coverage of the knowledge base, we have developed an adaptive modelling mechanism that updates the knowledge base when new relevant information becomes available. This work could be characterised as adapting to a changing environment, which falls within the area of *Intelligent Systems*. Therefore, our research contributes to this area. As the knowledge base adaptation aims to ensure adequate personalised responses to learners' behaviour, this work also contributes to the field of *Artificial Intelligence in Education*.

Our proposed technique for maintaining an optimal coverage of the knowledge base is also applicable to other ELEs and domains, as one of the issues in this type of learning environments is that not everything can be modelled at the beginning and that learners' behaviour changes over time. Moreover, we believe the high level idea could also be applied beyond educational systems, to knowledge based systems where new approaches can develop. For example, in an inconsistency detection systems, when a new potential inconsistency is suspected, some checks could be performed and if successful, it should be added to the knowledge base.

The holistic picture of the contribution evolves around the learner modelling mechanism that has been developed with the purpose of providing student diagnosis during as well as at the end of a task. It also has been enriched with adaptivity over time and has been deployed for educational purposes such as feedback prioritisation and grouping for collaborative activities.

8.3 Directions for Future Research

In this section we present several directions for future work, grouped by our three main topics: the learner modelling mechanism, the exploitation of the learner model and the approach for enriching the knowledge base.

The Learner Modelling Mechanism. In Chapter 4, and more specifically in Section 4.3.2 we presented a high level modelling process, in which the Learner Model has three parts: a short-term model (STM), a task long-term model (Task LTM) and a domain long-term model (Domain LTM). The research presented in this thesis focused on the first two and the most natural extension of this work would be to integrate a long-term domain model with the existing ones. This would require expert knowledge about mathematical generalisation concepts and how each task relates to these concepts, i.e. which concepts are covered by each task. Besides this knowledge, a modelling mechanism needs to be developed that updates the status of the students' domain knowledge based on their actions for solving particular tasks, which are currently stored in the Task LTM. Consequently, the modelling mechanism that has as input the information in the Task LTM should output a Domain LTM that reflects the concepts learned when solving the particular tasks, allowing also for different degrees of coverage corresponding to situations when the tasks are only partially solved.

Another extension of the learner modelling mechanism could be to include another layer that monitors the development of learners' skills related to exploratory learning, such as choosing variables and testing models. As mentioned in Chapter 4 Section 4.5, some information about exploratory learning skills is already included in the definition of the strategies in the task model. For example, the variables are defined in the strategies, allowing to identify whether learners choose the right ones or not. Other information, however, may need to be included; for example, in *eXpresser* testing a model involves pressing a 'Play' button. The frequency and timing of using this button could be used to model if learners are purposefully testing their models.

In Chapter 4 Section 4.4.3 we presented a rule-based approach for off-task behaviour detection. A direction for future research is to improve this mechanism or even to develop a detection model based on learners' actions. This could be accomplished by collecting and analysing usage data and experimenting with various educational data mining techniques.

Our approach, like most user modelling approaches, depends on the activity of the user. To be able to infer what a user is doing, a minimal level of activity is needed. Although it is easy to detect inactivity, it is less straight-forward how to remedy it. Moreover, the remedy depends on the cause of inactivity which needs to be further investigated. For example, the stage within a task at which inactivity occurs could shed some light on the matter.

The similarity metrics we use to identify what strategies a learner is using currently have fixed weights that maximise the identification of structural similarity. A direction for future research is to investigate the effect of changing the weights depending on the stage within a task to identify aspects that are more relevant at that particular stage. For example, the focus could be on the generality of the approach rather than the structure.

Another possible research extension is related to Open Learner Models (Bull et al., 2006), as these have been shown to encourage learners reflection on their own learning (Bull et al., 2006; Bull and Kay, 2007) and reflection is an important part of discovery learning (de Jong, 2006).

Exploiting the Learner Model. In Chapters 5 and 6 we presented two ways in which the information in the learner model can be used to address pedagogical issues. The first looked at feedback prioritisation, while the second addressed grouping for collaborative activities. Both approaches could be extended and improved.

The feedback prioritisation mechanism could be improved to cover all possible combinations of learner characteristics rather than the most common ones. This could be done for example by using the generalisation capabilities of neural networks.

As one of the outcomes of our evaluation was that experts do not always agree, a useful extension of our feedback prioritisation mechanism would be to allow the teachers to change the pairwise comparisons, which, in turn, would change the prioritisation output. As we found the disagreement to occur only for collaborative activities rather than individual ones, it would be interesting to investigate if this is related to the added complexity of the collaborative activity.

The grouping for collaboration mechanism could be extended to include constraints such as to avoid having certain learners in the same group. Such constraints could be identified by conducting a study with teachers, asking their opinion on the usefulness of such constraints, as well as asking them on the type of support and flexibility they would like from such a grouping mechanism.

A limiting aspect of our grouping for collaboration mechanism is the "black-and-white" way of defining similarities. For example, if a learner has used a combination of two strategies, his approach is defined as being equally similar to the two strategies. The Learner Model, however, contains information about the similarity of the learner's approach to the two strategies, from which it can be inferred if the learner's approach is more similar to one of them. A way of improving the mechanism would be to define a continuous scale that allows us to define different degrees of similarity.

Enriching the Knowledge Base. In Chapter 7 Section 7.3 we presented a study in which we identify the limits of our similarity metrics in identifying when a modified pattern is no longer recognisable as similar to the original pattern. One aspect that could improve the metrics capability to better identify modified patterns is to develop a mechanism that identifies sub-patterns. For our particular situation, the best approach would probably be to make use of the task knowledge, i.e. the basic patterns associated with various strategies, and check if a certain pattern is composed by repeating any of the basic patterns for a particular task.

A very useful extension of our proposed mechanism is to exploit the information about inefficient cases and new strategies in an automatic way to either incorporate it in the feedback and/or inform the teachers and allow them to author feedback.

One other direction for future research is to test our mechanism with more tasks and to adjust the threshold values if necessary. This could be extended even further to develop an evolving mechanism that would identify the optimal values for the thresholds after a new task has been introduced.

Another direction of further research is to develop an approach to populate the knowledge base of a new task by identifying strategies from learners' actions. This would be beneficial when teachers define new tasks for the more advanced students, while working on other tasks with the less advanced ones and thus, do not have the time (and may not even have the need) for defining strategies for that particular task. A possible approach would be to cluster the strategies used by the learners and present the cluster-centres to the teacher for approval before they are stored. Another possibility would be to use the most frequent used strategies within a particular cluster.

8.4 Concluding Remarks

This thesis has investigated the possibility of modelling exploratory learning behaviour in such a way as to enable support during as well as at the end of an exploratory task. Our evaluation indicates that despite the challenges of exploratory learning, modelling exploratory behaviour is possible. Moreover, it facilitates the development of personalised approaches to other pedagogical issues such as feedback prioritisation and grouping for collaboration. The solution presented in this thesis is one way of modelling learner behaviour and exploiting this knowledge in a particular exploratory learning environment (ELE). The results of this approach can be extended to other ELEs and provide evidence that learner modelling in exploratory learning deserves further investigation.

Bibliography

Aamodt, A. and Plaza, E. (1994). Case-based reasoning: Foundation issues: Methodological variations, and system approaches. *AI Communications*, 7(1):39–59.

Abraham, W. C. and Robins, A. (2005). Memory retention - the synaptic stability versus plasticity dilemma. *Trends in Neurosciences*, 28(2):73 – 78.

Acquaviva, M., Benini, M., and Trombetta, A. (2005). Short-term content adaptation in web-based learning systems. In Hamza, M. H., editor, *Proceedings of Web Technologies, Applications, and Services*, pages 198–203. IASTED/ACTA Press.

Ainsworth, S. (1999). The functions of multiple representations. *Computers & Education*, 33(2-3):131–152.

Akman, V., Bouquet, P., Thomason, R., and Young, R., editors (2001). *Modeling and Using Context: Third International and Interdisciplinary Conference, CONTEXT 2001*, volume 2116 of *LNAI*, Dundee, UK. Springer.

Aksoy, S. and Haralick, R. M. (2001). Feature normalisation and likelihood-based similarity measures for image retrieval. *Pattern Recognition Letters*, 22(5):563–582.

Aleven, V. (2003). Using background knowledge in case-based legal reasoning: A computational model and an intelligent learning environment. *Artificial Intelligence*, 150(1-2):183–237.

Anand, S. S. and Mobasher, B. (2007). Contextual recommendation. In Berendt, B., Hotho, A., Mladenic, D., and Semeraro, G., editors, *Proceedings of WebMine 2006: From Web to Social Web: Discovering and Deploying User and Content Profiles*, volume 4737 of *Lecture Notes in Computer Science*, pages 142–160. Springer, Berlin-Heidelberg.

Andaloro, G. and Bellomonte, L. (1998). Student knowledge and learning skill modeling in the learning environment 'forces'. *Computers & Education*, 30(3-4):209 – 217.

Anderson, J. R., Corbett, A. T., Koedinger, K. R., and Pelletier, R. (1995). Cognitive tutors: Lessons learned. *Journal of the Learning Sciences*, 4(2):167–207.

Angelov, P. (2003). An evolutionary approach to fuzzy rule-based model synthesis using indices for rules. *Fuzzy Sets and Systems*, 137(3):325 – 338.

Angelov, P. and Buswell, R. (2002). Identification of evolving fuzzy rule-based models. *IEEE Transactions on Fuzzy Systems*, 10(5):667–677.

Angelov, P., Lughofer, E., and Zhou, X. (2008). Evolving fuzzy classifiers using different model architectures. *Fuzzy Sets and Systems*, 159(23):3160 – 3182.

Anguita, D. (2001). Smart adaptive systems: State of the art and future directions of research. In *Proceedings of the 1st European Symposium on Intelligent Technologies, Hybrid Systems and Smart Adaptive Systems, EUNITE 2001*, pages 1–4.

Baker, R., Corbett, A., Gowda, S., Wagner, A., MacLaren, B., Kauffman, L., Mitchell, A., and Giguere, S. (2010). Contextual slip and prediction of student performance after use of an intelligent tutor. In De Bra, P., Kobsa, A., and Chin, D., editors, *Proceedings of User Modeling, Adaptation, and Personalization*, volume 6075 of *Lecture Notes in Computer Science*, pages 52–63. Springer Berlin / Heidelberg.

Baker, R. S., Corbett, A. T., Koedinger, K. R., and Wagner, A. Z. (2004). Off-task behavior in the cognitive tutor classroom: when students "game the system". In *CHI '04: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 383–390, New York, NY, USA. ACM.

Baker, R. S. J. d. (2007). Modeling and understanding students' off-task behavior in intelligent tutoring systems. *Proceedings of ACM CHI 2007: Computer-Human Interaction*, pages 1059–1068.

Beal, C. R., Qu, L., and Lee, H. (2006). Classifying learner engagement through integration of multiple data sources. In *Proceedings of the 21st national conference on Artificial intelligence - Volume 1*, pages 151–156. AAAI Press.

Beck, J. E. and Woolf, B. P. (1998). Using a learning agent with a student model. In Goettl, B., Halff, H., Redfield, C., and Shute, V., editors, *Proceedings of the*

4th International Conference on Intelligent Tutoring Systems, volume 1452 of *Lecture Notes in Computer Science*, pages 6–15. Springer.

Bednarz, N., Kieran, C., and Lee, L. (1991). Approaches to algebra: Perspectives for research and teaching. In Bednarz, N., Kieran, C., and Lee, L., editors, *Approaches to algebra: Perspectives for research and teaching*, pages 3–12. Kluwer Academic Publishers, Dordrecht.

Biswas, G., Schwartz, D., Leelawong, K., and Vye, N. (2005). Learning by teaching: A new agent paradigm for educational software. *Applied Artificial Intelligence*, 19(3-4):363–392.

Bliss, J. (1994). From mental models to modelling. In Mellar, H., Bliss, J., Boohan, R., Ogborn, J., and Tompsett, C., editors, *Learning with artificial worlds: Computer based modeling in the Curriculum*, pages 27–32. London: The Falmer Press.

Boehm, B. (1986). A spiral model of software development and enhancement. *ACM SIGSOFT Software Engineering Notes*, 11(4):14–24.

Brown, A. and Palincsar, A. (1989). Guided, cooperative learning and individual knowledge acquisition. In Resnick, L., editor, *Knowing, learning, and instruction*, pages 393–451. Hillsdale, NJ: Lawrence Erlbaum Associates.

Brusilovsky, P. (1994). Student model centered architecture for intelligent learning environments. In *Proceedings of Fourth International Conference on User Modeling*, pages 31–36. User Modeling Inc.

Brusilovsky, P. (2001). Adaptive hypermedia. *User Modeling and User-Adapted Interaction*, 11(1-2):87–110.

Bull, S. and Kay, J. (2007). Student models that invite the learner in: The smili open learner modelling framework. *International Journal of Artificial Intelligence in Education*, 17(2):89–120.

Bull, S., Quigley, S., and Mabbott, A. (2006). Computer-based formative assessment to promote reflection and learner autonomy. *Engineering Education: Journal of the Higher Education Academy Engineering Subject Centre*, 1(1):8–18.

Bunt, A. (2001). *On Creating a Student Model to Assess Effective Exploratory Behaviour in an Open Learning Environment.* MSc thesis, The University of British Columbia.

Bunt, A. and Conati, C. (2003). Probabilistic student modelling to improve exploratory behaviour. *User Modelling and User-Adaptive Interaction*, 13(3):269–309.

Bunt, A., Conati, C., and Muldner, K. (2004). Scaffolding self-explanation to improve learning in exploratory learning environments. In Lester, J., Vicari, R., and Paraguaçu, F., editors, *Proceedings of the 7th International Conference on Intelligent Tutoring Systems*, volume 3220 of *Lecture Notes in Computer Science*, pages 656–667. Springer.

Carroll, J. M. (2000). *Making Use: Scenario-Based Design of Human-Computer Interactions.* The MIT Press, 1st edition.

Carroll, J. M. and Rosson, M. B. (1992). Getting around the task-artifact cycle: how to make claims and design by scenario. *ACM Transactions on Information Systems*, 10(2):181–212.

Chan, F. T. S., Chung, S. H., and Choy, K. L. (2006). Optimisation of order fulfillment in distribution network problems. *Journal of Intelligent Manufactoring*, 17(3):307–319.

Chi, M. T. H. (2000). Self-explaining expository texts: The dual processes of generating inferences and repairing mental models. In Glaser, R. A., editor, *Advances in Instructional Psychology*, volume 5, pages 161–237. Erlbaum.

Chi, M. T. H., Bassok, M., Lewis, M., Reimann, P., and Glaser, R. (1989). Self-explanations: How students study and use examples in learning to solve problems. *Cognitive Science*, 13(2):145–182.

Chin, D. N. (2001). Empirical evaluation of user models and user-adapted systems. *User Modeling and User-Adapted Interaction*, 11(1-2):181–194.

Chinn, C. A. and Brewer, W. F. (1993). The role of anomalous data in knowledge acquisition: A theoretical framework and implications for science instruction. *Review of Educational Research*, 63(1):1–49.

Christodoulopoulos, C. E. and Papanikolaou, K. (2007). Investigation of group formation using low complexity algorithms. In *Proceedings of the Workshop on Personalisation in e-learning environments at individual and group level, User Modelling*, pages 57 – 60.

Chun, S.-H. and Park, Y.-J. (2005). Dynamic adaptive ensemble case-based reasoning: application to stock market prediction. *Expert Systems with Application*, 28(3):435–443.

Cockburn, A. (2008). Using both incremental and iterative development. *CrossTalk*, 21(5):27–30.

Conati, C., Gertner, A. S., and VanLehn, K. (2002). Using bayesian networks to manage uncertainty in student modeling. *User Modelling and User-Adaptive Interaction*, 12(4):371–417.

Conati, C. and Merten, C. (2007). Eye-tracking for user modeling in exploratory learning environments: An empirical evaluation. *Knowledge-Based Systems*, 20(6):557 – 574. Special Issue On Intelligent User Interfaces.

Conlon, T. and Pain, H. (1996). Persistent collaboration: a methodology for applied AIED. *International Journal of Artificial Intelligence in Education*, 7(3):219–252.

Crary, M., Nozick, L. K., and Whitaker, L. R. (2002). Sizing the US destroyer fleet. *European Journal of Operational Research*, 136(3):680–695.

Daradoumis, T., Guitert, M., Giménez, F., Marquès, J. M., and Lloret, T. (2002). Supporting the composition of effective virtual groups for collaborative learning. In *Proceedings of ICCE'02*, pages 332 – 336. IEEE Computer Society.

de Barros, J.-C. and Dexter, A. L. (2007). On-line identification of computationally undemanding evolving fuzzy models. *Fuzzy Sets and Systems*, 158(18):1997 – 2012.

de Jong, T. (2006). Scaffolds for scientific discovery learning. In Elen, J. and Clark, D., editors, *Handling complexity in learning environments: research and theory*, pages 107–128. London: Elsevier Science Publishers.

de Jong, T. and van Joolingen, W. R. (1998). Scientific discovery learning with computer simulations of conceptual domains. *Review of Educational Research*, 68(2):179–202.

de los Angeles Constantino-Gonzalez, M., Suthers, D. D., and de los Santos, J. G. E. (2003). Coaching web-based collaborative learning based on problem solution differences and participation. *International Journal of Artificial Intelligence in Education*, 13(2-4):263–299.

Dogantekin, E., Dogantekin, A., Avci, D., and Avci, L. (2010). An intelligent diagnosis system for diabetes on linear discriminant analysis and adaptive network based fuzzy inference system: LDA-ANFIS. *Digital Signal Processing*, 20(4):1248 – 1255.

Dorfler, W. (1991). Forms and means of generalisation in mathematics. In Bishop, A., Mellin-Olsen, S., and van Dormolen, J., editors, *Mathematical Knowledge: Its Growth through Teaching*, pages 63–85. Kluwer Academic Publishers, Dordrecht.

Druin, A. (1999). Cooperative inquiry: developing new technologies for children with children. In *CHI '99: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 592–599, New York, NY, USA. ACM.

Druin, A. (2002). The role of children in the design of new technology. *Behaviour and Information Technology*, 21(1):1–25.

Duke, R. and Graham, A. (2007). Inside the letter. *Mathematics Teaching Incorporating Micromath*, 200:42–45.

Ertay, T., Ruan, D., and Tuzkaya, U. R. (2006). Integrating data envelopment analysis and analytic hierarchy for the facility layout design in manufacturing systems. *Information Sciences*, 176(3):237–262.

Fernando, C. (2010). Neuronal replicators solve the stability-plasticity dilemma. In *GECCO '10: Proceedings of the 12th annual conference on Genetic and evolutionary computation*, pages 153–154, New York, NY, USA. ACM.

Fister, I., Mernik, M., and Filipic, B. (2010). A hybrid self-adaptive evolutionary algorithm for marker optimization in the clothing industry. *Applied Soft Computing*, 10(2):409 – 422.

Frías-Martínez, E., Magoulas, G. D., Chen, S. Y., and Macredie, R. D. (2005). Modeling human behavior in user-adaptive systems: Recent advances using soft computing techniques. *Expert Systems with Applications*, 29(2):320–329.

Gadkar, K. G., Mehra, S., and Gomes, J. (2005). On-line adaptation of neural networks for bioprocess control. *Computers & Chemical Engineering*, 29(5):1047 – 1057.

Geraniou, E., Mavrikis, M., Hoyles, C., and Noss, R. (2008). A constructionist approach to mathematical generalisation. In Joubert, M., editor, *Proceedings of the British Society for Research into Learning Mathematics*, volume 28 (2) of *BSRLM Proceedings*.

Geraniou, E., Mavrikis, M., Kahn, K., Hoyles, C., and Noss, R. (2009). Developing a microworld to support mathematical generalisation. In *PME 33: International Group for the Psychology of Mathematics Education*, volume 29 (3), pages 49–56.

Glotin, H., Warnier, P., Dandurand, F., Dufau, S., Lété, B., Touzet, C., Ziegler, J. C., and Grainger, J. (2010). An adaptive resonance theory account of the implicit learning of orthographic word forms. *Journal of Physiology Paris*, 104(1-2):19 – 26.

Gogoulou, A., Gouli, E., Boas, G., Liakou, E., and Grigoriadou, M. (2007). Forming homogeneous, heterogeneous and mixed groups of learners. In *Proceedings of the Workshop on Personalisation in e-learning environments at individual and group level, User Modelling 2007*, pages 33 – 40.

Good, J. and Robertson, J. (2006). CARSS: A framework for learner-centred design with children. *International Joural of Artificial Intelligence in Education*, 16(4):381–413.

Gouli, E., Gogoulou, A., Papanikolaou, K. A., and Grigoriadou, M. (2006). An adaptive feedback framework to support reflection, guiding and tutoring. In Magoulas, G. D. and Chen, S. Y., editors, *Advances in Web-Based Education: Personalized Learning Environments*, pages 178–202. Information Science Publishing.

Graf, S. and Bekele, R. (2006). Forming heterogeneous groups for intelligent collaborative learning systems with ant colony optimization. In Ikeda, M., Ashley, K. D., and Chan, T.-W., editors, *Proceedings of Intelligent Tutoring Systems*, volume 4053 of *LNCS*, pages 217–226. Springer.

Gregory, R. P. (1984). Streaming, setting and mixed ability grouping in primary and secondary schools: some research findings. *Educational Studies*, 10(3):209–226.

Grigoriadou, M., Kornilakis, H., Papanikolaou, K. A., and Magoulas, G. D. (2002). Fuzzy inference for student diagnosis in adaptive educational systems. In Vlahavas, I. and Spyropoulos, C., editors, *Methods and Applications of Artificial Intelligence: Proceedings of the 2nd Hellenic Conference on AI, SETN2002*, volume 2308 of *Lecture Notes in Artificial Intelligence*, pages 191–202. Berlin: Springer-Verlag.

Grigoriadou, M., Mitropoulos, D., Samarakou, M., Solomonidou, C., and Stavridou, E. (1999a). Methodology for the design of educational software in mathematics and physics for secondary education. In *Proceedings of the Computer-based Learning in Science Conference*, page B3.

Grigoriadou, M., Samarakou, M., Mitropoulos, D., Rigoutsos, A., Stavridou, E., and Solomonidou, C. (1999b). Vectors in physics and mathematics. In

Proceedings of the International Conference on Technology and Education (ICTE), Edinburgh, pages 71–73.

Grossberg, S. (1987). Competitive learning: From interactive activation to adaptive resonance. *Cognitive Science*, 11(1):23 – 63.

Gutiérrez, S., Mavrikis, M., and Pearce, D. (2008). A learning environment for promoting structured algebraic thinking in children. In *Proceedings of the 8th IEEE International Conference on Advanced Learning Technologies*, pages 74–76. IEEE.

Hamker, F. H. (2001). Life-long learning cell structures–continuously learning without catastrophic interference. *Neural Networks*, 14(4-5):551 – 573.

Han, S.-G., Lee, S.-G., and Jo, S. (2005). Case-based tutoring systems for procedural problem solving on the www. *Expert Systems with Applications*, 29(3):573–582.

Harel, G. and Tall, D. (1991). The general, the abstract and the generic in advanced mathematics. *For the Learning of Mathematics*, 11(1):38–42.

Hausmann, R. G. M. and VanLehn, K. (2007). Explaining self-explaining: A contrast between content and generation. In Luckin, R., Koedinger, K., and Greer, J., editors, *Proceedings of Artificial Intelligence in Education*, pages 417–4243. Amsterdam, Netherlands: IOS Press.

Haynes, S. R., Purao, S., and Skattebo, A. L. (2004). Situating evaluation in scenarios of use. In *CSCW '04: Proceedings of the 2004 ACM conference on Computer supported cooperative work*, pages 92–101, New York, NY, USA. ACM.

Healy, L., Hoelzl, R., Hoyles, C., and Noss, R. (1994). Messing up. *Micromath*, 10(1):14–16.

Hirayama, J., Yoshimoto, J., and Ishii, S. (2006). Balancing plasticity and stability of on-line learning based on hierarchical bayesian adaptation of forgetting factors. *Neurocomputing*, 69(16-18):1954 – 1961.

Ho, W. (2008). Integrated analytic hierarchy process and its applications – A literature review. *European Journal of Operational Research*, 186(1):211–228.

Hossain, S. and Brooks, L. (2008). Fuzzy cognitive map modelling educational software adoption. *Computers & Educa*, 51:1569–1588.

Hoyles, C. and Küchemann, D. (2002). Students understanding of logical implication. *Educational Studies in Mathematics*, 51(3):193–223.

Huang, M.-J., Huang, H.-S., and Chen, M.-Y. (2007). Constructing a personalized e-learning system based on genetic algorithm and case-based reasoning approach. *Expert Systems with Applications*, 33(3):551–564.

Igel, C. and Kreutz, M. (2003). Operator adaptation in evolutionary computation and its application to structure optimization of neural networks. *Neurocomputing*, 55(1-2):347 – 361.

"Iterative Design" (2006). A dictionary of business and management. Retrieved May 16, 2010 from Encyclopedia.com: http://www.encyclopedia.com/doc/1O18-iterativedesign.html.

Johnson, D. W. and Johnson, R. T. (1993). Cooperative learning and feedback in technology-based instruction. In Dempsey, J. and Sales, G., editors, *Interactive Instruction and Feedback*, pages 133–157. Englewood Cliffs, NJ.

Joines, J. A., King, R. E., and Culbreth, C. T. (1996). A comprehensive review of production-oriented manufacturing cell formation techniques. *International Journal of Flexible Manufacturing Systems*, 3(3-4):225–265.

Kerly, A., Ellis, R., and Bull, S. (2008). CALMsystem: A conversational agent for learner modelling. *Knowledge-Based Systems*, 21(3):238 – 246.

Keselman, A. (2003). Supporting inquiry learning by promoting normative understanding of multivariable causality. *Journal of Research in Science Teaching*, 40(9):898–921.

King, J. R. and Nakornchai, V. (1982). Machine-component group formation in group technology: Review and extension. *International Journal of Production Research*, 20(2):117 – 133.

Kirschner, P., Sweller, J., and Clark, R. E. (2006). Why minimal guidance during instruction does not work: An analysis of the failure of constructivist, discovery, problem-based, experiential and inquiry-based teaching. *Educational Psychologist*, 41(2):75–86.

Klahr, D. (2000). *Exploring Science: The Cognition and Development of Discovery Processes*. Cambridge, London: MIT Press.

Klahr, D. and Dunbar, K. (1988). Dual space search during scientific reasoning. *Cognitive Science*, 12(1):1–48.

Koksal, G. and Egitman, A. (1998). Planning and design of industrial engineering education quality. *Computers & Industrial Engineering*, 35(3-4):639–642.

Kolodner, J. L. (1993). *Case-Based Reasoning.* Morgan Kaufmann Publishers, Inc., 2nd edition.

Korkmaz, I., Gökcen, H., and Çetinyokus, T. (2008). An analytic hierarchy process and two-sided matching based decision support system for military personnel assignment. *Information Sciences*, 178(14):2915–2927.

Krutchen, P. (2003). *Rational unified process - An introduction.* Pearson Education, 2nd edition.

Küchemann, D. (1991). *Childrens Understanding of Mathematics :11 - 16.* John Murray, London.

Küchemann, D. and Hoyles, C. (2006). Influences on students' mathematical reasoning and patterns in its development: Insights from a longitudinal study with particular reference to geometry. *International Journal of Science and Mathematics Education*, 4(4):581–608.

Kumar, P., Gopalan, S., and Sridhar, V. (2005). Context enabled multi-CBR based recommendation engine for e-commerce. In *Proceedings of the IEEE International Conference on e-Business Engineering (ICEBE)*, pages 237 – 244. IEEE Press.

Kwak, N. K. and Lee, C. W. (1998). A multicriteria decision-making approach to university resource allocation and information infrastructure planning. *European Journal of Operational Research*, 110(2):234–242.

Kwak, N. K. and Lee, C. W. (2002). Business process reengineering for health-care system using multicriteria mathematical programming. *European Journal of Operational Research*, 140(2):447–458.

Kwon, O. (2006). The potential roles of context-aware computing technology in optimization-based intelligent decision-making. *Expert Systems with Applications*, 31(3):629–642.

Lalioti, V. and Theodoulidis, B. (1995). Use of scenarios for validation of conceptual specifications. In *Proceedings of the Sixth Workshop on the Next Generation of CASE Tools.* Available online at: http://makebelieve.gr/vl/Publications/CAISE95.pdf.

Lam, K. and Zhao, X. (1998). An application of quality function deployment to improve the quality of teaching. *International Journal of Quality and Reliability Management*, 15(4):389–413.

Lawson, A. E. (2002). Sound and faulty arguments generated by pre-service biology teachers when testing hypotheses involving un-observable entities. *Journal of Research in Science Teaching*, 39(3):237–252.

Lecerf, C. (1999). Tackling the stability/plasticity dilemma with double loop dynamic systems. In *Proceedings of ESANN, The 7th European Symposium on Artificial Neural Networks*, pages 153–158.

Lee, C. W. and Kwak, N. K. (1999). Information resource planning for a health-care system using an AHP-based goal programming method. *Journal of Operational Research Society*, 50(12):1191–1198.

Leornard, J. (2001). How group composition influenced the achievement of sixth-grade mathematics students. *Math Think Learn*, 3(2-3):175–200.

Lewis, E. L., Stern, J. L., and Linn, M. C. (1993). The effect of computer simulations on introductory thermodynamics understanding. *Educational Technology*, 33(1):45–58.

Li, L., Yang, Z., Wang, B., and Kitsuregawa, M. (2007). Dynamic adaptation strategies for long-term and short-term user profile to personalize search. In Dong, G., Lin, X., Wang, W., Yang, Y., and Yu, J. X., editors, *Proceedings of APWeb/WAIM*, volume 4505 of *Lecture Notes in Computer Science*, pages 228–240. Springer.

Löhner, S., van Joolingen, W. R., Savelsbergh, E. R., and van Hout-Wolters, B. (2005). Student's reasoning during modelling in an inquiry learning environment. *Computers in Human Behaviour*, 21(3):441–461.

Lynch, C., Ashley, K., Aleven, V., and Pinkwart, N. (2006). Defining "ill-defined domains"; a literature survey. In *Proceedings of the Workshop on Intelligent Tutoring Systems for Ill-Defined Domains at the 8th ITS Conference*, pages 1–10.

Macintyre, H. and Ireson, J. (2002). Within-class ability grouping: placement of pupils in groups and selfconcept. *British Educational Research Journal*, 28(2):249–263.

Malara, N. and Navarra, G. (2003). *ArAl Project: Arithmetic pathways towards favouring pre-algebraic thinking.* Pitagora Editrice, Bologna.

Mallipeddi, R., Mallipeddi, S., and Suganthan, P. N. (2010). Ensemble strategies with adaptive evolutionary programming. *Information Sciences*, 180(9):1571 – 1581.

Manlove, S., Lazonder, A. W., and de Jong, T. (2006). Regulative support for collaborative scientific inquiry learning. *Journal of Computer Assisted Learning*, 22(2):87–98.

Martin, R. C. (2002). *Agile Software Development: Principles, Patterns, and Practices.* Prentice Hall, 1st edition.

Marzano, R. J., Pickering, D. J., and Pollock, J. E. (2005). *Classroom Instruction That Works: Research-Based Strategies for Increasing Student Achievement.* Upper Saddle, NJ: Pearson.

Mason, B. J. and Bruning, R. (2001). Providing feedback in computer-based instruction: What the research tells us. http://dwb.unl.edu/Edit/MB/MasonBruning.html.

Mason, J. (2002). Generalisation and algebra: exploiting childrens' powers. In Haggarty, L., editor, *Aspects of Teaching Secondary Mathematics: Perspectives on Practice*, pages 105–120. Routledge Falmer and the Open University.

Mayer, R. (2004). Should there be a three-strikes rule against pure discovery learning? The case for guided methods of instruction. *American Psychologist*, 59(1):14–19.

McAuley, J. (1972). Machine grouping for efficient production. *Production Engineer*, 51(2):53 – 57.

McCloskey, M. and Cohen, N. J. (1989). Catastrofic interference in connexionist networks: The sequential learning problem. In Bower, G., editor, *The Psychology of Learning and Motivation*, volume 23, pages 109–164. Academic Press, New York.

Merten, C. and Conati, C. (2006). Eye-tracking to model and adapt to user meta-cognition in intelligent learning environments. In *IUI '06: Proceedings of the 11th international conference on Intelligent user interfaces*, pages 39–46, New York, NY, USA. ACM.

Michael, J. A., Haque, M. M., Rovick, A. A., and Evens, M. (1989). The pathophysiology tutor: a first step towards a smart tutor. In Maurer, H., editor, *Computer Assisted Learning*, pages 390–400. Berlin: Springer Verlag.

Mitrovic, A. (2003). An intelligent SQL tutor on the web. *International Journal of Artificial Intelligence in Education*, 13(2-4):197–243.

Montalvo, I., Izquierdo, J., Prez-Garca, R., and Herrera, M. (2010). Improved performance of PSO with self-adaptive parameters for computing the optimal design of Water Supply Systems. *Engineering Applications of Artificial Intelligence*, 23(5):727 – 735.

Morales Gamboa, R. (2000). *Exploring Participative Learner Modelling and Its Effects on Learner Behaviour*. PhD thesis, Univ. of Edinburgh.

Moss, J. and Beatty, R. (2006). Knowledge building in mathematics: Supporting collaborative learning in pattern problems. *International Journal of Computer-Supported Collaborative Learning*, 1(4):441–465.

National Science Foundation (2000). Inquiry: Thoughts, views, and strategies for the k-5 classroom. *Foundations*, 2:1–5.

Newell, A. and Simon, H. A. (1972). *Human problem solving*. Englewook Cliffs, NJ:Pentice-Hall.

Nickles, T. (1998). Kuhn, historical philosophy of science, and case-based reasoning. *Configurations*, 6(1):51–85.

Noss, R., Healy, L., and Hoyles, C. (1997). The construction of mathematical meanings: Connecting the visual with the symbolic. *Educational Studies in Mathematics*, 33(2):203–233.

Noss, R. and Hoyles, C. (1996). *Windows on Mathematical Meanings: Learning cultures and computers*. Kluwer Academic Publishers, Dordrecht.

Noss, R., Hoyles, C., Mavrikis, M., Geraniou, E., Gutierrez-Santos, S., and Pearce, D. (2009). Broadening the sense of 'dynamic': a microworld to support students mathematical generalisation. *Special Issue of the The International Journal on Mathematics Education (ZDM): Transforming Mathematics Education through the Use of Dynamic Mathematics Technologies*, 41(4):493–503.

Ounnas, A., Davis, H. C., and Millard, D. E. (2009). A framework for semantic group formation in education. *Educational Technology and Society*, 12(4):4355.

Ozdemir, M. S. and Gasimov, R. N. (2004). The analytic hierarchy process and multiobjective 0–1 faculty course assignment. *European Journal of Operational Research*, 157(2):398408.

Papert, S. (1993). *Mindstorms: children, computers and powerful ideas*. Basic-Books, New York.

Partovi, F. Y. (2006). An analytic model for locating facilities strategically. *Omega*, 34(1):41–55.

Petre, M., Blackwell, A. F., and Green, T. R. G. (1998). Cognitive questions in software visualization. In Stasko, J., Domingue, J., Brown, M., and Price, B., editors, *Software visualization: Programming as a multi-media experience*, pages 453–480. MIT Press.

Piaget, J. (1950). *The Psychology of Intelligence.* New York: Routledge.

Pintrich, P. and Schunk, D. (2002). *Motivation in Education: Theory, Research and Applications.* Prentice-Hall, Englewood Cliffs, NJ, 2nd edition.

Pollalis, Y. A. and Mavrommatis, G. (2009). Using similarity measures for collaborating groups formation: A model for distance learning environments. *European Journal of Operational Research*, 193(2):626 – 636.

Rieber, L. P. (2004). Microworlds. In Jonassen, D., editor, *Handbook of research for educational communications and technology*, pages 583–603. Mahwah, NJ: Lawrence Erlbaum Associates.

Robins, A. (2004). Sequential learning in neural networks: A review and a discussion of pseudorehearsal based methods. *Intelligent Data Analysis*, 8(3):301–322.

Rodríguez-Serrano, J. A., Perronnin, F., Sánchez, G., and Lladós, J. (2010). Unsupervised writer adaptation of whole-word HMMs with application to word-spotting. *Pattern Recognition Letters*, 31(8):742 – 749.

Romero, C., Ventura, S., de Bra, P., and de Castro, C. (2003). Discovering prediction rules in AHA! courses. In Brusilovsky, P., Corbett, A., and de Rosis, F., editors, *Proceedings of the 9th International Conference on User Modeling*, volume 2702 of *Lecture Notes in Computer Science*, pages 25–34. Springer.

Roscoe, R. D. and Chi, M. (2007). Understanding tutor learning: Knowledge-building and knowledge-telling in peer tutors explanations and questions. *Review of Educational Research*, 77(4):534–574.

Saaty, T. L. (1980). *The Analytic Hierarchy Process.* New York: McGraw-Hill.

Scaife, M. and Rogers, Y. (1998). Kids as informants: telling us what we didn't know or confirming what we knew already? In Druin, A., editor, *The design of children's technology*, pages 27–50, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Schmitt, C., Dengler, D., and Bauer, M. (2003). Multivariate preference models and decision making with the MAUT machine. In Brusilovsky, P., Corbett, A., and de Rosis, F., editors, *Proceedings of the 9th International Conference on User Modeling*, volume 2702 of *Lecture Notes in Artificial Intelligence*, pages 297–302. Springer.

Schoenfeld, A. H., Smith, J. P., and Arcavi, A. (2002). Learning: The microgenetic analysis of one student's evolving understanding of a complex subject matter domain. In Glaser, R., editor, *Advances in instructional psychology*, volume 4, pages 55–175. Hillside, NJ: Erlbaum.

Selim, H. M., Askin, R. G., and Vakharia, A. J. (1998). Cell formation in group technology: Review, evaluation and directions for future research. *Computers & Industrial Engineering*, 34(1):3 – 20.

Sharkey, N. E. and Sharkey, A. J. (1995). An analysis of catastrophic interference. *Connection Science*, 7(3-4):301–330.

Shinno, H., Yoshioka, H., Marpaung, S., and Hachiga, S. (2006). Quantitative SWOT analysis on global competitiveness of machine tool industry. *Journal of Engineering Design*, 17(3):251–258.

Simon, H. A. and Lea, G. (1974). Problem solving and rule induction: a unified view. In Gregg, L., editor, *Knowledge and Cognition*, pages 105–128. Hillsdale, New York: Lawrence Erlbaum.

Slavin, R. E. (2003). When and why does cooperative learning increase achievement. In Daniels, H. and Edwards, A., editors, *The RoutledgeFalmer Reader in Psychology of Education*, volume 1, pages 271 – 293. RoutledgeFalmer.

Soloway, E., Guzdial, M., and Hay, K. E. (1994). Learner-centered design: the challenge for HCI in the 21st century. *Interactions*, 1(2):36–48.

Soloway, E., Jackson, S. L., Klein, J., Quintana, C., Reed, J., Spitulnik, J., Stratford, S. J., Studer, S., Jul, S., Eng, J., and Scala, N. (1996). Learning theory in practice: Case studies of learner-centered design. In *Proceedings of CHI96*, pages 189–196. ACM Press.

Sommerville, I. (2001). *Software Engineering.* Addison Wesley Publishers Ltd., Harlow, England, 5th edition.

Stathacopoulou, R., Grigoriadou, M., Magoulas, G. D., and Mitropoulos, D. (2003). A neuro-fuzzy approach in student modeling. In Brusilovsky, P., Corbett, A., and de Rosis, F., editors, *Proceedings of User Modeling*, volume 2702 of *Lecture Notes in Artificial Intelligence*, pages 337–341. Springer.

Stathacopoulou, R., Grigoriadou, M., Samarakou, M., and Mitropoulos, D. (2007). Monitoring students' actions and using teachers' expertise in implementing and evaluating the neural network-based fuzzy diagnostic model. *Expert Systems with Applications*, 32(4):955–975.

Stathacopoulou, R., Magoulas, G. D., Grigoriadou, M., and Samarakou, M. (2005). Neuro-fuzzy knowledge processing in intelligent learning environments for improved student diagnosis. *Information Sciences*, 170(2-4):273–307.

Stottler, R. H. and Ramachandran, S. (1999). Case-based reasoning approach to internet intelligent tutoring systems (ITS) and ITS authoring. In *Proceedings of the Twelfth International Florida Artificial Intelligence Research Society Conference*, pages 181–186. AAAI Press.

Ting, C.-Y. and Phon-Amnuaisuk, S. (2009). Factors influencing the performance of dynamic decision network for INQPRO. *Computers & Education*, 52(4):762 – 780.

Tong, M.-W., Yang, Z.-K., and Liu, Q.-T. (2010). A novel model of adaptation decision-taking engine in multimedia adaptation. *Journal of Network and Computer Applications*, 33(1):43 – 49.

van der Meij, J. and de Jong, T. (2006). Supporting students' learning with multiple representations in a dynamic simulation-based learning environment. *Learning and Instruction*, 16(3):199–212.

van Joolingen, W. R. (1999). Cognitive tools for discovery learning. *International Journal of Artificial Intelligence in Education*, 10(3):385–397.

van Joolingen, W. R. and de Jong, T. (1997). An extended dual search space model of scientific discovery learning. *Instructional Science*, 25(5):307–346.

VanLehn, K. (1988). Student modeling. In Polson, M. and Richardson, J., editors, *Foundations of Intelligent Tutoring Systems*, pages 55–78. Hillsdale, NJ: Erlbaum.

VanLehn, K., Lynch, C., Schultz, K., Shapiro, J. A., Shelby, R. H., and Taylor, L. (2005). The Andes physics tutoring system: Lessons learned. *International Journal of Artificial Intelligence in Education*, 15(3):147–204.

Vanlehn, K., Siler, S., Murray, C., Yamauchi, T., and Baggett, W. B. (2003). Why do only some events cause learning during human tutoring. *Cognition and Instruction*, 21(3):209–249.

Veermans, K. and van Joolingen, W. R. (1998). Using induction to generate feedback in simulation based discovery learning environments. In Goettl, B., Halff, H., Redfield, C., and Shute, V., editors, *Proceedings of the 4th International Conference on Intelligent Tutoring Systems*, volume 1452 of *Lecture Notes in Computer Science*, pages 196–205. Springer.

Veermans, K. and van Joolingen, W. R. (2004). Combining heuristics and formal methods in a tool for supporting simulation-based discovery learning. In Lester, J., Vicari, R., and Paraguaçu, F., editors, *Proceedings of the 7th International Conference on Intelligent Tutoring Systems*, volume 3220 of *Lecture Notes in Computer Science*, pages 217–226. Springer.

Veermans, K. H. (2003). *Intelligent support for discovery learning.* PhD thesis, University of Twente.

Vrettos, S. and Stafylopatis, A. (2001). A fuzzy rule-based agent for web retrieval-filtering. In Zhong, N., Yao, Y., Liu, J., and Ohsuga, S., editors, *Proceedings of Web Intelligence*, volume 2198 of *Lecture Notes in Computer Science*, pages 448–453. Springer.

Vygotsky, L. S. (1978). *Mind and society: The development of higher mental processes.* Cambridge, MA: Harvard University Press.

Wang, S. S., Treat, T. A., and Brownell, K. D. (2008). Cognitive processing about classroom-relevant contexts: Teachers' attention to and utilization of girls' body size, ethnicity, attractiveness, and facial affect. *Journal of Educational Psychology*, 100(2):473–489.

Warren, E. and Cooper, T. J. (2008). Generalising the pattern rule for visual growth patterns: actions that support 8 year olds' thinking. *Educational Studies in Mathematics*, 67(2):171–185.

Watson, I. (1997). *Applying Case-Based Reasoning: Techniques for Enterprise Systems.* Morgan Kaufmann Publishers Inc.

Webb, N. M., Baxter, G., and Thompson, L. (1997). Teachers' grouping practices in fifth-grade science classrooms. *Elementary School Journal*, 98(2):91–124.

Wenger, E. (1987). *Artificial intelligence and tutoring systems: computational and cognitive approaches to the communication of knowledge.* Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

Wiggins, G. (2008). Feedback: how learning occurs. http://www.authenticeducation.org/bigideas/article.lasso?artId=61.

Wilhelm, P. and Beishuizen, J. J. (2003). Content effects in self-directed inductive learning. *Learning and Instruction*, 13(4):381–402.

Winters, N. and Mor, Y. (2008). IDR: A participatory methodology for interdisciplinary design in technology enhanced learning. *Computers & Education*, 50(2):579–600.

Xu, Z., Xuan, J., Shi, T., Wu, B., and Hu, Y. (2009). Application of a modified fuzzy artmap with feature-weight learning for the fault diagnosis of bearing. *Expert Systems with Applications*, 36(6):9961 – 9968.

Xydeas, C., Angelov, P., Chiao, S.-Y., and Reoullas, M. (2006). Advances in classification of EEG signals via evolving fuzzy classifiers and dependant multiple HMMs. *Computers in Biology and Medicine*, 36(10):1064 – 1083.

Yang, S. J. H. and Shao, N. W. Y. (2007). Enhancing pervasive web accessibility with rule-based adaptation strategy. *Expert Systems with Applications*, 32(4):1154 – 1167.

Yeo, N. C., Lee, K. H., Venkatesh, Y. V., and Ong, S. H. (2005). Colour image segmentation using the self-organizing map and adaptive resonance theory. *Image and Vision Computing*, 23(12):1060 – 1079.

Yerushalmy, M. (1991). Student perceptions of aspects of algebraic function using multiple representation software. *Journal of Computer Assisted Learning*, 7(1):42–57.

Yin, Y. and Yasuda, K. (2005). Similarity coefficient methods applied to the cell formation problem: a comparative investigation. *Computers & Industrial Engineering*, 48(3):471 – 489.

Zhang, Y., Ng, J. M., and Low, C. P. (2009). A distributed group mobility adaptive clustering algorithm for mobile ad hoc networks. *Computer Communications*, 32(1):189 – 202.

Zopounidis, C. and Doumpos, M. (2002). Multicriteria classification and sorting methods: A literature review. *European Journal of Operational Research*, 138(2):229–246.

Zukerman, I. and Albrecht, D. W. (2001). Predictive statistical models for user modeling. *User Modeling and User-Adapted Interaction*, 11(1-2):5–18.

Zurita, G., Nussbaum, M., and Salinas, R. (2005). Dynamic grouping in collaborative learning supported by wireless handhelds. *Educational Technology and Society*, 8(3):149–161.

Appendix A

List of Publications

The research presented in this thesis was partly published prior to thesis submission.

Research from **Chapter 4** was published in:

M. Cocea, G.D. Magoulas, “Identifying strategies in users exploratory learning behaviour for mathematical generalisation,” in *Artificial Intelligence in Education: Building Learning Systems that Care: From Knowledge Representation to Affective Modelling, Proceedings of the 14th International Conference on Artificial Intelligence in Education (AIED 2009)*, 2009, pp. 626-628

M. Cocea, G.D. Magoulas, “Task-oriented modeling of learner behaviour in exploratory learning for mathematical generalisation,” in *Proceedings of the 2nd International Workshop on Intelligent Support for Exploratory Environments (ISEE09), in conjunction with the 14th International Conference on Artificial Intelligence in Education (AIED 2009)*, 2009 pp. 16-24

M. Cocea, S. Gutierrez-Santos and G.D. Magoulas, “Challenges for Intelligent Support in Exploratory Learning: the case of ShapeBuilder,” in *Proceedings of the 1st International Workshop on Intelligent Support for Exploratory Environments (ISEE08), in conjunction with the third European Conference on Technology-Enhanced Learning (EC-TEL ’08)*, 2008, pp. 61-70

M. Cocea, “A Modelling Framework for Constructivist Learning in Exploratory Learning Environments,” in *Young Researchers Track Proceedings of the 9th International Conference on Intelligent Tutoring Systems, ITS 2008*, 2008, pp. 157-166

M. Cocea, “Learner Modelling in Exploratory Learning for Mathematical Generalisation,” in *Proceedings of Adaptive Hypermedia and Adaptive Web-Based Systems, 5th International Conference, AH 2008*, 2008, pp. 394-399

Research from **Chapter 5** was published in:

M. Cocea, G.D. Magoulas, "Context-dependent Personalised Feedback Prioritisation in Exploratory Learning for Mathematical Generalisation," in *User Modeling, Adaptation, and Personalization, 17th International Conference, UMAP 2009, (formerly UM and AH)*, 2009, pp. 271-282

An early version of **Chapters 4 and 5** was published in:

M. Cocea, G.D. Magoulas, "Hybrid Model for Learner Modelling and Feedback Prioritisation in Exploratory Learning," *The International Journal of Hybrid Intelligent Systems*, vol. 6, no. 4, pp. 211-230, 2009

M. Cocea, G.D. Magoulas, "Combining Intelligent Methods for Learner Modelling in Exploratory Learning Environments," in *Proceedings of the 1st International Workshop on Combinations of Intelligent Methods and Applications (CIMA 2008), in conjunction with the 18th European Conference on Artificial Intelligence (ECAI-08)*, 2008, pp. 13-18

Research from **Chapter 6** was published in:

M. Cocea, G.D. Magoulas, "Group Formation for Collaboration in Exploratory Learning using Group Technology Techniques," in *The 14th International Conference on Knowledge-Based and Intelligent Information & Engineering Systems, KES 2010*, 2010, pp. 103-113

Research from **Chapter 7** was published in:

M. Cocea, S. Gutierrez-Santos, G.D. Magoulas, "Adaptive Modelling of Users' Strategies in Exploratory Learning using Case-based Reasoning," in *The 14th International Conference on Knowledge-Based and Intelligent Information & Engineering Systems, KES 2010*, 2010, pp. 124-134 **[Won Best Student Paper Award]**

M. Cocea, G.D. Magoulas, "Identifying User Strategies in Exploratory Learning with Evolving Task Modelling," in *The 5th IEEE International Conference on Intelligent Systems, IS 2010*, 2010, pp. 13-18

All these publications are available at:
`http://coceam.myweb.port.ac.uk/publications.html`