A Modelling Framework for Constructivist Learning in Exploratory Learning Environments

Mihaela Cocea

London Knowledge Lab, Birkbeck College
23-29 Emerald St, WC1N 3QS London (UK)
mihaela@dcs.bbk.ac.uk

Abstract. Exploratory learning has been proved to be beneficial for learning when guidance and support is provided. This is a challenging issue as balance between freedom and control is a strong requirement in this context. In this paper the challenges involved are discussed together with the way they affect the process of learner modelling. We discuss previous attempts to model the learner in exploratory learning, and how they fall short, and propose as an alternative a learner modelling framework for exploratory learning environments that adopts the constructionist view of learning. We investigate the application of this framework in the domain of mathematical generalisation and discuss how the model can potentially support feedback generation. An example is discussed in detail, and an approach based on case-based reasoning and soft computing is presented to represent and process short-term and long-term learner behaviour and knowledge in the learner model.

Keywords: learner modelling, exploratory learning, feedback generation, mathematical generalisation, case-based reasoning, soft computing.

1. Introduction

Constructivism [16] sees learning as an active, constructive process in which knowledge is built and structured gradually. Exploratory/discovery learning supports this view of learning and has been argued to be particularly beneficial [2] in terms of providing opportunities for acquiring deep conceptual and structural knowledge. However pure discovery learning without any guidance and support is hardly beneficial [6]. The main challenge with this approach is to balance freedom with control: learners should be given enough freedom so that they can actively engage in constructing models and they should be offered enough guidance in order to assure that their constructions lead to useful knowledge [10].

In contrast to previous attempts [1], [14], [15], here we advocate an approach that extends user modelling in Exploratory Learning Environments (ELEs) by reflecting and supporting the constructionist learning process. Since the focus is on the process, interaction analysis [11] plays an essential part in learner modelling. Typically it starts with filtering raw data in order to extract some indicators related to the quality of the learning process. These indicators can be used for several purposes; in our case, the main purpose is the regulation of the learning process through feedback, while a secondary purpose is to inform teachers about students' learning process and progression.

Our approach will be integrated into an ELE for mathematical generalisation, called ShapeBuilder [4] which is under development in the context of the MiGen project[1]. This ELE aims to facilitate 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 concepts that lies at the centre of mathematical thinking.

In this paper, we present our framework for learner modelling. This follows principles of constructivism and supports provision of feedback in order to guide the learner towards useful and sound knowledge construction. The following section gives an overview of the research challenges involved and of previous research in ELEs. Section 3 presents the proposed framework. Section 4 illustrates the approach through one example and discusses knowledge representation using case-based reasoning. Section 5 presents the expected contributions of our research and concludes the paper.

2. Research Challenges and Related Work

Besides the clear and well-acknowledged challenge of balancing freedom with guidance, as mentioned in the previous section, there are other issues that make the process of learner modelling in ELEs 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 support can be provided during and at the end of knowledge construction.

(b) *Value of correct vs. incorrect actions*. In most e-Learning systems, feedback is related to correctness or incorrectness of answers/actions, while in ELEs learner's 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. Actually, one of the advantages of ELEs is that learners are given the opportunity to realise their own mistakes and learn from them; thus, rather then 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 abstract knowledge and forms of (re)presentation in the system.* ELEs have different ways of (re)presenting and exploring models that should gradually help the learner build abstract knowledge. Each part of the model and each type of exploration (e.g. changing parameters, creating new models, testing models etc.) contributes to this process. Identification of relevant abstract knowledge is needed as well as its representation in the learner model.

(d) *Identification of underlying strategies* from actions or sequences of actions. Sometimes is neither realistic nor feasible to include all possible outcomes (correct or incorrect) and ways to achieve them when modelling an extensive knowledge domain. Thus, a different approach to what is included in the knowledge structure is required; rather than storing complete information about a

[1] The MiGen project is funded by ESRC/EPSRC (TLRP); website: http://www.migen.org

task or expert knowledge, key information with informative educational value could be stored, such as strategies for approaching the (sub-) 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 abovementioned challenges, a classical 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. Previous attempts in this area include: (a) the use of heuristics to guide the learning process in a physics domain [15]; (b) Bayesian networks in a mathematical functions domain [1]; (c) neuro-fuzzy systems for student diagnosis in a physics domain [14]. The idea of intelligent support was tackled in [15] using induction and deduction, whilst templates were used to generate feedback; no learner model was used. The second approach addresses “effective exploration” [1], 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 classical overlay models. Two of the challenges previously mentioned, i.e. what to model and the difficulty of determining the (in-) correctness of an action, were also addressed. The third approach 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.

3. A Framework for Learner Modelling in ELE

ELEs involve knowledge discovery by means of constructive activities and the emphasis is on the process rather that the knowledge itself and thus, 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.

In our approach, the ELE includes two components (see Fig. 1): a domain and a task model. The domain model includes high level learning outcomes related to the domain and considers that each learning outcome can be achieved by exploring several tasks. The task model includes different types of information: (a) strategies of approaching the task which could be correct, incorrect or partially correct; (b) outcomes of the exploratory process and solutions to specific questions associated with each (sub-) task; (c) landmarks, i.e. relevant aspects or critical events occurring during the exploratory process; (d) context, i.e. reference to this particular task.

The main issues addressed in the proposed framework are the following: (a) What interactions are relevant and how can they be extracted from the flow of raw data and transformed into indicators? (b) 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? (c) How should the learner model be updated in order to reflect both the current knowledge and the evolution of knowledge? (d) Using the learner model, how can personalised feedback be provided to support the constructionist process and inform the teacher?

Below we present how the proposed framework can address these challenges.

(a) *The learner's interaction model.* 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. For example, to answer this question in the context of MiGen, teachers' expertise will be used and observations of children working with the ShapeBuilder in the context of small-scale exploratory studies. These are expected to provide an understanding of relevant actions that lead to both correct and incorrect solutions and would be represented as cases or constraints.

(b) *Representing the evolution of the learning process.* The proposed learner modelling process presented in Fig. 1 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. [7]), 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 domain and the task 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.

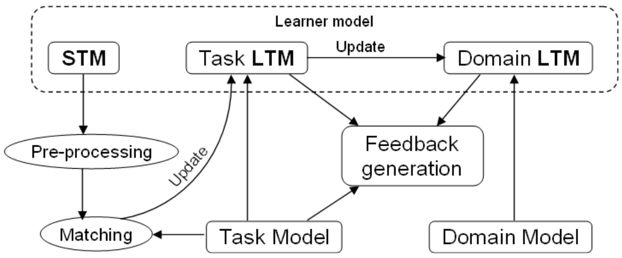


Fig. 1. Learner modelling process.

(c) *Updating the model.* The learner model update is illustrated in Fig. 1. During the (sub-) 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 relevant strategies, landmarks, outcomes and solutions for a learner in the current task or subtask. Knowledge of the domain and teachers' expertise together with findings from pilot studies will be used to derive these aspects for every (sub-) task and define a 'light-weight' model for mathematical generalisation. For pre-processing, a technique similar to *episodes identification and association* [8] can be used and comparisons will be made using *fuzzy similarity measures*. After matching, the Task LTM is updated; then, the degree of meeting the learning outcome that was explored through the (sub-) task is updated in the Domain LTM. Thus, the modelling process reflects the constructionist approach of incremental knowledge acquisition. As the three components of the learner model have different structure and functionality, there are different updating mechanisms and timeframes for each of the three parts of the model. The STM will be updated frequently with the interactions of the learner; the Task LTM will be updated less frequently based on the processed data from the STM and the data from the Task Model; the Domain LTM will be updated even less frequently based on information from the Task LTM.

(d) *Usage of the model for personalized feedback.* As illustrated in Fig. 1, the feedback is generated based on Task LTM, Task Model and Domain LTM. The learner modelling process supports two types of feedback: during the exploration process and at the end of certain processing stages. The first one aims to guide the learner in gradually constructing the knowledge, while the second one is more related to outcomes of the exploration and specific solutions.

Our framework will be implemented and validated in the context of MiGen project, through trials with pupils and testing in classrooms.

4. An Illustrative Example: the Pond Tiling Task in ShapeBuilder

To illustrate our approach we use an example from the mathematical generalisation domain, and a task called 'pond tiling', which is common in the English school curriculum and expects learners to produce a general expression for finding out how many tiles are required for surrounding *any* rectangular pond. The high level learning outcome in the Domain Model is the students' ability to perform structural reasoning. In order to achieve this, subtasks can be explored, e.g. construct a pond of fixed dimensions, surround it with tiles and determine how many are required; generalise the structure using variables. This task is performed in ShapeBuilder [4], which allows the 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. Constants, variables and numeric expressions lead to specific constructions/models, while icon variables and expressions using them 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" [9].

An overview of the Task Model for the pond tiling task is presented in Fig. 2. It includes: (a) strategies, e.g. thinking in terms of width and height, thinking in terms of areas; (b) landmarks, e.g. creating a rectangle that has the height and width of the pond incremented by two as an indication of the 'areas strategy'; (c) outcomes (e.g. model built, numerical answer for a particular pond) and solution, i.e. a general algebraic expression (e.g. for the 'areas strategy': *(width+2)* (height+2) – width * height*); (d) context, i.e. reference to the task.

In relation to the first challenge, i.e. the identification of relevant interactions from the flow of raw data and their transformation into indicators, we have explored relevant interactions during the pond tiling task (captured through small-scale exploratory studies), and worked on how to represent them in the Task Model. Case-base reasoning was chosen as an appropriate approach as it offers flexibility in the representation of information and has been proved to be successfully combined with other approaches for retrieval of relevant cases that handle imprecision, e.g. neural networks [12], fuzzy quantifiers [17].

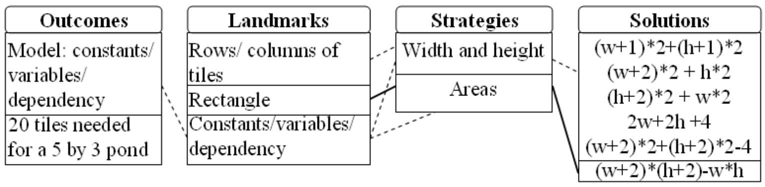


Fig. 2. Partial task model (slots connected by solid lines correspond to the example in the text).

Our case base includes two types of cases: *simple* and *composite*. Below, we present an example for each type in relation to the models that the learners construct, i.e. the pond and the way to surround it. The cases are also distinguished on a *specific vs. general* dimension, as the generality is the goal, but it is rarely achieved without a start from the specific.

Table 1 presents an example for a simple case as attribute-value pairs representing a rectangle; the structure applies to all types of shapes. In the category column, the various types of information included in the case are presented:

(a) The first category of information is related to the *shape*; for the pond tiling task, the only type of shape used is rectangle.

Table 1. Simple case presented as attribute-value pairs.

Category	Attributes	Possible values
Shape	shape type	rectangle
Dimensions of shape	width type	constant(c)/variable(v)/icon variable(iv)/numeric expressions (n_exp)/ expression using icon variable(s) (iv_exp)
	width value	numeric value
	height type	c/v/iv/n_exp/iv_exp
	height value	numeric value
Dependency relation	dependent on	reference to case (e.g. w(c_i))
Value relation	relation	expression (e.g. w=w(c_i))
Time relations	subsequent to	reference to case (e.g. c_j)
	followed by	reference to case (e.g. c_k)
Part-of relation	part-of strategy 1	0 (if it is not part-of) /1 (if it is)
	part-of strategy 2	0/1
	part-of strategy *n*	0/1

(b) Each shape has one ore more *dimensions*; for a rectangular shape the dimensions are width and height; for other shapes, e.g. T-shape, there are three dimensions: width, height and thickness. Each dimension has a type and a value. The type refers to the way that dimension of the shape is constructed and can have four values: constant, variable, icon variable, numeric expression or expression using icon variable(s). Values of dimensions are numerical.

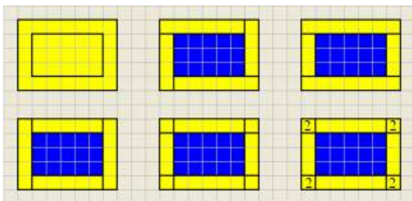
(c) *Dependency relations* can be defined between the current case and another case(s). These relations are present when at least one of the dimensions of the shape is of icon variable or expression using icon variable(s) type. The attribute value is a reference to the specific case and to the dimension(s) of the case that the current case depends on. A case can include 0, 1 or many dependency relations.

(d) *Value relations* can be related to or be independent of dependency relations; their value is an expression that defines the relation. For example, the expression $w=w(c_i)$ means that the width of the current case is equal to the width of case c_i. A case can include none, one or more value relations.

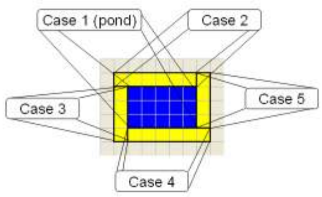
(e) *Time relations* are important for the composite cases; they provide a reference to the previous and the following case; the current case is *subsequent to* and *followed by* the cases referred. A case can include none or one of each of the two mentioned types of time relations. These relations are useful in organizing the cases, e.g. a string of cases ordered in time could be obtained from them that would define the current context or a particular strategy.

(f) *Part-of relations* refer to the participation of a simple case to a strategy (represented as a composite case). It takes binary values (1 if the case is part of a specific strategy; 0 if not). The same case could be part of more than one strategy.

The composite cases represent strategies and consist of instances of simple cases. Strategies were derived from small-scale exploratory studies of pupils using the ShapeBuilder and some examples are illustrated in Fig. 3a. The "2" flag in the strategy shown in the bottom right of Fig. 3a indicates that two tiles are overlapping. In the first strategy (top, left), named the *area strategy*, two steps are involved: constructing the pond and then constructing a rectangle that overlaps the pond and has an extra row/column on each side. The second strategy (see Fig. 3b), named the *spiral strategy*, includes five steps: constructing the pond and then creating four bars (arranged horizontally or vertically depending on the case), each of a length that is equal to the value of the corresponding dimension of the pond plus one.



(a)



(b)

Fig. 3. (a) Strategies for the pond tiling task; (b) Simple cases of the spiral strategy

Depending on the degree of generality, there are two types of composite cases: *specific* and *general*. Specific cases refer to surroundings that cannot be generalised and include value relations, but no dependency relations; the general cases refer to surroundings that can be generalised and are 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 abovementioned aspects apply to all simple cases that form the composite case with the exception of the simple case representing the pond. An illustration of each type is provided in Tables 2 and 3 for two strategies: the area and spiral strategy respectively. In Table 2, the specific case for the area strategy (strategy 1) does not include a dependency relation, while the general case includes one; the general part also has both dimensions as expression using icon variable(s) type. Case 1, i.e. the pond, is part of both strategies (as well as the other possible strategies), while Case 2 (the rectangle over pond) is only part of the area strategy.

Table 2. Area strategy: specific and general composite case.

	Case1 (c_1: pond)	**Case2 (c_2: rectangle over pond)**	
Attributes	Values	Values (specific)	Values (general)
shape type	rectangle	rectangle	rectangle
width type	c/v	c/v/n_exp	iv_exp
width value	5	7	7
height type	c/v	c/v/n_exp	iv_exp
height value	3	5	5
dependent on			$w(c_1)$; $h(c_1)$
relation1		$w=w(c_1)+2$	$w=w(c_1)+2$
relation2		$h=h(c_1)+2$	$h=h(c_1)+2$
Subsequent to		c_1	c_1
Followed by	c_2	c_3	c_3
part-of strategy 1	1	1	1
part-of strategy 2	1	0	0
part-of strategy *n*	1	0	0

Table 3 partially presents a specific and a general composite case for the spiral strategy (strategy 2). Case 1 is the pond, Case 2 is the bar of tiles for the width of the pond and Case 3 is the bar of tiles for the height of the pond. The complete strategy includes two more simple cases: Case 4 and Case 5 that would have the same entries as Case 2 and Case 3, respectively, with the exception of the time relation attribute; the cases for each rectangle in strategy 2 are displayed in Fig. 3b.

Table 3. Spiral strategy: partial specific and general composite case.

	Case1 (c_1: pond)	**Case2 (c_2: width tiling)**		**Case3 (c_3: height tiling)**	
Attributes	Values	Values (specific)	Values (general)	Values (specific)	Values (general)
shape type	rectangle	rectangle	rectangle	rectangle	rectangle
width type	c/v	c/v/n_exp	iv_exp	c/v/n_exp	iv_exp
width value	5	6	6	1	1
height type	c/v	c/v/n_exp	iv_exp	c/v/n_exp	iv_exp
height value	3	1	1	4	4
dependent on			$w(c_1)$		$h(c_1)$
value relation1		$w=w(c_1)+1$	$w=w(c_1)+1$	$h=h(c_1)+1$	$h=h(c_1)+1$
subsequent to		c_1	c_1	c_2	c_2
Followed by	c_2	c_3	c_3	c_4	c_4
part-of strategy 1	1	0	0	0	0
part-of strategy 2	1	1	1	1	1
part-of strategy *n*	1	0	0	0	0

The order of the simple cases (denoting a time relation) within the spiral strategy can vary depending on the starting point of the surrounding (the corner of the pond).

For example, in Fig. 3b we illustrated the surrounding starting from the up-right corner. Other possibilities correspond to the other three corners and the corresponding order is shifted by one case each time as we move anticlockwise; for example if the tiling starts from the bottom-left corner the order would be Case 1, Case 4, Case 5, Case 2 and Case 3. The same strategy could occur with the clockwise surrounding.

In relation to the second and the third challenges, i.e. how to represent the evolution of the learning process and how to update the model, the following process is taking place. As the learner constructs his/her model, raw data are stored in the STM. They are *pre-processed* and the transformed data are *matched* to the cases from the Task Model; any identified strategies (e.g. area strategy) together with landmarks (e.g. rectangle over pond), outcomes (e.g. the specific/ general construction) and context (pond tiling task) are stored or updated in the Task LTM. Finally, the degree of meeting the learning outcome that was explored through the (sub-) task is *updated* in the Domain LTM. For example, at the end of the "generalise the structure with variables" subtask, the knowledge associated with variables manipulation, which is considered an important step in the process of developing mathematical reasoning and generalisation ability, is updated in the Domain LTM. Thus, the evolution of the learning process is reflected in the learner model. Nevertheless, depending on the context of the (sub-) task, some attributes are more relevant than others in retrieving cases from the case base. For example, in the pond tiling task, when the learner is constructing a specific (as opposed to general) tiling of the pond, the value relation attribute is more relevant, while when dealing with a general tiling, the dependency relation attribute is more important. To address this issue we intend to use a combined approach of case-based reasoning and neural networks (NN) for the matching process. Local feature weighting [12] will be used; the neural network is trained to learn the local feature weights as opposed to global weights where a single weight vector is used across the whole domain [13]. Thus, the variation in importance of the different attributes of a case depending on the context is taken into consideration. This approach is more robust due to the generalisation capacities of the NN that can produce weights even if there is no exact match to the cases.

In relation to the fourth research challenge, i.e. how to provide personalized feedback, in the pond tiling task feedback could be provided in relation to different aspects. For example, if the learner has surrounded the pond following a strategy that does not generalise well, the feedback can suggest resizing the pond, which would result in "messing up" [5] the model, and encourage the learner to reflect on what is missing in order to make the solution general. When feedback needs to be provided in relation to more than one aspects, such as in the case of a learner who did not include the corners of the pond in the model (see Fig. 3a, bottom row, middle) and at the same time her construction is specific, two aspects should be addressed: (a) the rules of pond tiling; (b) the fact that the solution is not general, which could be handled as suggested in the previous paragraph. In this situation a decision is required about the order of feedback provision. To this end, an approach based on multicriteria decision making [3] will be used. In the above example, the aim is to achieve the learning goal, i.e. structural reasoning, the criteria are the fact that the four corners are missing (stored as a landmark) and that the solution is specific (as opposed to general) (identified from the strategy) and the alternatives are the two aspects that require feedback. The outcome will be an ordering of alternatives in terms of their priorities.

5. Concluding remarks and contribution

In exploratory learning learner's knowledge is built gradually as a result of active participation in the learning process. In this paper, we proposed a novel framework for user modelling that reflects the constructionist learning approach, and proposed a mechanism for knowledge representing and updating the user model. Our approach employs a Task Model and combines case-based reasoning with other intelligent technologies in order to improve retrieval of relevant cases and to establish priorities in feedback provision when multiple feedbacks are required. Lastly, we briefly described how the framework can be used in an ELE for mathematical generalisation.

References

1. Bunt, A., Conati, C.: Probabilistic Student Modeling to Improve Exploratory Behaviour. Journal of User Modeling and User-Adapted Interaction. 13(3), 269-309 (2003)
2. de Jong, T., van Joolingen, W.R.: Scientific discovery learning with computer simulations of conceptual domains. Review of Educational Research, 68, 179-202 (1998)
3. Ghinea, G.; Magoulas, G.D., Siamitros, C.: Multicriteria decision making for enhanced perception based multimedia communication. IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans 35(6), 855-866 (2005)
4. Gutierrez, S, Mavrikis, M, Pearce, D.: A Learning Environment for Promoting Structured Algebraic Thinking in Children, Proceedings of the 8th IEEE International Conference on Advanced Learning Technologies, forthcoming.
5. Healy L., Hoelzl R., Hoyles C., Noss, R.: Messing Up. Micromath, 10(1), 14-16 (1994)
6. Kirschner, P. A., Sweller, J., Clark, R. E.: 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 (2006)
7. Li, L., Yang, Z., Wang, B., Kitsuregawa, M.: Dynamic Adaptation Strategies for Long-Term and Short-Term User Profile to Personalize Search. APWeb/WAIM, 228-240 (2007)
8. Liu, J., Wong, C.K., Hui, K.K.: An Adaptive User Interface Based on Personalized Learning, IEEE Intelligent Systems, 18(2), 52-57 (2003)
9. Mason, J.: Generalisation and algebra: Exploiting children's powers. In L. Haggarty (ed), Aspects of Teaching Secondary Mathematics: Perspectives on Practice, Routledge Falmer and the Open University, 105-120 (2002)
10. Mayer, R.: Should There Be a Three-Strikes Rule Against Pure Discovery Learning? The Case for Guided Methods of Instruction. American Psychologist, 59(1), 14-19 (2004)
11. Papanikolaou, K., Grigoriadou, M.: Modelling and Externalising Learners' Interaction Behaviour. In Proceedings of the LeMoRe05 workshop, AIED2005, 52-61 (2005)
12. Park, J.H., Im, K.H., Shin, C.-K., Park, S.C.: MBNR: Case-based Reasoning with Local Feature Weighting by Neural Network, Applied Intelligence, 21, 265-276 (2004)
13. Shin, C., Yun, U.T., Kim, H.K., Park, S.C.: A hybrid approach of neural network and memory based learning for data mining, IEEE Trans. on NN, 11 (3), 637–646, (2000)
14. Stathacopoulou, R., Magoulas, G.D., Grigoriadou, M., Samarakou, M.: Neuro-fuzzy knowledge processing in intelligent learning environments for improved student diagnosis, Information Sciences, 170, 273-307 (2005)
15. Veermans, K.H.: Intelligent support for discovery learning, PhD thesis, Twente Univ(2003)
16. Vygotsky, L.S.: Mind and society: The development of higher mental processes. Cambridge, MA: Harvard University Press (1978)
17. Yager, R.R.: Soft Aggregation Methods in Case Based Reasoning, Applied Intelligence, 21, 277-288 (2004)