Combining Intelligent Methods for Learner Modelling in Exploratory Learning Environments

Mihaela Cocea and **George D. Magoulas** [1]

Abstract. Most of the existing learning environments work in well-structured domains by making use of or combining AI techniques in order to create and update a learner model, provide individual and/or collaboration support and perform learner diagnosis. In this paper we present an approach that exploits the synergy of case-base reasoning and soft-computing for learner modelling in an ill-structured domain for exploratory learning. We present the architecture of the learner model, the knowledge formulation in terms of cases and illustrate its application in an exploratory learning environment for mathematical generalisation.

1 INTRODUCTION

Several AI techniques have been proposed in intelligent learning environments, such as case-based reasoning [27], [10], bayesian networks [4], [6], neural networks [2], genetic and evolutionary algorithms [24], neuro–fuzzy systems [26], as well as synergistic approaches, such as genetic algorithms and case-based reasoning [13], hybrid rules integrating symbolic rules with neurocomputing [11], and expert systems with genetic algorithms [18].

Exploratory Learning Environments (ELEs) belong to a particular class of learning environments built on the principles of constructivism paradigm for teaching and learning. ELEs place the emphasis on the opportunity to learn through free exploration and discovery rather than guided tutoring. This approach has proved to be beneficial for learners in terms of acquiring deep conceptual and structural knowledge. However, discovery learning without guidance and support appears to be less effective than step-by-step guiding learning environments [16]. To this end, an understanding of learner's behaviour and knowledge construction is needed [22].

Most existing ELEs use simulations as a way of actively involving learners in the learning process (e.g. [28], [14]) and exploit cognitive tools [29] to support their learning. Few such systems model learner's knowledge/skills; for example [4] and [6] use bayesian networks and [26] combines neural networks with fuzzy representation of knowledge. Another category of ELEs is closer to the constructivist approach by allowing the learner to construct their own models rather than explore a "predefined" one. Compared to conventional learning environments (even environments that use simulations), this type of ELE requires approaches to learner modelling that would be able to capture and model the useful interactions that take place as learners construct their models.

In this paper, we present an approach to learner modelling in ELEs (suitable for both exploring simulations and constructing models) that combines case-based reasoning with other AI techniques. The subsequent section briefly introduces the application domain, namely mathematical generalisation, and the ELE used, called ShapeBuilder, and discusses the challenges involved in performing learner modelling. Section 3 presents a conceptual framework for the learner modelling process and describes the case-based formulation. Section 4 illustrates the process with an example, while Section 5 concludes the paper and outlines future work.

2 EXPLORATORY LEARNING FOR MATHEMATICAL GENERALISATION

Mathematical generalisation (MG) is associated with algebra, 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" [20].

However, students do not associate algebra with generalisation as the algebraic language is perceived as been separate from what it represents [15]. To address this problem the ShapeBuilder [8] system, which is an ELE under development in the context of the MiGen project [2], aims to facilitate the correspondence between the models, patterns and structures (visual representations) that the learners build, on one hand, and their numeric, iconic and symbolic representations, on the other hand. ShapeBuilder allows the construction of different shapes [9], e.g. rectangles, L-shapes, T-shapes and supports the three types of representations aforementioned: (a) *numeric representations* that include numbers (constants or variables) and expressions with numbers; (b) *iconic representations* which correspond to icon variables; (c) *symbolic representations* that 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 1a).

Constants, variables and numeric expressions lead to specific constructions/models, while icon variables and expressions using them lead to general ones. 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" [20] (see Figure 1b). It also uses the "messing up" metaphor [12] 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 1c and d), indicating learner's lack of generalisation ability.

When attempting to model the learner in an ELE for such a wide domain as MG, several challenges arise. The main and widely ac-
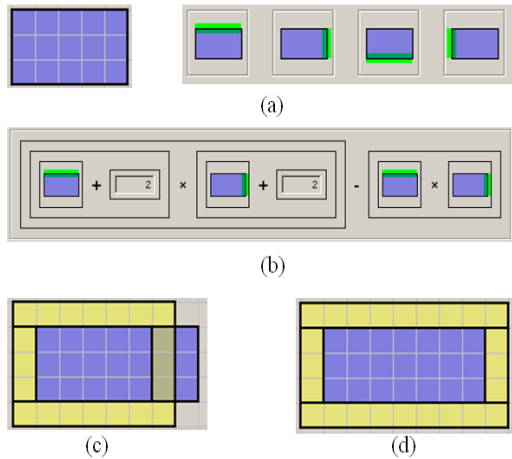
Figure 1. (a) A rectangular shape and its icon variable; (b) an expression using icon variables; (c) "messing up"; (d) general solution that does not "mess up".

knowledged challenge is to balance freedom with control: learners should be given enough freedom so that they can actively engage in activities but they should be offered enough guidance in order to assure that the whole process reflects constructivist learning and leads to useful knowledge [21]. This and some other challenges are illustrated in Table 1 with examples from the domain of MG.

Table 1. Applying learner modelling in ELEs for mathematical generalisation.

| Challenge	Example
Balance between freedom and control	When a learner is trying to produce a general representation, for how long should he be left alone to explore and when does guidance become necessary?
What should be modelled?	Besides learner's knowledge of MG concepts (e.g. use of variables, consistency between representations, etc.), other aspects need to be modelled in order to support the learner during exploration: shapes constructed, relations between shapes, etc.
Do both correct and incorrect actions or behaviours have value?	In exploratory learning it is difficult to categorise actions or learner's explorations into "correct" and "incorrect". Moreover, actions that might lead to incorrect outcomes such as *resizing* can be more valuable for constructivist learning than "correct" actions.
Reasoning about abstract knowledge	Can consistency be inferred from the fact that a learner is checking the correspondence between various forms of representations? If so, is that always true? Are there any exceptions to this rule?
Underlying strategies	As it is neither realistic nor feasible to include all possible outcomes (correct or incorrect) to model the domain of MG, only key information with educational value could be stored, such as strategies in solving a task. The challenge is how to represent and detect them.

3 A CONCEPTUAL FRAMEWORK FOR LEARNER MODELLING

Given the challenges mentioned in Table 1 a conventional learner modelling approach does not fit the purposes of ELEs. Due to the exploratory nature of the activities and the diversity of possible trajectories, flexibility in the representation of information and handling of uncertainty are two important aspects for effectively supporting the

learning process. As case-based reasoning offers flexibility of information representation and soft computing techniques handle uncertainty, a combination of the two is used. Moreover, previous research has proved the benefits of combining case-based reasoning with neural networks [23] and fuzzy quantifiers [30]. In the following subsections, the architecture of the system, the AI components and their role are described.

3.1 The Architecture

The architecture of the "Intelligent" ShapeBuilder is represented in Figure 2. As the learner interacts with the system through the interface, the actions of the learner are stored in the Learner Model (LM) and they are passed to the Interactive Behaviour Analysis Module (IBAM) where they are processed in cooperation with the Knowledge Base (KB); the results are fed into the LM. The Feedback Module (FM) is informed by the LM and the KB and feeds back to the learner through the interface.
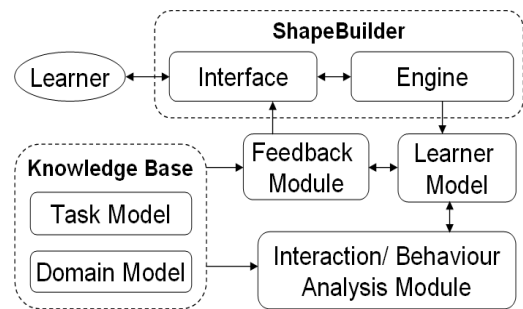
Figure 2. Schematic of an intelligent architecture for ShapeBuilder.

The KB includes two components (see Figure 2): a domain and a task model. The domain model includes high level learning outcomes related to the domain (e.g. using variables, structural reasoning, consistency, etc.) 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) contexts, i.e. reference to particular (sub)tasks.

The IBAM component combines case-based reasoning with soft computing in order to identify what learners are doing and be able to provide feedback as they explore a (sub)task. More specifically, as they are working in a specific subtask, which specifies a certain *context*, their actions are preprocessed, current cases are identified and matched to the cases from the Task Model (the case base). Prior to matching, local feature weighting [23] is applied in order to reflect the importance of the attributes in the current context.

In the FM component, multicriteria decision making [7] will be used to obtain priorities between several aspects that require feedback depending on the context.

3.2 Case-based Knowledge Representation

In case-based reasoning (CBR) [17] 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.

Although CBR has been used successfully in applications for domains like legal reasoning [1], stock market prediction [5], recommender systems [19], and other areas, there is little research on using CBR for e-Learning environments. For example, [10] use CBR in the learner modelling process and call this approach case-based student modelling; while [13] use CBR and genetic algorithms to construct an optimal learning path for each learner. CBR is used also in [27] 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 explicit representation of general knowledge about a domain, but it can also use specific knowledge previously experienced [10]. It also seems promising for improving the effectiveness of complex and unstructured decision making [13] in combination with other computing methods.

In our research, CBR is used in the learner modelling process. The cases contain information describing models that learners should construct using ShapeBuilder. Different strategies in approaching a problem (i.e. constructing a model to meet a particular learning objective) are represented as a series of cases that reflect possible exploratory trajectories of learners as they construct models during the various (sub-)tasks.

Table 2. 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

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. RA_i is a set of relations between attributes and RC_i is a set of relations between C_i and other cases respectively.

The set of attributes is represented as $F_i = \{\alpha_{i_1}, \alpha_{i_2}, \ldots, \alpha_{i_N}\}$. It includes three types of attributes: (a) numeric, (b) variables and (c) binary. Variables refer to different string values that an attribute can take, and binary attributes indicate whether a case can be considered in formulating a particular strategy or not. This could be represented as a "part of strategy" function: $PartOfS_u : C_i \rightarrow \{0, 1\}$,

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

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 2. 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 set of relations between attributes of the current case and attributes of other cases (as well as attributes of the same 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 the set of attributes of the current case F_i. Two types of binary relations are used: (a) 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 $i \neq j$), and means that α_{i_k} depends on (is built upon) 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 built upon the height type of the same case; the width type of a case is built upon the width type of another case, an so on); (b) 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.). The set of relations between attributes is presented in Table 3.

Table 3. 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	$V_{i_1}(RA_{i_{t+1}})$	$(\alpha_{i_k}, \alpha_{j_l}) ; k, l = \overline{v+1, w}; \forall j$
	$\vdots$	$\vdots$
	$V_{i_z}(RA_{i_M})$	$(\alpha_{i_k}, \alpha_{j_l}) ; k, l = \overline{v+1, w}; \forall j$

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). Two relations about order in time are defined: (a) $Prev$ relation indicates the previous case with respect to the current case: $(C_i, C_j) \in Prev$ if $t(C_j) < t(C_i)$ and (b) $Next$ relation indicates the next case with respect to the current case: $(C_i, C_k) \in Next$ if $t(C_i) < t(C_k)$. Each case includes at most one of each of these two relations ($p \leq 2$).

A strategy is defined as $S_u = \{N_u(C), N_u(RA), N_u(RC)\}$, $u = \overline{1, r}$, where $N_u(C_i)$ is a set of cases, $N_u(RA_i)$ is a set of relation between attributes of cases and $N_u(RC_i)$ is a set of relations between cases.

3.3 Comparing Cases, Exploiting Context and Modelling Learning Trajectories

In this section we present three distinctive features of the proposed framework: comparing cases, exploiting context and modelling of learning trajectories.

Comparing cases. The most common definition of similarity is a weighted sum of similarities of attributes of cases [17]:

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

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. In our case, four similarity measures are defined for comparing cases:

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

2. The following metric is used for attributes that are variables:
$V_{IR} = \frac{\sum_{j=1}^{v} g(\alpha_{I_j}, \alpha_{R_j})}{v}$, where g is defined as:

$$g(\alpha_{I_j}, \alpha_{R_j}) = \left\{ \begin{array}{ll} 1 & \text{if } \alpha_{I_j} = \alpha_{R_j} \\ 0 & \text{if } \alpha_{I_j} \neq \alpha_{R_j}, \end{array} \right.$$

3. In a similar way to [25], we define the following metric for comparing relations between attributes: $P_{IR} = \frac{|RA_I \cap RA_R|}{|RA_I \cup RA_R|}$. P_{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.

4. Similarity in terms of relations between cases is defined by $T_{IR} = \frac{|RC_I \cap RC_R|}{|RC_I \cup RC_R|}$, where T_{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: $(\sum_{i=1}^{z} D_{I_i R_i})/z$.
2. Variables: $(\sum_{i=1}^{z} V_{I_i R_i})/z$.
3. Relations between attributes: $(\sum_{i=1}^{z} P_{I_i R_i})/z$.
4. Relations between cases. $(\sum_{i=1}^{z} T_{I_i R_i})/z$.

where z represents the minimum number of cases among the two compared strategies. The strength of similarity between the current strategy and the various stored strategies is defined as the maximum combined similarity of these four measures among the various strategies compared.

Exploiting context. Attributes and relations stored in cases have different relevance depending on the context, which in ShapeBuilder corresponds to different stages of the constructivist learning process that learners go through as they explore the various sub-tasks within a learning activity. Typically, a task includes several sub-tasks, and the activity is sequenced within the system so as to know at any time the current context. As the environment allows the learners to explore, they may "jump" to different stages in the activity sequence.

Context dependence can be taken into account by having different weights for attributes and relations depending on the stage of the learning process within a task or activity. The weights could be obtained through an approach called *local feature weighting* [23] that uses Neural Networks (NNs). The principle of the training algorithm is to reduce the distance between cases of the same class and increase the distance between cases of different classes [23], where the various classes in ShapeBuilder correspond to types of context (stages of the learning process) of the various (sub-)tasks. Thus, a neural network is trained in order to identify the context and several networks (one for each context) are used to provide the context-specific weights. This approach appears to be more robust than other weighting schemes due to the generalisation capacities of the NNs that can produce weights even in imprecise situations [23].

Learning trajectories. A string of cases connected with relations in time yields a knowledge structure that represents learner's explorations/learning trajectory in the ELE during a task or sub-task. Such a learning trajectory is constructed by successively applying $Prev$ and $Next$ relations to C_i in order to get cases previous in time to C_i and cases following C_i, respectively. Comparing trajectories in the KB to the current trajectory (this is useful to provide support and decide on scaffolding techniques) is done in two stages: comparing the past and evaluating the future.

Comparison of the past with respect to a reference point (e.g. a selected case) depends on the depth of the evaluation in terms of samples taken into account and rules than concern comparisons of the past, e.g. IF the actual trajectory is similar to a trajectory in the KB, indicated by a reference case representing a starting point in the past, THEN this trajectory is a past-matching trajectory.

When it comes to evaluating the future of a trajectory, comparison is based on the similarity between the future of a trajectory in the KB with a desired future for the current trajectory. This is expressed by rules of the general form: IF a piece of the future trajectory of a past-matching trajectory resembles the reference starting from a selected case, THEN the reference can be met by applying certain strategies.

As it is not possible to represent all learning trajectories in the KB of an ELE, similarity is measured in terms of convex fuzzy sets, whose width might change depending on the context and the amount of information available, i.e. the current trajectory can be interpreted in more vague way by increasing the width of the fuzzy set. Also if the distance between past and future is large for certain tasks, it does not make sense to evaluate the future carefully. Nevertheless, if the distance to a reference (desired outcome) is small, the future needs to be evaluated accurately. So the depth of the evaluation is measured by a fuzzy time distance set to evaluate both short and long time distances.

4 AN ILLUSTRATIVE EXAMPLE

To illustrate the combination of intelligent methods for learner modelling 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 [8]. The high level learning objective in the Domain Model is to acquire the ability to perform structural reasoning [9]. In order to achieve this, sub-tasks can be explored in ShapeBuilder, e.g. construct a pond of fixed dimensions, surround the pond with tiles and determine how many are required; generalise the structure using icon variables.

Knowledge representation. The Task Model for pond tiling includes: (a) strategies identified in pilot studies [9], e.g. thinking in terms of areas (see Figure 3a) or in terms of width and height (see Figures 3b, c, d, e and f); (b) outcomes, e.g. model built, number of tiles for surrounding a particular pond, and solution, i.e. the general expression (see Figure 3 for the solutions corresponding to each strategy; for the "area strategy" the solution with icon variables is displayed in Figure 1b); (c) landmarks, e.g. for the area strategy: creating a rectangle with height and width greater by 2 than the pond; for the width and height strategies: using rows/column of tiles; slips: several correct actions followed by an incorrect one (e.g. correct surrounding of the pond, partially correct expression, but missing a 2 in the formula); (d) the context of each (sub-)task.

Figure 3. (a) "Area strategy"; (b) "H strategy"; (c) "I strategy"; (d) "Spiral strategy"; (e) "+4 strategy"; (f) "−4 strategy"; (g) Steps and relations of "area strategy"; (h) Steps and relations of "I strategy".

The six strategies and their associated solutions (the general expressions for surrounding any rectangular pond) are displayed in Figures 3(a–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 and Table 5, respectively. The steps and the sets of relations between attributes and between cases are displayed in Figure 3g and Figure 3h, respectively.

A particular order between cases is presented for the "I strategy" in Figure 3h. For the same strategy, the surrounding of the pond could be done in several other different orders; there are $4! = 24$ such possibilities (the pond is always first).

Table 4. 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

There are two types of strategies depending on the degree of generality: 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 cases that form the composite case with the exception of the first case representing the pond. The "area" and the "I strategy" presented previously fall into the category of general strategies.

The strategies displayed in Figure 3 are correct symmetrical "elegant" solutions, but trials with pupils have shown that not all of them

Table 5. 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
⋮	⋮	⋮	⋮	⋮	⋮
α_{i_8}	1	0	0	1	1

use this type of approach [8, 9]. Some pupils surround the pond in a non–systematic manner and with variable degrees of symmetry. Such examples are illustrated in Figure 4.

Comparing cases. To illustrate the operation of similarity measures we use two non–symmetrical examples of surrounding the pond, displayed in Figure 4. The similarity measures are the ones presented in Section 3.3.

The first example (Figure 4a), 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). When comparing it with the "I strategy" $z = 5$ (minimum between 6 and 5) and the combined similarity is: $\frac{\sqrt{1}}{5} + \frac{5}{5} + \frac{7/4}{5} + \frac{10/4}{5} = 2.05$. When comparing with the "+4" strategy, $z = 6$ (minimum between 6 and 9) the combined similarity is: $\frac{\sqrt{5}}{6} + \frac{5+2/3}{6} + \frac{6/4}{6} + \frac{10/4+1/3}{6} = 2.04$. Thus, in this case the learner will be guided towards the "I strategy".

The second example (Figure 4b), 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). When comparing it with the "spiral strategy" as well as the "H strategy", $z = 5$ (minimum between 5 and 5), and the combined similarity is: $\frac{\sqrt{2}}{5} + \frac{4+2/3}{5} + \frac{8/4}{5} + \frac{10/4}{5} = 2.12$. In this situation, when the learner's construction is equally similar to two strategies, the following options could be offered: (a) 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); (b) automatically suggest one of the two in a systematic way, e.g. present the one that occurs more/less often with other learners; (c) inform the teacher about the learner's trajectory and the frequency of strategies and let him/her decide between the two.

Figure 4. Non-symmetrical strategies: (a) combination of 'I' and '+4' strategies; (b) combination of 'spiral' and 'H' strategies.

Exploiting context. 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.

Local feature weighting in this case involves two trained neural networks for each context and applying the weights delivered by the NNs before the matching process.

Learning trajectories. Lets consider the example in Figure 4b and a comparison after C_3. The current trajectory includes the creation of 3 rectangles corresponding to C_1, C_2 and C_3; this trajectory is considered to be far from the desired outcome (surrounding the pond), and thus, the future does not need to be evaluated accurately. At this point with respect to the past, two strategies partially match the learner's current trajectory: "I" (C_1, C_2) and "spiral" (C_1, C_3) strategy; the learner could be left to continue with his/her model construction without intervention. With respect to the future, the desired outcome can be obtained by following one of the two strategies previously identified.

If the comparison takes place after C_5, the trajectory would include the creation of 5 rectangles (C_1 to C_5) and thus it can be concluded that the learner has reached the desired outcome of surrounding the pond. However, in this process the learner did not use any of the desirable strategies, i.e. any of the six strategies presented in Figure 3. At this point in time two trajectories match the past and indicate the future, as before, but now it might be considered pedagogically important to intervene and guide the learner towards a trajectory that corresponds to one of the two identified desirable strategies.

5 CONCLUSIONS

In this paper a learner modelling process involving a combination of intelligent methods was presented for the domain of mathematical generalisation. Case-based reasoning is used in combination with soft computing (fuzzy sets and neural networks) in order to process the models that the learners construct and thus be able to provide feedback while learners work on the task.

Further work includes expanding the conceptual framework by defining a strength as the maximum combined similarity measure (similarity of the past and similarity of the future at a particular distance) for various evaluated trajectories and a reliability index that will reflect the extent to which the similarities can be relied upon to provide the right support.

REFERENCES

[1] V. Aleven, 'Using background knowledge in case-based legal reasoning: A computational model and an intelligent learning environment', *Artificial Intelligence*, **150**(1-2), 183–237, (2003).

[2] J.E. Beck and B.P. Woolf, 'Using a Learning Agent with a Student Model', in *Intelligent Tutoring Systems*, eds., B.P. Goettl, H.M. Halff, C.L. Redfield, and V.J. Shute, volume 1452 of *Lecture Notes in Computer Science*, pp. 6–15. Springer, (1998).

[3] P. Brusilovsky, A.T. Corbett, and F. de Rosis, eds. *User Modeling 2003, 9th International Conference, UM 2003, Johnstown, PA, USA, June 22-26, 2003, Proceedings*, volume 2702 of *Lecture Notes in Computer Science*. Springer, 2003.

[4] A. Bunt and C. Conati, 'Probabilistic Student Modelling to Improve Exploratory Behaviour', *User Modelling and User-Adaptive Interaction*, **13**(3), 269–309, (2003).

[5] S-H. Chun and Y-J. Park, 'Dynamic adaptive ensemble case-based reasoning: application to stock market prediction', *Expert Systems with Application*, **28**(3), 435–443, (2005).

[6] C. Conati, A.S. Gertner, and K. VanLehn, 'Using Bayesian Networks to Manage Uncertainty in Student Modeling', *User Modelling and User-Adaptive Interaction*, **12**(4), 371–417, (2002).

[7] G. Ghinea, G.D. Magoulas, and C. Siamitros, 'Multicriteria decision making for enhanced perception-based multimedia communication', *IEEE Transactions on Systems, Man, and Cybernetics, Part A*, **35**(6), 855–866, (2005).

[8] S. Gutiérrez, M. Mavrikis, and D. Pierce, 'A Learning Environment for Promoting Structured Algebraic Thinking in Children', in *Proceedings of the 8th IEEE International Conference on Advanced Learning Technologies*, (2008). forthcoming.

[9] S. Gutiérrez, D. Pierce, E. Geraniou, and M. Mavrikis, 'Supporting Reasoning and Problem-Solving in Mathematical Generalisation with Dependency Graphs', in *Proceedings of the 5th International Conference on the Theory and Application of Diagrams*, (2008). forthcoming.

[10] S-G. Han, S-G. Lee, and S. Jo, 'Case-based tutoring systems for procedural problem solving on the www', *Expert Systems with Applications*, **29**(3), 573–582, (2005).

[11] I. Hatzilygeroudis and J. Prentzas, 'Using a hybrid rule-based approach in developing an intelligent tutoring system with knowledge acquisition and update capabilities', *Expert Systems with Applications*, **26**(4), 477–492, (2004).

[12] L. Healy, Hoelzl R., Hoyles C., and R. Noss, 'Messing Up', *Micromath*, **10**(1), 14–16, (1994).

[13] M-J. Huang, H-S. Huang, and M-Y. Chen, 'Constructing a personalized e-learning system based on genetic algorithm and case-based reasoning approach', *Expert Systems with Applications*, **33**(3), 551–564, (2007).

[14] C.D. Hulshof, T.H.S. Eysink, S. Loyens, and T. de Jong, 'Zaps: Using interactive programs for learning psychology', *Interactive Learning Environments*, **13**, 39–53, (2005).

[15] J. Kaput, *Technology and Mathematics education*, 515–556, D. Grouws (ed.) Handbook of Research on Mathematics Teaching and Learning, New York: Macmillan, 1992.

[16] P. Kirschner, J. Sweller, and R.E. Clark, '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).

[17] J.L. Kolodner, *Case-Based Reasoning*, Morgan Kaufmann Publishers, Inc., 2nd edn., 1993.

[18] C. Koutsojannis, G. Beligiannis, I. Hatzilygeroudis, C. Papavlasopoulos, and J. Prentzas, 'Using a hybrid ai approach for exercise difficulty level adaptation', *International Journal of Continuing Engineering Education and Life-Long Learning*, **17**(4-5), 256–272, (2007).

[19] P. Kumar, S. Gopalan, and V. Sridhar, 'Context enabled multi-CBR based recommendation engine for e-commerce', in *Proceedings of the IEEE International Conference on e-Business Engineering (ICEBE)*, pp. 237 – 244. IEEE Press, (2005).

[20] J. Mason, *Generalisation and algebra: Exploiting children's powers*, 105–120, L. Haggarty, Aspects of Teaching Secondary Mathematics: Perspectives on Practice, Routledge Falmer and the Open University, 2002.

[21] R. Mayer, '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).

[22] R. Morales Gamboa, *Exploring Participative Learner Modelling and Its Effects on Learner Behaviour*, PhD thesis, Univ. of Edinburgh, 2000.

[23] J.H. Park, K.H. Im, C-K. Shin, and S-C. Park, 'MBNR: Case-Based Reasoning with Local Feature Weighting by Neural Network', *Applied Intelligence*, **21**(3), 265–276, (2004).

[24] C. Romero, S. Ventura, P. de Bra, and C. de Castro, 'Discovering Prediction Rules in AHA! Courses', In Brusilovsky et al. [3], pp. 25–34.

[25] Y. Seo, D. Sheen, and T. Kim, 'Block assembly planning in shipbuilding using case-based reasoning', *Expert Systems with Applications*, **32**(1), 245–253, (2007).

[26] R. Stathacopoulou, M. Grigoriadou, G.D. Magoulas, and D. Mitropoulos, 'A Neuro-fuzzy Approach in Student Modeling', In Brusilovsky et al. [3], pp. 337–341.

[27] R.H. Stottler and S. Ramachandran, 'A 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*, pp. 181–186. AAAI Press, (1999).

[28] J. Swaak, W.R. van Joolingen, and T. de Jong, 'Supporting simulation-based learning; the effects of model progression and assignments on definitional and intuitive knowledge', *Learning and Instruction*, **8**, 235–253, (1998).

[29] W.R. van Joolingen, 'Cognitive tools for discovery learning', *International Journal of Artificial Intelligence and Education*, **10**, 385–397, (1999).

[30] R.R. Yager, 'Soft Aggregation Methods in Case Based Reasoning', *Applied Intelligence*, **21**(3), 277–288, (2004).