Design of teacher assistance tools in an exploratory learning environment for algebraic generalisation

Journal Article

Citation:

Gutierrez Santos, S.; Geraniou, E.; Pearce Lazard, D.; Poulovassilis, A.; (2011)
Design of teacher assistance tools in an exploratory
learning environment for algebraic generalisation –
IEEE Transactions On Learning Technologies (In Press)

Design of Teacher Assistance Tools in an Exploratory Learning Environment for Algebraic Generalisation

Sergio Gutierrez-Santos, Eirini Geraniou, Darren Pearce-Lazard, and Alexandra Poulovassilis

Abstract—The MiGen project is designing and developing an intelligent exploratory environment to support 11–14 year-old students in their learning of algebraic generalisation. Deployed within the classroom, the system also provides tools to assist teachers in monitoring students' activities and progress. This paper describes the design of these Teacher Assistance tools and gives a detailed description of one such tool, focussing in particular on the research challenges faced, and the technologies and approaches chosen to implement the necessary functionalities given the context of the project.

Index Terms—Computer-assisted instruction, mathematics education, teacher support, data visualisation, software architecture patterns.

1 Introduction

The MiGen project is co-designing, developing and evaluating with teachers and teacher educators a pedagogical and technical environment for improving 11–14 year-old students' learning of algebraic generalisation: the capacity of appreciating generality and structure in problems, and being able to solve them in general and not only for specific instances. Although expressing generality, recognising and analysing patterns and articulating structure are at the core of mathematical thinking and scientific enquiry, these ideas are notoriously elusive for students [1]–[3]. In MiGen, we are adopting a constructionist approach [4], allowing students to create and manipulate patterns and algebraic expressions and to perceive the relationships between them. The idea of 'seeing the general through the particular' has been shown to be a powerful way of introducing students to generalisation [1]. Following [5], MiGen is based on the premise that learners build knowledge structures particularly well in situations where they are engaged in constructing public entities — in our context, these are two-dimensional patterns and algebraic rules about these patterns that they will share with other students. The MiGen system acts as a mediator of social interaction between students, and also between the teacher and the students, through which shared algebraic expressions are constructed and observed. The system provides not only a means by which students construct their own problem space but also a communicative function through which algebraic interpretations are made explicit and can be discussed with others.

It has been argued that considerable guidance is required to ensure learning in open-ended contexts [6]. The exploratory nature of our learning environment requires that personalised feedback is provided to students by the system during their construction process. Also, since students are undertaking loosely-defined rather than structured tasks, teachers need to be assisted in monitoring their students' actions and progress by appropriate visualisation and notification tools [7].

The aim of these teacher assistance tools is to inform teachers of students' progress, the appearance of common misconceptions, students who may be disengaged from the task, and students who may be in difficulty. This allows teachers to support learners in a personalised way, assisting them to formulate their own interventions in encouraging students to reflect on their constructions, on the feedback the system is giving them, on working towards specific goals, and in communicating and sharing their constructions with others.

In this paper, we discuss the design of MiGen's teacher assistance tools, motivating and describing their architectural design, and giving a detailed description of one such tool, the Student Tracking tool. Although some context is given about the overall project, readers interested in the pedagogical underpinnings and how the dynamic potential of digital

- *Dr. Gutierrez-Santos and Prof. Poulovassilis are with London Knowledge Lab, Birkbeck, University of London; as was Dr. Pearce-Lazard at the time when this research was undertaken.*
- *Dr. Geraniou is with Institute of Education, University of London.*

The MiGen project is funded by the ESRC/EPSRC TEL programme (RES-139-25-0381). Additional background, publications, and releases of the software can be found at www.migen.org. *The source code is available at* code.google.com/p/migen *under a GPL licence.*
Contact E-mail: {sergut,ap}@dcs.bbk.ac.uk, e.geraniou@ioe.ac.uk.

technologies is used to enhance the students' understanding of algebraic generalisation are referred to [2], [8].

The paper is structured as follows. Section 2 gives an overview of the MiGen system, to the level of detail necessary for this paper, and introduces the teacher assistance tools. Section 3 describes the types of events that these tools are able to present visually to the teacher. Section 4 gives a detailed description of the Student Tracking tool. Section 5 discusses the iterative process through which this tool has been co-designed with our teacher collaborators on the MiGen project. Section 6 reviews related work and compares our own work with this. In Section 7 we present our conclusions and discuss directions of future work.

2 The MiGen System Context

The MiGen system is intended to be deployed within the classroom. During a lesson, students work on algebraic generalisation problems as selected by their teacher and presented to them by the system. While this is happening, the teacher may wish to view real-time representations of the students' activities and progress. At other times, teachers may also wish to view historical information about their students' progress as maintained by the system.

MiGen comprises a number of tools, which we describe here to the level of detail necessary for this paper. We refer the reader to the cited references for more details of some of these tools.

At the core of the system is the **eXpresser**, a mathematical microworld [9], [10] which supports students in undertaking algebraic generalisation tasks. As part of a possibly larger activity sequence (see the Activity Tool below), students are asked to construct 'generalised models out of patterns' (see below) using the eXpresser. Figure 1a and Figure 1b show two instances of an example model. Instances of a model such as these are presented to students and they are asked to construct the model in the eXpresser and to derive a general rule for the total number of tiles required for any instance of the model. For this task, they are prompted to create and use different patterns made out of different coloured tiles, depending on their perceptions of the model's structure. Every pattern can be made of a single square tile or group of tiles repeated a number of times. Tiles and groups of tiles can be repeated horizontally, vertically or diagonally with or without spaces in between its repeated components (for more details, the reader is referred to [2]). The eXpresser supports students not only in their construction of the model using patterns, but also in deriving algebraic expressions underpinning the model, for example, referring to Figures 1a and 1b, the number of green tiles (light grey) required given the number of red tiles (dark grey).

The eXpresser microworld gives a lot of freedom to students, who may construct their models in a variety of different ways for the same task. For example, if a student has constructed their model using a series of 'C' shapes as illustrated in Figure 1c, they may derive an expression such as $5r + 3$ for the number of green tiles, where r is the number of red tiles. A range of other possible constructions that students may follow are shown in the remaining diagrams of Figure 1 which also show that the form of the resulting expressions (all of which are equivalent to $5r+3$ of course) can vary widely. If r is a variable, all of these constructions are *general*, and changing the current value of r will lead to the student's current instance of the model changing accordingly too.

Figure 2 shows the eXpresser user interface. Students use building blocks that they make up from square unit tiles in order to construct their patterns, which they can subsequently colour. When constructing a pattern, they make use of numbers which they can subsequently 'unlock' to turn them into variables in order to generalise their pattern. Both locked and unlocked numbers can be used in expressions. The eXpresser has an 'animation' facility which allows the student to test out the generality of their model by automatically applying different values to unlocked numbers and displaying the resulting instances of the model.

Microworlds such as the eXpresser are designed to provide opportunities for learners to develop complex cognitive skills rather than just knowledge of concepts in the subject domain [9], [10]. The tasks that students are asked to undertake are usually open-ended in nature, have multiple possible solutions, and encourage students to explore the construction environment and follow a variety of construction strategies. Through such interactions, they make explicit the mathematical relationships between and within these objects and in this process identify the variant and invariant components in their constructions, express relationships between these components, form semi-algebraic expressions with the eXpresser language and consequently engage with mathematical concepts, such as variables, constants, the expression of relationships and algebraic generality. [11] describes MiGen's multi-layered learner model which, as well as modelling the attainment of concepts in the subject domain, also includes a 'layer' of knowledge comprising microworld-specific concepts that operationalise the concepts of the subject domain, as well as encompassing the affordances of the microworld itself. In MiGen, tasks are designed to contextualise students' interaction with the eXpresser, including specific learning objectives that the learner should achieve as they undertake a task e.g. 'find a general expression to colour your pattern' — see Figure 2(G).

As a student interacts with the eXpresser, so a series of *indicators* are automatically detected or inferred by the system, which can then be notified to the teacher via the Teacher Assistance tools (which we will dis-

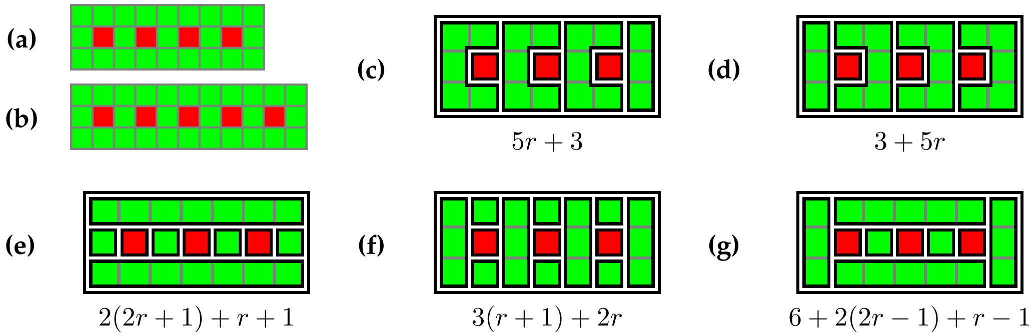


Fig. 1. **(a)–(b)** Instances of a “stepping stones” task model and **(c)–(g)** several possible general constructions where each expression specifies the number of green tiles in terms of r, the number of red tiles.

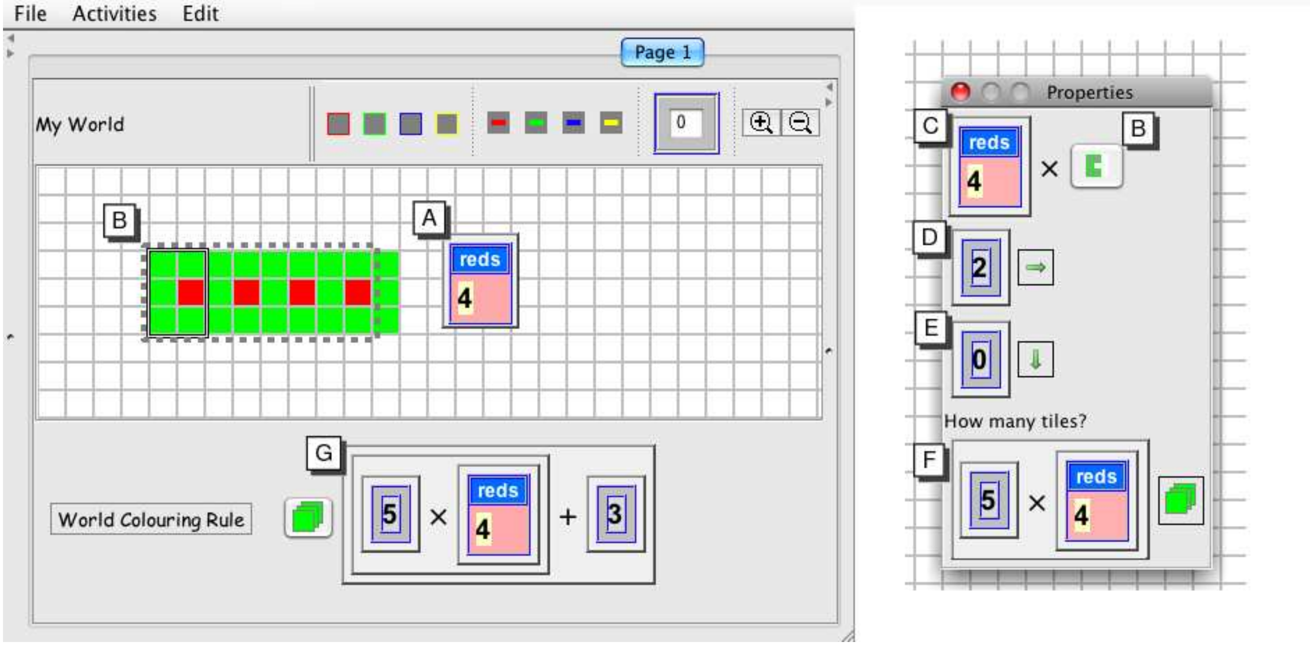


Fig. 2. Constructing a model in the eXpresser and describing it with a rule. Letters highlight the main features: (A) An ‘unlocked’ number that acts like a variable is given the name ‘reds’ and signifies the number of red (dark grey) tiles in the model. (B) Building block to be repeated to make a pattern. (C) Number of repetitions (in this case, the value of the variable ‘reds’). (D,E) Number of grid squares to translate B to the right and downwards after each repetition. (F) Units of colour required to paint the pattern. (G) General expression that gives the total number of units of colour required to paint the whole model.

cuss shortly). There are two categories of indicators. First, *task-independent* (TI) indicators refer to aspects of the student’s interaction that are microworld-related but do not require knowledge of the specific task the student is currently working on. They always refer to single actions undertaken by the student e.g. ‘student has placed a tile on the canvas’, ‘student has made a building block’, ‘student has unlocked a number’. In contrast, *task-dependent* (TD) indicators require access to knowledge about the specific task the student is working on. They can relate to individual actions or to the result of sequences of actions, and they require a level of intelligent reasoning to be applied by the system. In some cases, their detection may have a degree of uncertainty associated with it. Examples of TD indicators include: ‘student has made a plausible building block’ (i.e. a building block that can potentially lead to a valid solution), which requires knowledge of the set of possible solutions to a task; ‘student has unlocked too many numbers’, which requires knowledge about how many variables a task needs; ‘student has coloured their model generally’, which requires reasoning on the student’s expression. We discuss TI and TD indicators in more detail in Section 3. We also list the full set of indicators currently supported in the appendix.

The **Activity Tool** presents students with activity sequences targeting algebraic generalisation, as designed by the activity designer or the teacher. These activity sequences include phases such as introduction to an exploratory learning task, undertaking a task using the eXpresser, reflecting on their interaction, and sharing and discussing their constructions and rules with other students.

A **Task Design Tool** is currently under development. This will allow the designer or teacher to describe new algebraic generalisation tasks, and to input into the system a set of possible solutions for them, i.e. possible constructions and their associated algebraic expressions. Currently, the set of possible solutions for a task is input into the system by the research team using the eXpresser. Part of the description of a task is the set of goals that students need to achieve as they work on the task (e.g. 'make a building block', 'colour a pattern', 'write an expression for the number of tiles of each colour in your model'), and the set of learning objectives supported by the task (ranging from microworld-related knowledge such as 'knowledge on how to animate' or 'knowledge about names for numbers' to domain knowledge such as 'understanding the connection between the construction structure and its associated pre-algebraic expression' or 'understanding the use of animation to validate the generality of models'). The Task Design Tool will allow the designer/teacher to select these from the overall sets of goals and learning objectives supported by the system. Currently, such information describing tasks is entered directly into the MiGen database by the research team. The total sets of task goals, learning objectives and indicators detectable by the system are easily extensible with new ones as the system is developed and extended over time.

The **eGeneraliser** is a suite of intelligent components which take as their input information from the eXpresser as students undertake tasks, as well as information in the MiGen database relating to students (their learner model) and to tasks (the task description and set of possible solutions). These intelligent components analyse the interaction of the students with the eXpresser, generate real-time feedback for students (prompts to help students engage with a task, improve their solutions, and generalise their solutions), infer the occurrence of TD indicators, and update students' learner models during, and at the end of, each student's usage of the eXpresser tool. A hybrid of case-based and rule-based reasoning is used in the eGeneraliser in order to infer the occurrence of TD indicators and to update students' learner models. We discuss some of the techniques employed in Section 3.

Finally, the **Teacher Assistance Tools** is a suite of tools aiming to assist the teacher in monitoring students' interactions and progress and in intervening as she decides appropriate. An extensive requirements analysis for these tools has been undertaken since early 2010 with the teachers involved in the MiGen project, and this has driven the iterative specification and co-design of these tools.

The overall MiGen system has a client-server architecture, as discussed in [12], [13]. The client software is executed on each student's computer (without the Teacher Assistance and Task Design tools) and on the teacher's computer (the whole suite of tools), while the server software is executed on one server computer. As students interact with the Activity Tool and with the eXpresser, these tools post information about students' actions to the MiGen Server, including the occurrence of TI indicators. The MiGen server stores this data in the MiGen Database which is implemented in JavaDB.

The eGeneraliser monitors updates occurring in the eXpresser and the Activity Tool due to students' interactions. It uses this information to infer the occurrence of TD indicators and to decide if it is appropriate to generate feedback for this particular student at this time, and what that feedback should be. Both the inference of TD indicators and the production of personalised feedback are posted by the eGeneraliser to the MiGen Server. Feedback is presented to students via the eGeneraliser's User Interface, which comprises a set of graphical components designed to present feedback to the student without interrupting the flow of their thoughts or compromising their exploration [14]. The eGeneraliser may also generate updates to the student's learner model, which are also posted to the MiGen Server.

The Teacher Assistance tools derive their information from the MiGen Database. This information includes: log data relating to students' activities and constructions, as posted to the MiGen Server by the Activity tool and the eXpresser; TI indicators detected by the Activity tool and the eXpresser as students interact with these; TD indicators inferred by the eGeneraliser; data posted to the MiGen Server by the eGeneraliser relating to updates that it has made to a student's learner model and feedback that it has generated for a student; the students' learner models; the task descriptions and possible solutions; and students' own constructions during, and at the end of, an activity. The Teacher Assistance tools can subscribe to the MiGen Server to be notified of the occurrence of updates to this information. They may also generate updates of their own, which they can post to the MiGen server in the same way as the other tools.

This communication between the student-facing tools, the MiGen Server and the Teacher Assistance tools provides a general architectural template for supporting all of the information requirements of MiGen's Teacher Assistance tools. In Section 4 we discuss in detail the design of one of the Teacher Assistance tools — the Student Tracking (ST) tool. We first describe in the next section the types of events that the ST tool is able to show visually to the teacher.

3 Interaction Indicators

The main goal of the Teacher Assistance tools is to enable teachers in the classroom to visualise information about their students' constructions. Given the exploratory nature of the tasks that students undertake with the eXpresser, it would be difficult for teachers to obtain and retain this information in their minds without the help of appropriate tools. There is generally a wide variety of approaches that students can take to construct their models using the eXpresser. Some of these will be valid approaches leading towards achievement of the task goals while others will not allow the student to fully achieve the task goals. Just inspecting a student's screen as they work on an eXpresser task may not be sufficient for the teacher to understand the model construction strategy that the student is following. This problem becomes even more acute when considering that teachers normally need to support classes of 25-35 students at the same time.

Teachers need to be able to help students working with the eXpresser in the classroom, and they also need to have a clear picture of the degree of achievement of the lesson's learning objectives by the whole class. This is a challenging task in any learning environment, but more so in the case of open and unstructured learning activities. The ST tool has been created with the aim of enabling teachers to track what their students are doing during the lesson, and also to examine this information after the class session in order to determine the degree to which students have achieved the lesson's learning objectives, so that the teacher can plan appropriately for the next lesson.

In order to achieve these aims, we began by identifying the set of indicators that are meaningful for teachers to be informed of while, and after, students work on eXpresser tasks. These *interaction indicators* serve as an abstract representation of the interaction between the student and the system. An appropriate visualisation of these indicators allows the teacher to be informed of important aspects of the student's construction, including the evolution of students' strategies during the lesson, possible learning trajectories followed by students, the feedback they received from the system, and how this influenced their subsequent actions.

The identification of the set of indicators was achieved through an iterative process undertaken over several months as a joint activity with our teacher collaborators on the project. We discuss next the current set of indicators supported and their detection. The evaluation of the indicators and the Student Tracking tool with teachers is discussed in Section 5.

3.1 Types of indicators

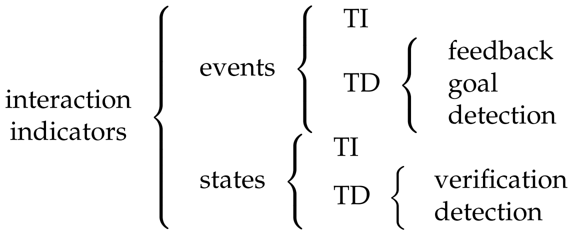


Fig. 3. Taxonomy of interaction indicators. TI = task-independent, TD = task-dependent.

Interaction indicators are automatically inferred by the system as a student uses the eXpresser. Figure 3 shows a taxonomy of the different types of indicators. Indicators that refer to a discrete action, or to a discrete inference from a combination of actions, at a specific point in time are called *events*. There are three types of TD events: Feedback-related events are generated when feedback is produced for students by the eGeneraliser or is explicitly requested by them, through the eXpresser. Goal-related events are produced when there is a change relating to the goals of the current task, e.g. the student considers they have achieved a goal, or the system detects that a goal has been achieved even if the student has not realised it yet. Finally, a third type of TD event involves the detection of some particular feature in the interactions of the student, e.g. that their actions follow a rhythmical pattern.

On the other hand, *states* are indicators relating to some aspect of the student's interactions that can be observed continuously as they work on a task. Some of the state indicators refer to instantaneous information (e.g. whether the construction is being animated at the moment) while others refer to historical information (e.g. whether the student has been able to construct a general solution at some point during their interaction). There are two types of TD state indicators: Some are related to the *verification* of a boolean condition, e.g. whether a plausible building block is being used by the student, or whether their colour allocation is correct. The others involve the *detection* of some particular feature in the actions of the student, e.g. 'clutter' being detected on the canvas by the system (those tiles that are not being used in the student's model, and may be distracting the student). Tables 3 to 7 in the Appendix list these different subsets of interaction indicators.

Over the course of this research and our elicitation of meaningful and useful indicators with teachers, the number of indicators has grown considerably. At some point, it became evident that it would be infeasible for the teacher (especially teachers who are not familiar with the eXpresser) to comprehend all of the information that the indicators can potentially give during the course of the lesson (see the discussion in Section 5.2 below). Larger combinations of indicators would be likely to be useful for after-class analysis, but the number of indicators to be displayed during the classroom session needed to be reduced. In Tables 3 to 7, the indicators which have been selected to be displayed within the Student Tracking tool by default are marked with the symbol † next to their descriptions. We stress however that teachers can choose to 'switch on' more indicators to be displayed if they wish, or to 'swich off' any of this initial default set.

3.2 Detection of indicators

Some of the indicators, especially the Task Independent ones, are detected easily by monitoring the appropriate events in the eXpresser. Indicators relating to the creation or use of expressions for allocating colours to patterns are examples of this kind. Other indicators require additional intelligent processing. This section describes two such processes of indicator detection, by way of illustration, one for a TD event indicator, followed by one for a TD state indicator. A description of all the variety of computational intelligence techniques employed in the eGeneraliser lies beyond the scope of this paper and we refer readers to [15]–[17] for further details.

3.2.1 Rhythm detection

In the early stages of the development of their thinking about generalisation, students may find it difficult to perceive the structure of models in their minds. One strategy that teachers can exploit is observing the way in which students describe patterns, or how they construct them using tiles. Sometimes students will have some implicit structure in their minds and this will be evident in their actions, even if they cannot make it explicit. The teacher can at this point encourage students to focus on their implicit structure, thus providing scaffolding towards students making their structure explicit and understanding how it relates to the generality of a pattern [18].

To support the teacher in this aim, one of the modules of the eGeneraliser is responsible for detecting 'rhythm' in the way that students place tiles onto the canvas of the eXpresser. When a repetitive sequence of tile placements is detected in the actions of the student, this can be used by the system in order to provide feedback to the student suggesting the creation of such a building block in order to construct their model in a more structured way (repeating the building block to create their model, rather than using single tiles at a time). At the same time, an instance of the corresponding indicator is also inferred by the eGeneraliser (this is the seventh indicator in Table 5).

In order to detect these regularities in students' actions, the tile placements made by a student are converted internally by the system into a sequence of positions on the canvas. The sequence of tile positions is analysed using two sliding windows containing sub-sequences from the whole sequence. The sliding windows traverse the sequence and, for each pair of positions, the distance between the windows is calculated using a string similarity metric. The use of a similarity metric, rather than a precise comparison, allows for small differences in consecutive repetitions of the same structure (for example, to allow for students' 'slips' in using the eXpresser). Those windows that exhibit a higher number of repetitions of high similarity to other subsequent windows are selected as possible indications of rhythm in the actions of the student. We refer the interested reader to [17] for more details of the process.

3.2.2 Apparent solution on canvas

One important challenge when providing feedback to students is understanding whether the student's current construction is a valid solution. The answer to this question has ramifications for the system's entire strategy for providing support and many other considerations depend on this one: Is the student's construction general or not? How does it relate to the student's final expression? Are the local expressions correct (i.e. the rules about how to colour individual patterns within the overall model)? Have they been combined correctly to obtain to the global expression (i.e. the final expression for colouring the whole model)?

Comparing two constructions made up from square tiles is relatively easy, but there are several difficulties in our case. The first difficulty stems from the fact that the construction of patterns using the eXpresser is highly exploratory. Given a task model, it can generally be constructed in large numbers of different ways e.g. using large or small patterns, with or without overlaps between patterns, on different parts of the canvas, etc. The second difficulty arises from the dynamicity of task solutions. Students are expected to make constructions that 'animate' i.e. that generalise correctly for any values of the task variables; but our studies have shown that many students are content to make just one instance of the model and they expect the system to do all the rest of the work for them. Detecting that they have 'finished' their construction has two aspects therefore: first, the system needs to detect that they have created a correct solution; and second, the system needs to evaluate the generality of the solution.

Regarding the first of these aspects, a module of the eGeneraliser is responsible for detecting constructions that have the same appearance as a valid solution. In this context, having "the same appearance" means looking the same from the point of view of the student, regardless of internal structure or actual equality tile-by-tile. For example, students will perceive a "stepping stones" model with 5 red tiles as looking the "same" as a model with 4 red tiles (see Figure 1a-b). Our algorithm is based on constructing a 'mask' based on each of the known solutions to the task (we recall that known solutions are identified using the Task Design Tool). This mask is projected onto the student's construction to see if there is a match for some value of the task variables. If a match is found, then the indicator Apparent Solution on Canvas is turned "on"; otherwise, it is turned as "off" (this is the first indicator in Table 8).

There is also another intelligent module that is responsible for evaluating whether the construction built by the student is general, meaning that it is a correct solution for any value of the task variables and not only for some i.e. that it cannot be "messed-up" by changing the variables (c.f. the concept of "messing-up" in dynamic geometry learning environments [19]). The algorithm used is again based on superposition of masks onto the student's construction for different values of the task variables. The number of values that need to be tested depends on the type of the task e.g. for a linear task that has just one task variable, only two values need to be tested. If a match is found for all the sampled values, the student's construction qualifies as "unmessable". If the student checks the generality of a model that is "unmessable" by using the animation features of the eXpresser to explore several possible values for the task variables, the indicator Unmessable Pattern Animated is turned "on"; otherwise, it remains "off" (this is the fourth indicator in Table 8).

4 The Student Tracking Tool

The Student Tracking (ST) tool provides teachers with information about the occurrence of TI and TD indicators as students are working on a task using the eXpresser in the classroom. This information can also be viewed by the teacher for after-class analysis of what students have achieved during the lesson, so as to help in planning the next lesson. One of the principles in the design of MiGen's teacher tools has been to make them as unobtrusive as possible with respect to current teacher practice in the UK classroom. The tools collect information about students passively in the background [12] and show it to teachers through their computer in the classroom or a mobile terminal (e.g. a tablet). Teachers can therefore continue their usual routine in the classroom and increase gradually their use of the teacher tools as they become more confident with this new source of information.

Figure 4 shows a screenshot of the current ST tool[1]. This view shows the sequential occurrence of indicators for a group of students during their interaction with the eXpresser. Each column represents the progress of the student named at the top of the column. A coloured horizontal cell indicates that a particular event indicator has occurred for a particular student. Each event indicator occurrence is positioned according to the time at which it occurred (with time increasing downwards). The colour of the cell indicates whether the indicator is regarded as being 'positive', 'negative' or 'neutral', as described in Table 1. By hovering with the cursor over an indicator the teacher can obtain additional information (e.g. a complete description of the indicator, the time at which it occurred, etc).

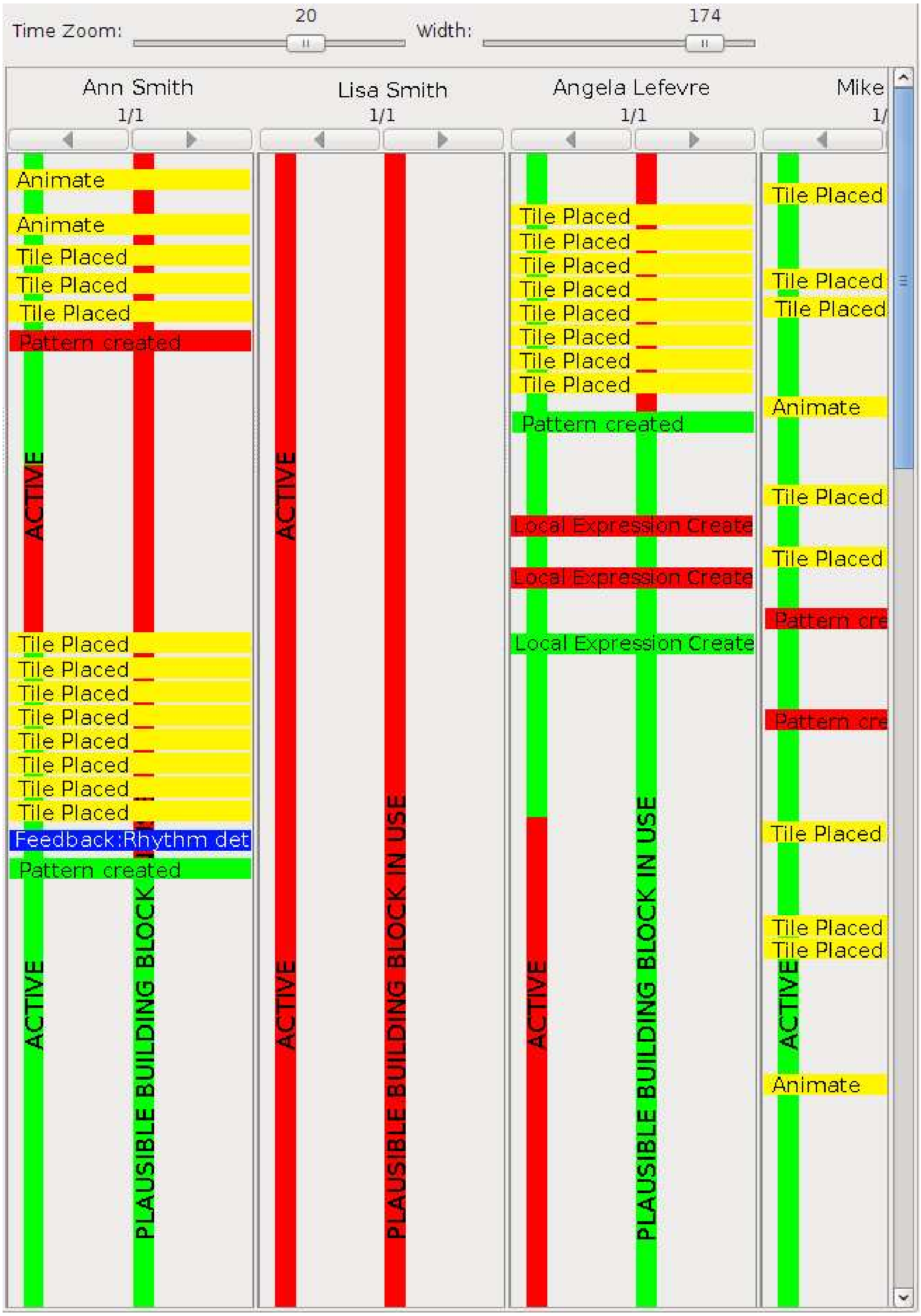
State indicators are represented by vertical lines that start at the top of the display and continue downwards until the end of the student's interaction with the eXpresser. The vertical lines representing states use the same colour encoding as for event indicators, and change colour accordingly e.g. when a student is detected as being inactive, the "Active" line will change colour from green (light grey) to red (dark grey) at that point in time. For the sake of clarity, the screenshot has been partially edited to show illustrative examples of interaction with the MiGen system within the space of one page; in particular, in reality students will need to be inactive for a longer period of time than shown for Ann and Angela before their "Active" line turns from green to red.

From Figure 4, it is immediately apparent that several students have been inactive at some point during their interaction with the system. However, many of them have achieved other indicators.

Taking Ann Smith as an example, this representation shows that, initially, Ann seemed to be disengaged from the task: she clicked on the 'animate' button a couple of times despite her canvas being empty. She then proceeded to place a handful of tiles. At this stage, the system detected that she was inactive for a while. If the teacher had viewed this representation at that point, she could have intervened and encouraged Ann to try placing single tiles to construct a pattern. She would then be able to see that Ann subsequently followed her advice and the system detected rhythm in Ann's placement of the tiles. Futhermore, the representation shows that the system detected rhythm in the way Ann placed the tiles in the canvas, and provided some feedback related to this fact. Apparently, the feedback was helpful because Ann immediately created a pattern using a plausible building block (as evidenced by the positive change in the state indicator "Plausible Building Block"). Lisa Smith, on the other hand, has just been inactive, as demonstrated by her long red "Active" state-line. Arrangements of indicator occurrences such as these can assist the teacher in making decisions about which students need help the most.

In contrast to Ann and Lisa, we see that Angela Lefevre has made good progress with the task: she has placed tiles, made a plausible building block, and built a pattern using this block. She initially had problems specifying a correct local expression (i.e. an expression for the number of tiles of a particular

[1] All students' names (both screenshot and text) have been changed.



Time Zoom: 20

Width: 174

Fig. 4. A portion of the ST Tool User Interface showing when events (i.e. horizontal bars) occur and states (i.e. vertical bars) change for each student. The selection of indicators is a subset of the total range of indicators available, as explained in Section 3. The meaning of the colours is explained in Table 1.

Colour	Type	Description
Green (mid-grey)	Positive	The occurrence of such an indicator indicates that the actions of the student are in line with what would be expected from constructive interaction with the system.
Red (dark-grey)	Negative	The occurrence of such an indicator is regarded as an obstruction to constructive use of the system. For example, the student being inactive.
Yellow (light-grey)	Neutral	The occurrence of such an indicator indicates some aspect of the student's interaction that may be positive or negative depending on the context. For example, the student has changed the value of an unlocked number.

TABLE 1
Types and colours of interaction indicators as represented in the ST tool.

colour in a specific pattern), as shown by the two red instances of the "Local Expression Created" indicator. However, at the third attempt, she has managed to find the right expression. However, she has not yet animated her construction to test out its generality, or done anything else from that point on. Maybe she thinks she has already completed all steps she was expected to, and is therefore waiting until the next activity is explained to the class. The teacher can use this information to remind Angela to try animating her construction. Angela would then be able to see whether she has built it generally. This demonstrates therefore that not only can the ST tool show the teacher who needs help the most but can also provide more subtle information which can be used to ask students thought-provoking questions or further facilitate them on their way to successful completion of the task.

5 Evaluation

The design and evaluation of the ST tool has involved an iterative process in which our teacher collaborators have played a central role. Because the number of teachers we could practically collaborate with during the course of the project was small, and their time available to experiment with early prototypes in the classroom was limited due to the need for them to deliver a tightly timetabled curriculum, interviews with a small group of teachers played a prominent role, especially in the early stages of our research.

5.1 First trial

The first version of the ST tool to be used in a classroom context was trialled with one of our teacher collaborators in July 2010. That teacher had been involved in the design of the tool from its inception and therefore had a good understanding of its functionalities. At that time, there was no separation between 'event' and 'state' indicators, and all indicators were shown as horizontal cells (a screenshot of this version appears in [13]).

The teacher used the tool to observe the actions of a class of around twenty 13-year-old students. One major item of feedback that we received in the post-lesson interview with this teacher was that some of the indicators were actually showing changes in *state* rather than being single events — e.g. whether the students were active or inactive, whether all shapes they were currently using were patterns, etc. The teacher suggested that this family of indicators would be better visualised as vertical bars that change colour when their status changes. It also became evident that feedback-related events should also be visualised within the ST tool. Up to that point, we had focused on presenting the *student's* actions, but during the post-lesson interview it became apparent that teachers would also find it helpful to know what feedback is being generated for students by the *system* as they are using the eXpresser, because this has an impact on students' subsequent actions. Since feedback-related indicators are neither positive nor negative, and moreover they need to be distinguished from the other 'neutral' indicators, they are displayed in a different colour (blue) in the current ST tool. We co-designed with this teacher these additional functionalities, and the resulting user interface is illustrated in Figure 4.

5.2 Second trial

A second classroom trial of the eXpresser and ST tools was carried out in November–December 2010, involving the same teacher as earlier (Teacher 1, in School 1) using this new version of the ST tool, and also with another of our teacher collaborators in a different school (Teacher 2, in School 2). Prior to the lessons, in both schools, a meeting of the research team with the teacher took place with the aim of discussing the functionalities of the tools and familiarising the teachers with them. Teacher 1 was already very familiar with both tools, having been intimately involved in their design and having participated in the first trial. Teacher 2 had not seen the tools before.

Teacher 1's class comprised approximately 20 students aged 12-13 in the average to higher attainment spectrum, from a suburban population. Teacher 2's class comprised approximately 25 students aged 12-13 of mixed attainment levels from central London. In both lessons the students worked individually using the eXpresser on their computers. During both lessons, the teachers spent most of their time going around the class helping students in using the eXpresser and working on the task that had been set, and they afforded little time to consult the ST tool that was running on their own computer (i.e. the teacher's computer, situated at the front of the class). This behaviour seems very much linked to the current mathematics classroom culture in England, where teachers only stay by their desk in the early part of a lesson, and they walk around the classroom helping students for the remainder of the lesson, answering questions and giving feedback. It seems that teachers are not used to having access to tools that can provide them with real-time information about their students' actions and progress and they revert to their usual teaching 'habits' when confronted with the pressure of having to help their students in the class.

In post-lesson interviews, the teachers suggested two possible ways of alleviating this situation. One teacher suggested to install the Teacher tools on tablet PCs, which would then allow teachers to view these tools as they are walking around the classroom. The other teacher suggested to project the tools' display onto the whiteboard at the front of the class, again allowing teachers to monitor the progress of the class as they are walking around the classroom. This latter proposal needs careful consideration however, as there are ethical issues to consider: even though it could prove to be quite motivating for some students, there is a possibility of students feeling demoralised or demotivated as they observe their own progress compared to others.

A major item of feedback received from both teachers was that the information shown by the ST tool was too detailed to be useful to them during the lesson. However, they both felt it would be useful to be able to track this level of detail for individual students after the end of the lesson, in order to provide more individualised support to the students in the next lesson. As a result of this feedback, the indicators now displayed by default in the ST tool have been reduced to the subset marked with the symbol † next to their descriptions in the Appendix. Teachers can choose to 'switch on' more indicators to be displayed if they wish, or to 'swich off' any of this initial default set by means of an indicator-selection feature. The teachers also mentioned that they would like to see a higher level of information displayed than with the current ST tool, focussing on students' achievement of task goals.

Another major item of feedback was that the ST tool would benefit from the ability to:

(i) define and view 'higher-level', derived indicators from the current set of 'low-level' indicators, for example occurrences of sequences of indicators;

(ii) see how many times a particular indicator has occurred, for example the level of achievement of each task goal over the whole class, so that the teacher can reinforce in the next lesson those aspects of the task that students found difficult.

The request for capability (i) has led to ongoing research effort to identify which indicators are useful for teachers in which usage scenarios. Capability (ii) was in fact provided in an early prototype of the ST tool (a screenshot of which appears in [13]) but at that time its development was de-prioritised on the advice of our teacher collaborators in favour of the more detailed view presented in Figure 4. This sometimes conflicting advice received from the teachers who have been working with us serves to underline the novelty of the tools that we are aiming to provide, the teachers' lack of familiarity with such tools (and hence their difficulty in evaluating their own needs in the classroom), and the necessity of an iterative co-design process.

In summary, we have learnt from the evaluation of the ST tool that it is possible to identify a set of interaction indicators showing students' progress as they are undertaking their construction tasks that can be displayed to teachers in a way that facilitates their work in the classroom. We have identified a reduced set of interaction indicators that teachers find particularly useful in the classroom, and an extended set of indicators that can be visualised on demand, e.g. for after-class analysis. The main directions of future work to enhance the ST tool lie in the identification and detection of derived indicators, and in the provision of summary information showing how many times a particular indicator has occurred during a particular student's construction and for the class as a whole.

6 Related Work

To our knowledge, ours is the first work targeted at notifying the teacher about students' attainment of indicators during constructionist learning tasks. There has of course been much work in developing tools that assist teachers' instructionist role, or that aim to help teachers to structure the use of exploratory tasks in the classroom [20]. However, to our knowledge, there are no other tools that assist teachers in tracking their students' progress during constructionist learning activities. This novelty has had several implications for our work, ranging from computational aspects such as developing appropriate techniques for inferring each of the TD indicators, to the iterative co-design process with our teacher collaborators that we have reported on above.

The work closest to ours is that of [21] which uses Web log data generated by course management systems (WebCT in their case) to help instructors be aware of students' activities in distance learning classes; [22] which presents tools for helping teachers understand students' behaviour in adaptive tutorials through post-analysis of the system's data logs; [23]

which presents tools for teachers to visualise students' progress through simulation-based practical work sessions, and [24] which provides awareness information to teachers so as to support their role as moderators of multiple e-discussions.

Mazza and Dimitrova [21] use techniques from information visualisation to represent multidimensional student tracking data; and our Student Tracking tool has a similar representation to some of their visualisations. However, they do not focus on detecting and visualising the occurrence of indicators in open-ended constructionist learning tasks but rather on visualising students' social and behavioural aspects, their progress with respect to the course schedule, and their performance on quizzes and assignments and the level of knowledge achieved for each domain concept of the course. Another important difference is that their approach is at a higher level of granularity (i.e. information about a whole course over several days or weeks), while we focus on helping teachers supporting exploratory activities both *in* the classroom and *after* the class session, and therefore our level of granularity needs to be at the same time finer and broader than theirs. We have therefore had to undertake an iterative process to elicit from teachers which indicators were appropriate for different usages, as reported in Section 4. We refer the reader to [21] for an extensive review of other work in visualisation of data collected by course management systems, some of which also uses data mining and intelligent techniques to analyse student data and generate feedback to users — though, again, none of this earlier work focuses on constructionist learning.

Ben-Anim *et al.* [22] adopt a hybrid approach whereby part of the data mining effort is teacher-driven and part is automated, but like much other work in educational data mining it does not focus on monitoring students' ongoing progress through constructionist learning tasks. Their map-based approach is not feasible for tasks where there is a high degree of freedom and uncertainty in the students' interactions (as is the case with the eXpresser), as the number of states needed in the map would quickly become unmanageable.

Gueraud *et al.* [23] compose a practical work session from a sequence of simulation-based problems. Their focus on simulation-based learning in classrooms has some similarities with our work with microworlds: their simulators of electric circuits would be analogous to our eXpresser tasks, although the level of exploration granted to students is more limited. In their approach, there are explicit conditions on simulation states that evaluate to correct/incorrect and there are explicit requests by learners for validation of conditions, neither of which is the case with our more open-ended constructionist learning tasks in MiGen. Additionally, in our case the set of indicators were identified with our teacher collaborators through an iterative process. Different indicator visualisations are needed in different situations, and also different teachers may wish to select different subsets of indicators at different times, and hence our Student Tracking tool includes such a customisation facility to allow this.

The SCY (Science Created by You) project aims to develop an open-ended learning environment for the learning of science. In that respect, the goals and aims of SCY and MiGen have much in common despite the different knowledge domains. The work by Wichmann et al. [24] has some similarities with the work presented here as it also aims at increasing the awareness of teacher about what their students do. The difference is in focus: while we concentrate in reporting information about what students do in the context of individual constructionist taks, they target discussion moderation.

7 Conclusions

In this paper we have described the design of the Teacher Assistance tools of the MiGen system, an intelligent exploratory enviroment aiming to support 11–14 year-old students in their learning of algebraic generalisation. MiGen's Teacher Assistance tools aim to provide information to teachers about their students' activities and progress as they use the eXpresser and Activity tools, assisting them in the detection of students who may be experiencing difficulties, alerting them to students who seem to be demonstrating misconceptions and to those who may benefit from further, more challenging tasks. The tools aim to allow the teacher to facilitate the students' productive interaction with the system by increasing their awareness of the classroom situation.

We have given in this paper a detailed description of one of the MiGen teacher assistance tools, the Student Tracking tool. Several MiGen system components are used to infer the occurrence of task-independent and task-dependent indicators of relevance to the teacher as students undertake tasks using the eXpresser microworld, and this information can be presented to the teacher through the Student Tracking tool both during and after the lesson.

To our knowledge, ours is the first work targeted at visualising students' progress through constructionist learning tasks and at notifying teachers of students' attainment of specific landmarks as they are undertaking their constructions. We believe that there is potential for more general application of our design approach and implementation techniques to other exploratory learning environments. Our indicator-based approach is not bound to a specific domain or to a specific combination of exploratory environment and constructionist task. Given an appropriate set of indicators, our tools could be used in other domains and learning environments (e.g. virtual chemistry labs, medical simulators). Some of our indicators are domain-dependent and some are not. Eliciting the right selection of indicators is a challenging endeavour, and we plan to establish a set of best practices based on our experiences of working with teachers in MiGen.

The Student Tracking tool has shown that our middleware architecture for the detection, retrieval and visualisation of indicators is viable in the context of the classroom after being used in two different schools, in three different physical classrooms, with different classes of students (around 100 students in total). We are now in the position of being able to rapidly design and develop additional tools to assist the teacher, based on the lessons reported in Section 5.

ACKNOWLEDGMENT

We thank the other members of the MiGen team for our ongoing stimulating collaborative research on the project, and especially the team of teacher collaborators for their insightful feedback.

REFERENCES

[1] Mason, J., Graham, A., Johnston-Wilder, S.: Developing Thinking in Algebra. Paul Chapman Publishing (2005)

[2] Noss, R., Hoyles, C., Mavrikis, M., Geraniou, E., Gutierrez-Santos, S., Pearce, D.: Broadening the sense of 'dynamic': a microworld to support students' mathematical generalisation. The International Journal on Mathematics Education (ZDM) **41**(4) (2009) 493–503

[3] Kuchemann, D.: Using patterns generically to see structure. Pedagogies **5** (2010)

[4] Harel, I., Papert, S., eds.: Constructionism. Ablex Publishing Corporation, Norwood, NJ (1991)

[5] Noss, R., Hoyles, C.: Windows on Mathematical Meanings. Kluwer Academic Publishers, The Netherlands (1996)

[6] Kirscher, P., Sweller, J., Clark, R.E.: Why minimal guidance during instruction does not work: An analysis of the failure of constructivist, discovery, problem-based, experiental and inquiry-based learning. Educational Psychologist **41**(2) (2006) 75–86

[7] Mavrikis, M., Geraniou, E., Noss, R., Hoyles, C.: Revisiting pedagogic strategies for supporting students' learning in mathematical microworlds. In: Proceedings of the International Workshop on Intelligent Support for Exploratory Environments at EC-TEL'08. (2008)

[8] Noss, R., Poulovassilis, A., Geraniou, E., Gutierrez-Santos, S., Hoyles, C., Kahn, K., Magoulas, G., Mavrikis, M.: The design of a system to support exploratory learning of algebraic generalisation. Computers and Education ((in press))

[9] Thompson, P.W.: Mathematical microworlds and intelligent computer-assisted instruction. In: Artificial intelligence and instruction: Applications and methods, Boston, MA, Addison-Wesley Longman Publishing Co., Inc. (1987) 83–109

[10] Hoyles, C.: Microworlds/schoolworlds: The transformation of an innovation. In Keitel, C., Ruthven, K., eds.: Learning from computers: mathematics education and technology. Springer-Verlag, Berlin (1993) 1–17

[11] Mavrikis, M., Gutierrez-Santos, S., Pearce-Lazard, D., Poulovassilis, A., Magoulas, G.: Learner modelling in microworlds: conceptual model and architecture in MiGen. Technical Report BBKCS-10-04, Birkbeck College, University of London (April 2010) Available at http://www.dcs.bbk.ac.uk//research/techreps/2010/.

[12] Pearce, D., Poulovassilis, A.: The conceptual and architectural design of a system supporting exploratory learning of mathematics generalisation. In: In Proceedings of the 4th European Conference on Technology Enhanced Learning (EC-TEL), Nice, France (2009) 22–36

[13] Pearce-Lazard, D., Poulovassilis, A., Geraniou, E.: The design of teacher assistance tools in an exploratory learning environment for mathematics generalisation. In: In Proceedings of the 5th European Conference on Technology Enhanced Learning (EC-TEL), Barcelona, Spain (2010) 260–275

[14] Mavrikis, M., Gutierrez-Santos, S., Geraniou, E.: Facilitating effective exploratory interaction: Design and evaluation of intelligent support in migen. In: Proceedings of the European Conference on Technology-Enhanced Learning (EC-TEL). (2010) 548–553

[15] Cocea, M., Gutierrez-Santos, S., Magoulas, G.: Challenges for intelligent support in exploratory learning: the case of ShapeBuilder. In: Proceedings of the International Workshop on Intelligent Support for Exploratory Environments at EC-TEL'08. (2008)

[16] Gutierrez-Santos, S., Cocea, M., Magoulas, G.: A case-based reasoning approach to provide adaptive feedback in microworlds. In: Proceedings of Intelligent Tutoring Systems (ITS 2010). (2010) 330–333

[17] Gutierrez-Santos, S., Mavrikis, M., Magoulas, G.D.: Sequence detection for adaptive feedback generation in an exploratory environment for mathematical generalisation. In: Proceedings of the Conference on Artificial Intelligence: Methodology, Systems, Applications (AIMSA). (2010) 181–190

[18] Radford, L., Bardini, C., Sabena, C.: Rhythm and the grasping of the general. In Novotná, J., Moraová, H., Krátká, M., Stehlíková, N., eds.: Proceedings of the 30th Conference of the International Group for the Psychology of Mathematics Education. Volume 4. (2006) 393–400

[19] Healy, L., Hoelzl, R., Hoyles, C., Noss, R.: Messing up. Micromath **10** (1994)

[20] Mulholland, P., Collins, T., Gaved, M., Wright, M., Sharples, M., Greenhalgh, C., Kerawalla, C., Scanlon, E., Littleton, K.: Activity guide: An approach to scripting inquiry learning. In: Int. Workshop on Intelligent Support for Exploratory Environments, in Artificial Intelligence in Education (AIEd'09). (2009)

[21] Mazza, R., Dimitrova, V.: CourseVis: A graphical student monitoring tool for supporting instructors in web-based distance courses. International Journal of Man-Machine Studies **65**(2) (2007) 125–139

[22] Ben-Naim, D., Marcus, N., Bain, M.: Visualization and analysis of student interactions in an exploratory learning environment. In: Proceedings of the 1st International Workshop on Intelligent Support for Exploratory Environments (part of ECTEL 2008). (2008)

[23] Gueraud, V., Adam, J.M., Lejeune, A., Dubois, M., Mandran, N.: Teachers need support too: Formid-observer, a flexible environment for supervising simulation-based learning situations. In: Proceedings of the 2nd International Workshop on Intelligent Support for Exploratory Environments (part of AIED 2009). (2009) 19–28

[24] Wichmann, A., Giemza, A., Krauß, M., Hoppe, H.U.: Effects of awareness support on moderating multiple parallel e-discussions. In: Proceedings of the 8th International Conference on Computer Supported Collaborative Learning (CSCL 2009), Rhodes, Greece (June 2009) 646–650

APPENDIX: INDICATOR TABLES

Indicator	Status
Feedback is available for the student	N/A
Feedback has been shown to the student†	N/A
Student has requested help from the system†	N/A

TABLE 2
Task-dependent feedback-related event indicators

Indicator	Status
Student created a building block	Positive
Student made a pattern	Positive
Tile placed on canvas	Neutral
Number changed value	Neutral
Global colour allocation changed	Neutral
Number created	Positive
Number unlocked†	Positive
Student gives a name to a number	Positive
Shape deleted	Neutral
All shapes deleted	Neutral
The student marks the task as "done"	Positive
Local colour allocation set correctly†	Positive
Local colour allocation set incorrectly†	Negative
Global colour allocation set correctly†	Positive
Global colour allocation set incorrectly†	Negative

TABLE 3
Task-independent event indicators

Indicator	Status
Goal n checked by student	Positive
Goal n un-checked by student	Negative
Goal n detected as accomplished by the system†	Positive
Goal n detected as not accomplished by the system	Negative

TABLE 4
Task-dependent goal-related event indicators

Indicator	Status
Plausible building block created†	Positive
Implausible building block created†	Negative
Pattern created using a plausible building block created†	Positive
Pattern created using an implausible building block created†	Negative
Model animated, does not *mess up*†	Positive
Model animated, it *messes up*†	Negative
Rhythm detected in the actions of the student	Positive
Correct answer given to a question on the Task Tool	Positive
Incorrect answer given (scaling error)	Negative
Incorrect answer given (unknown type of mistake)	Negative

TABLE 5
Task-dependent detection-based event indicators

Indicator	Status
All local allocations are correct	Positive
Model is animating	Positive
Student is inactive	Negative
Model is built using only tiles, without patterns or structure	Negative
Every shape on the canvas is a pattern	Positive
There are no uncorrected overlaps on the canvas	Positive
All negative tiles on the canvas have been used to correct overlaps	Positive
All unlocked numbers have a name	Positive
At least one of the unlocked numbers has a name	Positive
At least one of the locked numbers has a name	Positive

TABLE 6
Task-independent state indicators

Indicator	Status
Spurious tiles (i.e. clutter) detected on the canvas	Neutral
Task model built in a general way (i.e. structure cannot be messed-up)	Positive
Pattern coloured in a general way	Positive
Detected shape with an incorrect colour allocation	Negative

TABLE 7
Task-dependent detection-based state indicators

Indicator	Status
A pattern that looks like the task model is on the canvas (i.e. apparent solution on canvas)	Positive
Task model built using only single tiles (i.e. no patterns)	Neutral
Plausible building block in use	Positive
Student has made an unmessable model and animated it (i.e. unmessable model animated)	Positive
Student has made a solution that can be messed-up and animated it	Neutral
Appropriate rule given for the general colour allocation	Positive
The student is using the right amount of unlocked numbers for this task	Positive

TABLE 8
Task-dependent verification-based state indicators