

Design Requirements, Student Perception Indicators and Validation Metrics for Intelligent Exploratory Learning Environments

Manolis Mavrikis · Sergio Gutierrez-Santos ·
Eirini Geraniou · Richard Noss

the date of receipt and acceptance should be inserted later

Abstract The new forms of interaction afforded by innovative technology and open-ended environments provide promising opportunities for exploratory learning. Exploratory environments, however, require appropriate support to lead to meaningful learning outcomes. This paper focuses on the design and validation of intelligent exploratory environments. The goal is twofold: requirements that guide the operationalisation of pedagogical strategies to computer-based support and methodology for the validation of the system. As designers we need to understand what kind of interaction is conducive to learning and aligned with the theoretical principles behind exploratory learning. We summarise this in the form of three requirements — rare interruption of interaction, co-location of feedback, and support towards specific goals. Additionally, developing intelligent systems requires many resources and a long time to build. To facilitate their evaluation, we define three indicators — helpfulness, repetitiveness and comprehension — of students' perception of the intelligent system and three metrics — relevance, coverage, and scope — that allow the identification of design or implementation problems at various phases of the development. The paper provides a case study with a mathematical microworld that demonstrates how the three requirements are taken into account in the design of the user-facing components of the system and outline the methodology for formative validation of the intelligent support.

The authors would like to acknowledge the rest of the members of the MiGen project, which was supported by ESRC/TLRP Grant RES-139-25-0381 (see <http://www.migen.org>).

London Knowledge Lab
E-mail: {m.mavrikis,sergut}@lkl.ac.uk {e.geraniou,r.noss}@ioe.ac.uk

1 Introduction

The explosion of technologies for child-computer interaction is creating new possibilities for exploratory learning. Tablet PCs, mobile devices, and tabletops are emerging as alternative media where learning applications (from educational games to simulators) can encourage interaction and experimentation. Such learning applications, by providing direct access to a domain (or to its alternative representation) and by offering a context and appropriate tools to scaffold the learning experience, are aligned with constructivist theories of learning that emphasise the role of learners in constructing their own learning. As such, and regardless of their underlying technology, Exploratory Learning Environments (henceforth ELEs) are subject to the challenge (and often criticism) of exploratory learning: the need to provide significant pedagogic support [18, 24]. Research with microworlds, simulators, educational games and other ELEs repeatedly reports the need of student-adaptive support (see [25, 16] for examples). We elaborate on this on Section 2.

Unfortunately, in several settings, particularly in the classroom, the demands of individual support render dedicated human availability impractical. Despite attempts to find optimal ways to allow children to interact naturally (e.g. via direct manipulation) with the interface of a learning application, unless proper support is provided, these applications are difficult to integrate into the classroom routine, neither can they be used to support learning anywhere-anytime. Research is therefore needed on the computer-side of the HCI equation i.e., to investigate ways in which the computer can directly support students during exploratory learning.

The challenge lies in the fact that effective pedagogic support in ELEs comprises guiding students towards beneficial interactions without compromising their exploratory potential. It would be easy to develop an environment that can instruct students what to do and how but this would be against its constructionist stance — and probably not very effective (c.f. [8, 6]). This paper is focused on finding the elusive balance between control and freedom of exploration in intelligent exploratory environments. The challenge, therefore, is interdisciplinary and in order to be met requires both the underlying technology and appropriate pedagogic and interface design.

In previous work [12] we presented steps towards solving the technical challenges behind the implementation of intelligent support in exploratory environments. Other technical aspects are addressed by other teams (e.g. [7]). This paper is centered around the challenges associated with the design of user-facing elements and the feedback characteristics that fulfil the intelligent support requirements that emerge from pedagogical strategies that human teachers follow when supporting students.

Another set of challenges relates to iterative design and evaluation. In previous work we presented a methodology that employed iterative wizard-of-Oz studies that allowed the identification of specific technical requirements and evaluation of the effectiveness of several pedagogic strategies [23]. In this paper we answer calls for establishing ‘meaningful evaluation metrics’ [33] and focus on the design phase when appropriate intelligent components are integrated in the overall environment and studies can be performed under ecologically valid settings. We present the methodology

we employ that allows measuring students' perception of the intelligent support and identifying design problems and at various implementation phases.

Our research is taking place in the context of a project that is developing intelligent support for a mathematical microworld, called eXpresser, in classrooms. Therefore our particular attention has been twofold; ensuring that the exploratory nature of the environment is not compromised by the provision of intelligent support and that the teacher's role in the classroom is maintained and supported. The discussion is therefore generally applicable to classroom-deployed ELEs but also has implications for supporting learning in individual or other less traditional settings.

Section 2 presents a set of empirically and theoretically informed pedagogic strategies for supporting students interactions in ELEs. Section 3 outlines three key requirements and a four-axis characterisation of the possible feedback types that the system provides, which emerge from the effort of operationalising the pedagogical strategies. Section 4 presents as a case study the eXpresser microworld, and the design of the student-facing elements of the system that help fulfilling the requirements. Section 5 presents our approach to formative validation and Section 6 demonstrates its implementation in the context of MiGen discussing the findings as a means of demonstrating the potential of the methodology. Section 7 concludes the paper.

2 Pedagogic support in students' interactions with ELEs

Interacting with ELEs does not always require *a priori* understanding of the domain principles that enable the action to be performed. On the contrary, meaning is, at least partly, generated during interactions with the system. Although the design of an ELE and the guidance provided by structuring activities should, ideally, enable learners to connect their actions with the principles or ideas that are afforded by its design, inevitably explicit pedagogic support is required by a tutor, a teacher or a peer (see [15],[25, p70-71],[19]). It is aspects of this type of support that we would like to operationalise and enable an intelligent system to help where appropriate.

To inform the design of the intelligent support we drew on a collection of empirical evidence (observations from experts tutors and teachers including our own experience from interviewing students when interacting with exploratory learning environments) and theoretical perspectives. This way we developed a framework of necessary — but of course not sufficient — strategies to support students' to undertake activities in a meaningful manner. We can characterise these strategies as a bridge between a general 'framework of actions' [10] and domain specific strategies. These strategies informed the design of the intelligent support component for the eXpresser microworld, led to the emergence of the design requirements that we present in the next section and subsequently aided the domain specific implementation. We provide below a brief description of the categories that appear in Table 1.

Supporting interaction with the environment

Due to complexity of ELEs, apart from a careful introduction and familiarisation, students often require reminders of various affordances of the system. Although the need for this can of course be reduced by careful and intuitive design, the fact that students interact with several learning environments in the course of their studies

- | |
|---|
| <ol style="list-style-type: none"> 1. Supporting interaction with the learning environment <ul style="list-style-type: none"> Introducing or reminding students of the environment's affordances Helping students organise their working space 2. Encouraging goal-orientation <ul style="list-style-type: none"> Structuring activities Supporting students to set and work towards explicit goals 3. Exploiting learning opportunities <ul style="list-style-type: none"> Directing students attention Introducing cognitive conflicts, or counter-examples Supporting reflection on actions and on task |
|---|

Table 1 Pedagogic strategies for supporting students' exploration in ELEs

introduces difficulties in being able to remember how all of them work and what are their critical aspects.

In addition, we have observed interventions targeted specifically to helping students work effectively towards a specific goal or in order to help them perceive relationships between objects. For example, students may be asked to change the location of a shape, its properties or delete unnecessary shapes [22]. The importance of supporting perception explicitly has been noted in Gestalt psychology with the principles of perceptual organisation.

Encouraging goal-orientation

The importance of directing students' goals during exploration cannot be overstated. It is of course clear that the effective teacher's role is to orient students to work on well-defined investigations (e.g. "Investigate the relationship between these two shapes") aligned with a pedagogical agenda (see the 'play paradox' notion in [25] and other learning vignettes in [19, 14]). A difficulty that students face when solving problems in general is a tendency to lose sight of their overall goal as well as their immediate goals. As emphasised in [21], attention is usually caught up by current actions which are sometimes only intermediate towards a goal. In direct manipulation environments, some actions may not be directly relevant to the domain aspects that are being explored, yet necessary in order to reach a goal. Observations suggest that often simple questions like "What were you trying to do?", "Do you remember the question?" can orient students towards their goal. Before providing help, effective teachers establish the goal students are trying to achieve and may adopt the student's way of thinking about the problem or, in other words, try to maintain a balance between solving problems for students (or providing the way to solve a problem) and, leaving students on their own and unable to proceed if stuck.

Exploiting learning opportunities

There is of course a variety of ways that learning opportunities can be exploited. From a constructionist perspective and in order to avoid disruptive interventions and provide minimal guidance, empirical data and related research (e.g., [20, 21]) suggest that a successful strategy is to target students' attention and attempt to direct it towards objects and phenomena that are conducive to learning. For example, detect-

ing that a student has not noticed a particularly interesting aspect may lead one to ask a ‘nudging’ question (e.g. “Did you notice what happens when you resized the circle?”). Questions like this help students to start noticing invariants or other details which are important towards their learning.

It is worth noting, however, that an effective strategy might require refraining from saying anything if a student’s attention seems to be directed towards something that could lead to a learning opportunity. Examples include employing the need for making explicit a shape’s properties and relationships in dynamic geometry to introduce surprises or counter-examples that can demonstrate inconsistencies between the students’ actions and their understanding. We have observed the same techniques applied on virtual labs or tabletop interactions (c.f. [28]) particularly to challenge students to make predictions that lead to cognitive conflict or surprise. Lastly, supporting reflection as a metacognitive-skill is important and requires being aware on students’ actions and prompting them to reflect on them when necessary.

3 Computer-based Support for Adaptive Exploratory Learning Environments

3.1 Design Methodology and Requirements

We have presented elsewhere in detail our methodology for informing the design of intelligent support and technical evaluation of the various modules for the exploratory learning environment MiGen [23, 12]. In brief, the set of strategies presented in the previous section had to be operationalised in the particular environment. In order to achieve this we required — as in other occasions — an in-depth understanding of user behaviour that can only be developed by observing and analysing situations in their actual context. We therefore conducted studies where experts and students are physically separated, also known as ‘wizard-of-Oz’ (WOZ) studies. The experts employed a set of tools to allow them to provide support to students aged 11-12 year-old [23]. We also performed a series of classroom studies that allowed us to refine the set of strategies presented in the previous section and to identify the specific ones that we conjectured would be effective in our case.

This iterative design process allowed us to investigate and evaluate the general pedagogic strategies mentioned in the previous section in the light of empirical data with the specific system at hand. The attempt to operationalise these strategies led to the identification of several requirements that drive the design of the intelligent subsystem. While some of these are of course domain-specific and out of the scope of this paper, we provide three key requirements that are general and straightforward enough to be applicable for the design of other intelligent exploratory learning environments.

- (R1) **Students’ interaction should be, for the most part, interrupted only when they ask for help or to take advantage of a learning opportunity.**

Actions in a exploratory environment are continuous and compared to a structured question–answer environment there is no clear time when an interruption is appropriate. For example, in [22] students reported their discontent with an intelligent system’s provision of feedback when this was interrupting their dragging

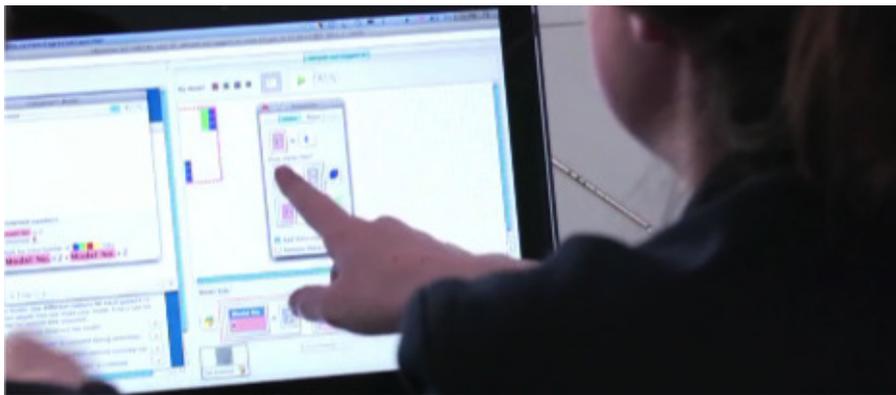


Fig. 1: Deictic language and gestures are necessary to provide support in ELEs

of an object. Similarly, in early WOZ studies it transpired that experts would often generate feedback in anticipation of some student moves but it often turned out that students managed to complete their goal without that help. That is, instead of providing just steps towards a solution challenges should be introduced whenever possible or students' attention should be brought to environment affordances that they have failed to notice. This relates to the pedagogic strategies for exploiting learning opportunities presented in Section 2 and related research that recommends recognising the need for students' autonomy and responsibility [14].

- (R2) **Support messages should be, to the extent possible, presented co-located to the objects they refer to.**

This is especially important, when a message pertains to something difficult to describe in words and when it requires the use of anaphoric or deictic language. The importance and effectiveness of such messages has also been discussed in [35] and was confirmed in several sessions with early prototypes of the eXpresser microworld, that required frequent pointing to aspects of the screen by the experts. Research on tabletop interaction also demonstrates this findings (see examples in [28]). An example from our data appears on Figure 1.

- (R3) **The support provided should be aligned with students' preferred solution strategy and adapted to the goals they are trying to accomplish.**

The ill-defined nature of ELEs and the loosely structured activities associated with them implies that in any given instance where a student needs help it will be likely that they can be scaffolded in more than one ways. Although a task usually provides an overall objective, in experimentation with several environments we have observed that students have a tendency to lose track of their goals or to stop or change focus of activity after reaching a successful sub-goal. Related research also suggests that establishing and maintaining an orientation towards task-specific goals is important [34] and that employing similar techniques helps students keep track of their goals as well as influencing their attitudes towards the use of the system [3, 22].

We recognise of course that the exact way that these requirements can be implemented depends on the system and the underlying technology. With these three requirements as a starting point, we present below a characterisation of the possible aspects of feedback provision by an intelligent system.

3.2 Feedback Characteristics

Among the many aspects that providing feedback entails, there are four especially important which we have formalised in the system and which form the basis for the design of feedback strategies.

First, the type of *interruption* represents the alteration of the student's action flow because of the feedback (see Table 2). The low-high-interruption characterisation of feedback in microworlds bears similarities to the immediate-deferred distinction of feedback made in [9], but that work was focused on structured systems with little exploration.

Type	Description
Integrated	Feedback is integrated with the normal interaction with the environment, e.g., a red border to show illegal placement, behavioural change of objects as a response to changes in parameter values, or formation of a desired or undesired output given an action.
Low Interruption	Only a notice of "feedback is available" is given, usually involving some little icon or sound. The action flow of the student is not explicitly interrupted. The student has a way to access the full feedback if so desired, e.g. clicking on the "feedback available" icon.
High Interruption	Feedback interrupts the interaction, which cannot continue until the feedback has been acknowledged; e.g. a modal prompt is opened in the middle of the screen, and the student cannot continue with the task unless an "OK" button is pressed, closing the window.
Explicitly requested	The student interrupts voluntarily his or her flow of interaction and requests help explicitly, e.g. pressing a "HELP" button.

Table 2 Feedback interruption type

Second, the level of *learner interactivity* relates to the interaction of the student and the feedback intervention itself. This can be *passive* or *active* depending on whether there is a need for some action on behalf of the user (e.g. choosing from a multiple-choice). The level of interactivity can also be at a form of *dialogue* where feedback consists on several steps that vary depending on the student's answers.

Third, the level of *system interactivity* indicates the effect of the feedback intervention on the exploratory environment. For example, the feedback can involve changing something at the *model* level (and subsequently some change in the user interface) e.g. a new object can be added, or the properties of an object can be changed, to trigger reflection on the student. Alternatively, the feedback involves changing

Level	Description
Passive	Users do not need any special interaction with the feedback
Active	Active feedback requires some thinking on the part of the user, e.g. choosing from a multiple-choice or solving a mini-puzzle.
Dialogue	Feedback consists on several layers or steps that depend on the user's answers.

Table 3 Learner interactivity

something at the UI level. e.g. a menu can be opened or a veil can be put over some objects to focus the student's attention on the others.

Type	Description
Model	The feedback involves changing something at the "model" level, which usually involves a subsequent change at the UI. E.g. a new object can be added, or the properties of an object can be changed, to trigger reflection on the student.
View	The feedback involves changing something at the UI level. E.g. a menu can be opened or a veil can be put over some objects to focus the student's attention on the others.
None	The feedback does not involve immediate changes.

Table 4 System interactivity

Lastly, related to the second requirement (R2), we found very important to specify carefully the most appropriate modality (or combination of them) to use for each intervention. In our case simple textual messages, occasionally with figures were enough but in other cases audio or video might be necessary. To adhere to the requirement of providing messages co-located to objects different systems will have of course different needs. In our case, we designed a toolkit that allows the provision of information in rectangular or elliptical call-outs that have a tail-end that can pointed to different locations on the working area. Examples of specific implementations of this idea appear in Figure 5 of Section 4.¹

The requirements in the previous section and the characterisation of feedback above does not make any assumptions about the nature of the ELE or the pedagogical strategies employed. They permit the implementation of a variety of pedagogical approaches from providing structured guidance to allowing total freedom to explore and can therefore guide the design of intelligent support in other applications. In the current stage of development the architecture described in [12], the general feedback toolkit mentioned above and the representation of the interventions provide us with

¹ The inspiration for similar messages comes from previous work in implementing tutorials for Alice; a 3D programming environment for introductory computing [17]. In the context of the MiGen project we developed a Java Feedback Toolkit (JFT) that allows the generation of such messages in a general way and therefore can be used in other Java-based environments. See <http://www.migen.org> for details.

the required flexibility to experiment for the most effective approach as co-designed with teachers and experts. The next section presents the approach we have taken in the design of the specific microworld eXpresser and how the requirements and feedback characteristics discussed here formed the framework for the design decisions for the support that the system provides.

4 The design of Intelligent Support in the eXpresser microworld

4.1 eXpresser: a Microworld for Algebraic Generalisation

The MiGen project revolves around the development of intelligent support for a mathematical microworld, called eXpresser, to help secondary education students develop an appreciation for algebra. The rationale of the project, its aims and the environment are described in detail in [26]. In the microworld students undertake activities that require identifying relationships that underpin figural patterns (like the ones that appear in Figures 2,4,5). Similar activities are often found in the UK National curriculum and emphasise the structural aspect of patterning rather than the purely numerical. As explained in more detail in [26] this is a key difficulty students face.

The eXpresser microworld allows students to build their own constructions (referred to as ‘models’) and expressions for the activity they are undertaking. In order to do that, students can use building blocks of square tiles to make patterns. These can be combined to models. To represent the generalities and relationships they perceive, students can use entities that act like ‘variables’. A model with variables can be ‘animated’. This results in the variables taking random values providing a rationale for generality. Figure 2 shows the key features of eXpresser.

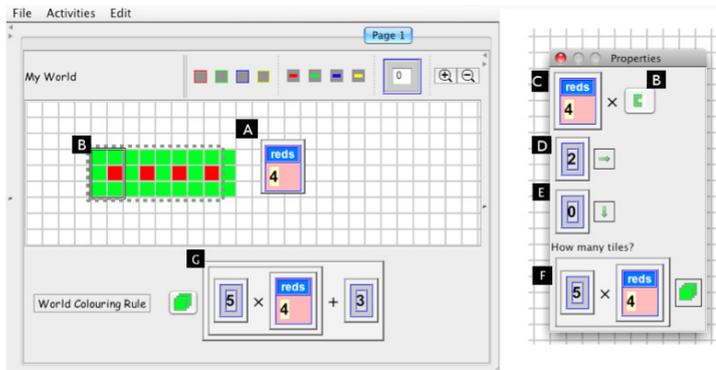


Fig. 2: Key features of the eXpresser microworld: (A) An ‘unlocked’ number that acts like a variable is given here the name ‘reds’ to signify the number of red tiles in the pattern. (B) A building block repeated to make a pattern. (C) Number of repetitions (in this case, the value of ‘reds’). (D,E) Number of grid squares to translate B to the right and down after each repetition. General expressions describe the number of tiles required for the specific pattern (F) and for the ones surrounding the red pattern (G).

4.2 Tools for indirect and direct help-seeking

In order to conform to the first requirement (R1) and to maintain students' agency and allow freedom of exploration, we designed the following tools that allow students to seek help directly when needed and enable the system to provide appropriate feedback, in addition to any possible unrequested feedback, when required.

4.2.1 Suggestion button

As it is important for students to have ownership of their actions and their solution, we introduced a suggestion button (see Fig. 3). The button is disabled unless the system 'observes' something that warrants feedback. Instead of interrupting the student, the icon changes to indicate the existence of a suggestion and the button gets enabled. This way, if in the meantime the student's actions change the state of the microworld and the feedback becomes irrelevant then the button might get disabled again, effectively achieving not having interrupted the student for something that they did not need help after all. On the other hand, if the student seems unable to progress, then a suggestion is available to scaffold their interaction.

4.2.2 Drop-down sentence maker

A direct result from early WOZ studies was that in the absence of dialogue, even experts observing students' interactions lacked the necessary information that would help them provide feedback on students' help request. Therefore, we enhanced the system with a simple 'sentence maker'; a list of drop-down menus that allow the construction of a sentence before asking for help from the system.

In general this approach brought a twofold benefit. First, it provided the system with an indication of the nature of the student's needs. Second, because we designed the interface so as the help button is enabled only after constructing a sentence, we hypothesised that this would also provide an incentive to reflect on actions and terminology used in the microworld. Similar sentence makers have been used successfully before but either as a means to reflection [1] or student communication [32].

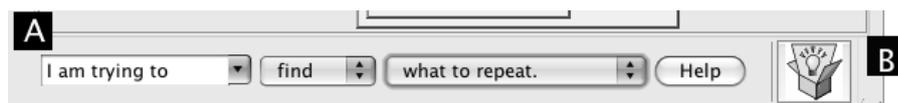


Fig. 3: (A) Help-seeking area with drop down menus and (B) suggestion box for feedback provision.

4.2.3 Explicit Annotation of Goals

To adhere to the third requirement (R3), we introduced an explicit list of goals that students have to reach in order to move to the next tasks (see Fig 4). Accordingly, the

activity designer must specify the goals of the task and provide declarative rules for their completion. To allow freedom to the students to interact in their own pace and also to reflect on their actions, they are responsible for clicking the check button that corresponds to the goal that they feel they have accomplished. The system provides feedback on the completion of the goal or suggestions on what to do next in order to complete it. Similarly, a button besides each goal can provide adaptive help on the particular goals. The latter, in combination with the drop-down sentence maker above allow us to conform to the third requirement (R3).

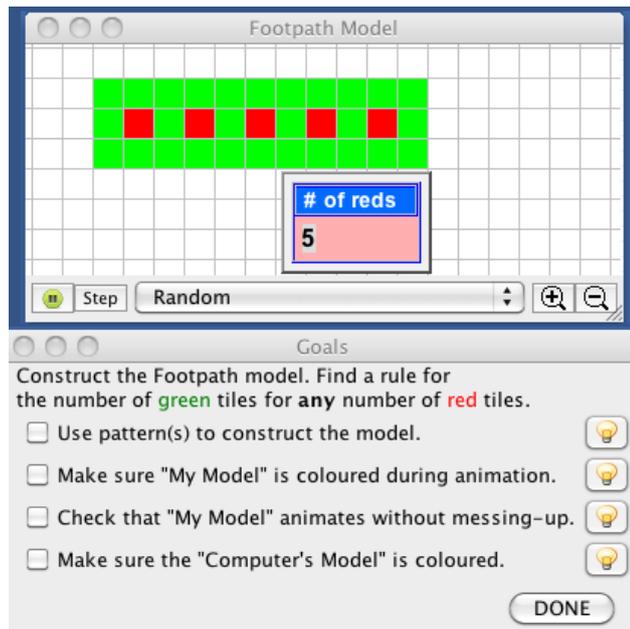


Fig. 4: An activity and its goals explicitly listed. Students can check a goal when they consider it completed. If the goal is not achieved the students will receive help towards the goal. Students can also click on the bulb icon to get more help on the particular goal. In both cases the help is adapted to the state of the microworld.

4.3 Feedback Strategies

For the actual provision of feedback we have followed a layered approach, which divides the problem into four layers: model and events, analysis, reasoning, and feedback [12]. Of particular concern here is the feedback layer that takes the relevant decisions to produce specific *feedback interventions* i.e. determining the specific wording of a feedback message according to its severity, relating it to particular elements in the microworld, and adapting it according to the learner model.

In our implementation, the feedback layer provides scaffolds consistent with the principles of 'contingent tutoring' [34]. This is achieved first by grading the phrasing

of the feedback according to the following three types: ‘nudge’ questions, comments and suggestions and subsequently the degree of intervention on the students’ construction. Some of the strategies afford additional adaptation. They are sequenced according to the level of specificity of the information provided. In collaboration with the teacher advisers the reasoning layer is authored in such a way that after the identification of the need to follow a certain strategy, putting it into effect is adaptive to the previous help the student has received on the same issue as well as their general progress within the task. Below we provide a detailed scenario and more examples appear in Table 7 in the Appendix.

4.4 Example Scenario

Using the MiGen system, an activity designer can develop and set different activities in eXpresser. We envisage that eXpresser is introduced gradually through activities that allow students to familiarise first with its features, progressively undertaking more complex activities, but still scaffolded by specific goals and the intelligent components of the system. Eventually, eXpresser can be used for more open-ended activities where the intelligent support may not be required.

To demonstrate the role of intelligent support in the system, we present here a particular example scenario. The scenario is relevant in activities where students are asked to find a general rule that gives the total number of tiles for any number of repetitions of a pattern’s building block. Among other difficulties, which due to space constraints we do not discuss here, novice students who undertake such an activity tend to overlook the relationship between the number of tiles in a building block and the number of times it is repeated and find difficulty in articulating this relationship in a symbolic way.

In order to help students appreciate the need for a general approach, a particular pedagogical strategy was identified. The strategy is referred to as ‘messing-up’ and alludes to a challenge that requires students to construct models that are impervious to changing values of the various parameters of their construction. Changes to a variable’s value should not ‘mess-up’ their model. This strategy is inspired by previous work in dynamic geometry environments [13] and we focus particularly on this as it demonstrates the potential of using the approach present here in other microworlds.

In this activity, one of the goals that the students must accomplish is to make their model animate and remain coloured. As mentioned in Section 4.1 animation is associated with the unlocking of numbers. Once students unlock a number, their model can be animated as the computer is ‘allowed’ to change the value of the unlocked number. This introduces a challenge in that a specific expression or number for the amount of tiles in the pattern will not be always be correct. Let us demonstrate how a student, Ted, is supported by the system when undertaking such an introductory activity. The snapshots in Figure 5 demonstrate Ted’s ‘dialogue’ with the system, using a sequence of questions, nudges, challenges, and conflicts that provide incentives and opportunities to analyse and express a general rule for the number of tiles in the pattern.

Ted has coloured his pattern (see Figure 5a) by placing the number 12 in the property box, which he found—as many students do—by counting the number of squares on his canvas. Ted fails to notice the lack of generality in his model, which will remain coloured only for 4 repetitions of his L-shaped building block.

According to the first requirement (see R1 in Section 3.1) the system does not interrupt but generates a possible message for Ted (the suggestion button in Figure 2B gets highlighted). Ted reaches an impasse as prior to the prompt he considered his model coloured and the corresponding goal accomplished. He therefore requests a suggestion from the system. A message appears (see Figure 5b) directing his attention to the unlocked number of repetitions (c.f. requirement R2). Note that the message here is designed on purpose not to give away the answer (which would be trivial) but to challenge Ted and to relinquish the control back to him.

Ted takes up the challenge and changes the number of repetitions to 6. He notices that the pattern is not coloured and corrects the number of tiles needed - but still uses a specific number 18. In accordance again with R1, the system this time intervenes with a prompt to encourage him to reflect on his action and articulate what he had done (Figure 5c). Ted begins to pay more attention to his actions. He animates the pattern and notices that it is not always coloured. He stops the animation and colours the pattern again (5 repetitions, 15 tiles).

Ted realises that he has not achieved the goal of colouring the pattern for any number of building blocks and asked for help using the sentence maker (Figure 2a). In Figure 5d the prompt he receives emphasises (consistent with R3) the need to stop the pattern being messed-up given that the system will now change the unlocked number to a different value. Referring to the feedback characteristics in Section 3.2, this is an example of explicitly requested help with system interactivity at the model level. Before making the change, the system draws Ted's attention to the need to keep an eye on the link between the number of repetitions and the number of tiles in the building block (Figure 5e). The number of repetitions changes then to 8. Having drawn his attention to the need to link the two quantities Ted eventually creates an expression (Figure 5f). We do not claim that the latter is an easy step and precisely for this reason we acknowledge the role that a teacher or a more able peer could play here in helping Ted achieve his goal. However, the preliminary analysis of students' interactions with eXpresser suggests that the feedback strategy followed in Figure 7 and similar scaffolds that the system provides gradually become internalised. In later tasks, similar prompts are no longer necessary as Ted began to realise that a general rule for the number of tiles in a pattern could always be expressed as a multiplication of the number of repetitions of a building block and the number of tiles in the block.

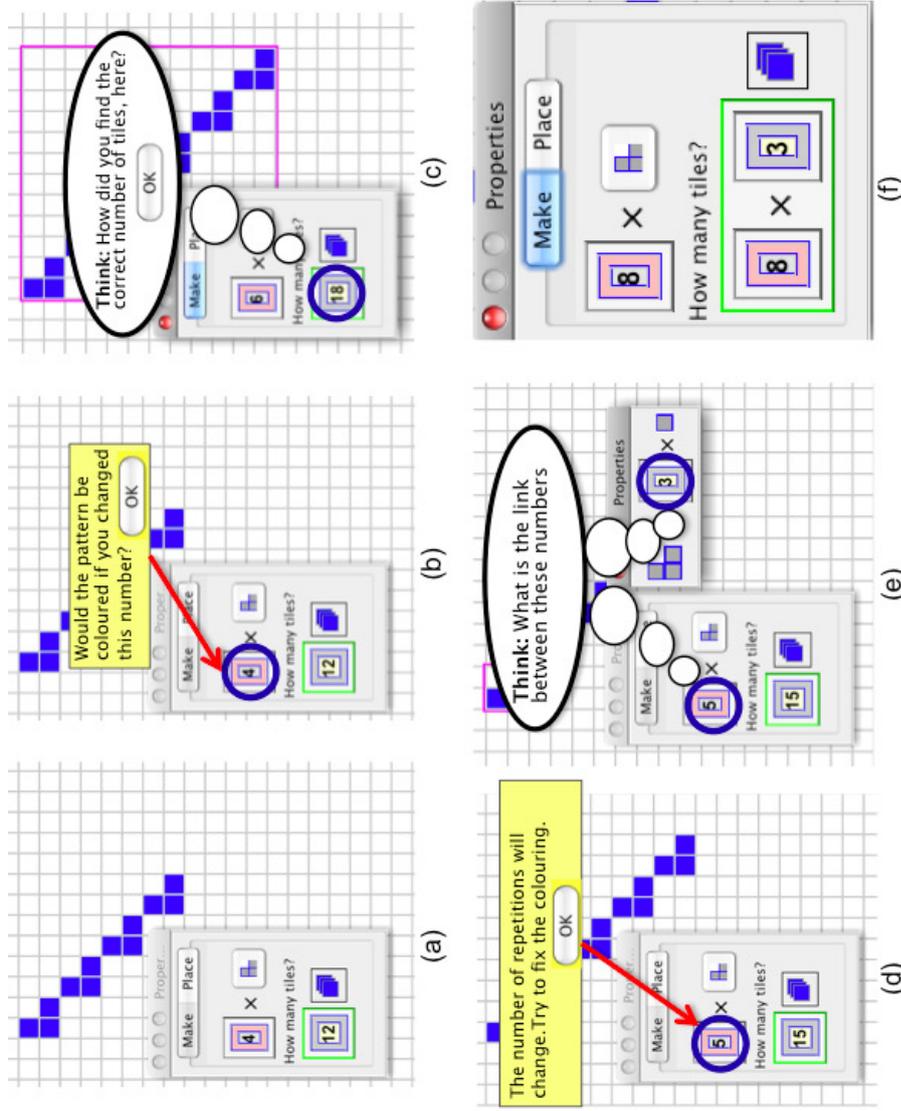


Fig. 5: An example of intelligent support in eXpresser (narrative in Section 4.4).

5 Iterative Formative Validation

Design of intelligent support for ELEs is a lengthy task: developers need to create and integrate many components, implement several feedback strategies and messages before the system can be used in a classroom.

A major concern therefore during design is how to prioritise implementation and how to measure development progress. Notwithstanding the required summative pedagogical evaluation, development of such complex systems requires performing several stages of what we refer to as ‘formative validation’ along the lines of the usual ‘formative evaluation’ but different in the sense that we are concerned with checking the validity of the implemented system. We assume here that (a) the educational value of the exploratory environment has been established (e.g. in one-to-one or small-group sessions) and (b) that pedagogical strategies have been implemented after modelling effective human strategies (the only available golden standard) and have been evaluated

These assumptions, however, do not guarantee that the system can be integrated into the classroom ecology. Evaluation poses unique challenges both from a pedagogical but also from an HCI point of view. For example, the field already recognises the need for adapting techniques traditionally used with adults to evaluate interfaces designed for children [30, 31]. More importantly, as also discussed in [11] evaluation based on learning outcomes can often be problematic particularly since the effects from learning in exploratory environments cannot be observed in the short sessions available for early prototype testing let alone the various uncontrolled factors that come into play in classroom evaluations. These common difficulties of children-computer educational evaluation are exacerbated in intelligent exploratory environments as evaluating the system holistically would not allow us to identify which aspects of the system and particularly which adaptive strategies are effective or not (c.f., [27]). Additionally, the question of what constitutes ‘useful’ support is difficult to answer particularly in the case of ELEs. Lastly, the widely acknowledged need in HCI for involving users early on in the design process and performing formative studies is condition *sine qua non* in educational applications [23].

5.1 Measuring children’s perception on the intelligent support

It is clearly vital to have an appreciation on children’s opinion on the exploratory environment in general but also gauge their perception of the intelligent support in particular, in order to help us make design decisions such as the actual messages, their appearance and the general approach we are taking. As mentioned already, we conduct one-to-one iterative evaluations where we follow a talk-aloud protocol or engage a pair of students in discussions about what they understand from the system’s messages. These methods, however, powerful as they may be, can be quite distracting for the children involved, are very demanding on the researchers and would be very difficult to perform in a classroom or even with a small group of students. As reasonable ecological validity is paramount, it is important to strive for some collective measure and an efficient way to get answers to design questions, for example, during classroom studies. Therefore when conducting studies with the intelligent system we

introduce surveys due to their reported usefulness in usability studies (see a review in [5]), to support both our qualitative and log-file analysis.

Several surveys have been designed for children in technology studies (see a review in [30]). Based on positive success in the literature with visual analogue scale surveys that employ pictorial representations that children can relate to in order to respond to pertinent questions, we decided to employ the visual analogue scale from the Fun Toolkit [29] (see Fig 6) albeit not to measure the construct of *fun* but to gauge students' perception of the support provided by the system.

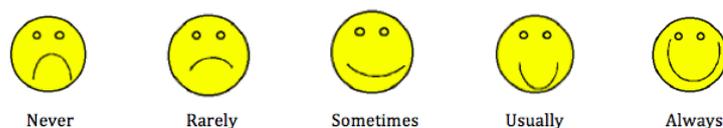


Fig. 6: A visual analogue instead of a textual 5-point Likert scale (adapted from [29])

We designed, therefore, a questionnaire that employs a 5-point Likert scale for several constructs that we identified as important based on students' comments on early one-to-one interviews or discussions with their peers. Initially these included *helpfulness*, *repetitiveness*, *comprehension*, *relevance*, *perceived necessity* and *careful reading* of the feedback messages. After pilots of the questionnaires in small groups we decided to keep only *helpfulness*, *repetitiveness* and *comprehension* (see Table 5) mainly because they were the ones that students seemed to understand easily but also in order to reduce the number of questions, particularly as we identified ways to measure some of these from posthoc log-file analysis (as discussed in the next section).

- | |
|---|
| <ol style="list-style-type: none"> 1. Did you find the system suggestions helpful?
Not helpful at all, A bit helpful, So and so, Helpful, Very helpful 2. Did you find the help repetitive?
Very repetitive, Repetitive, Sometimes, Little, Not at all 3. Did you understand the help from the system?
Never, Rarely, Sometimes, Usually, Always |
|---|

Table 5 Questions to measure *helpfulness*, *repetitiveness*, *comprehension* employing the scale in Fig 6

Although there are of course additional constructs that are important to measure in intelligent systems for children, we think that this triad can provide a good picture of students' impression of the system. Consistent with the findings in the literature, students responded positively on the questionnaire and perceived the smileys as expected. For this reason we did not design our own smileys but adopted the ones from [29]. We advocate therefore including the three questions that appear in Table 5 in surveys and assigning appropriate scales. Section 6.2 presents briefly, as a case

study, the interpretation of results from surveys after the usage of the MiGen system in classroom.

5.2 Validation Metrics

We define three important metrics to observe during the design and development of the system *relevance*, *coverage*, and *scope* of the feedback. Relevance is a measure of how many feedback interventions made by the system are relevant for the student i.e. appropriate to the situation as judged by experts (our only available gold standard). Coverage is a measurement of the feedback needed by the students that could be handled by the system. Finally, scope measures the completeness of the pedagogical analysis that drives the design of intelligent support for the microworld. Section 6.1 presents one approach for measuring these.

$$\textit{feedback relevance} = \frac{\# \text{ of relevant feedback interventions by system}}{\# \text{ of feedback interventions by system}} \quad (1)$$

$$\textit{feedback coverage} = \frac{\# \text{ of feedback interventions by system}}{\# \text{ of feedback interventions by system or humans}} \quad (2)$$

$$\textit{feedback scope} = \frac{\# \text{ feedback interventions known before the study}}{\# \text{ feedback interventions known after the study}} \quad (3)$$

It is worth mentioning that we do not present these metrics as a means of qualifying the fitness of the final system for its pedagogical purpose. The metrics provide just a broad understanding of the current state of the system as it progresses towards its final state, where it can undergo a summative pedagogical evaluation. The process of determining values for each metric and their evolution, as the intelligent support subsystem is expanded and more formative studies are performed, provide evidence of the progress of the system and alert of possible problems.

All metrics have a range of $[0, 1]$ with 1 being the ideal. However, depending on the nature of the learning environment, we do not expect coverage to be 1 since, as we discussed, in systems that consider the teacher as crucial part of the classroom culture and we assume that there will always be unforeseen situations for which we did not design support, or that were intentionally left for the teacher to provide.

6 Formative Validation of the intelligent support in MiGen

6.1 Small group studies

This section presents, as case study, an approach for deriving the metrics in Section 5.2. We conducted a series of small group studies usually with six 11-year-old students interacting with the eXpresser microworld. Among other pedagogical goals

(out of the scope of this paper) one of the objectives was to measure the progress of the intelligent subsystem. Examples of the feedback strategies that were implemented appear on Table 7 in the Appendix.

In these studies we tried to reproduce a situation as similar as possible to a normal classroom. Each student was given a computer with eXpresser running, and a description of the task they were asked to undertake. They were told to work on the task for 60 minutes, and were instructed to ask for help from the system if needed. The students interacted with the computer in their usual classroom at the school where no other distractions were present. Two researchers stayed with the students at all times. They presented the task to the students, and observed their performance during the study.

Researchers were aware of the capabilities of the intelligent support subsystem and its limitations, i.e. they knew which problems were expected to receive adequate support from the system and which ones were not at this stage. The researchers have appropriate pedagogical experience with eXpresser, the task and the subject domain to support the students if needed. They were therefore ready to provide feedback that could not be provided by the system at this stage. They were also paying attention to the feedback provided by the system in case it was not relevant for the student's needs; if that was the case, they were responsible of acting and providing appropriate support.

The researchers kept track of the whole session (by means of notes and audio recording of the interaction, synchronised with video recording of the students' screen) paying special attention to feedback interventions, either by them or by the system. Feedback interventions by the system were marked as "relevant" if they helped the student (according to the researcher's expert judgement), and "not relevant" otherwise. They were marked as "covered" if they related to the goals or the feedback strategies specified above, or "not covered" otherwise.

In our case, there are three causes for "not covered" feedback. First, the intervention can be related to a feedback strategy that is not covered by the system at the current stage. Second, some feedback strategies have been identified as falling beyond the scope of the system, usually because they require capabilities that the system cannot provide (e.g. natural language dialogue). These feedback strategies are left for teachers, and although are detected (and appropriate information is provided to the teacher), they are not directly addressed by the system. Finally, sometimes students find new approaches to tasks that experts have not seen before, and therefore need support that has not been taken into account. Researchers (or a teacher acting as researcher) had the responsibility of covering for the system's limitations (i.e. give feedback to the student) and take note of the intervention, the feedback strategy employed, and the reason why their help was needed (i.e. non relevant, non covered, left for teacher, or new feedback strategy).

Researchers' feedback prevented the students from being stuck if they had problems and the system was not able to provide appropriate feedback. There is a methodological and a pragmatic reason for this. Methodologically, the human presence and freedom to intervene allows the study to cover different aspects of the intelligent support subsystem, rather than testing only a specific scenario or step in a task. From a pragmatic point of view, it is difficult to find schools and students to collaborate with

educational research projects and/or try educational tools that are still in the research stage. We wanted therefore to provide a constructive learning experience for students that also was a pedagogically valid activity from the point of view of the teachers, making it worth their time and effort.

6.1.1 Indicative results

A sample of the results from the latest of these studies are summarised in Table 6. The table shows the number of feedback interventions received by a student during the course of a task with eXpresser according to different criteria. For example, student-1 received help in 42 occasions: 20 times the help came from the system and was relevant, 4 more times the system offered irrelevant help and the teacher had to intervene to cover the student's need, and 18 more times the teacher intervened to help where the system could not. From these 18 non-implemented feedback strategies, 6 were planned for a future version ("designed"), 9 are considered too complex or costly to implement and are left for the teacher by design ("teacher"), and 3 had not been considered ("not designed"). Sometimes the teacher acted because the student requested help ("*requested*") and sometimes she acted on her own when she considered the student would really benefit from the intervention ("*unrequested*").

<i>student</i>	addressed by system		<i>cover</i>	addressed by expert						<i>relevance</i>	<i>coverage</i>	<i>scope</i>
	relevant	not relevant		not implemented								
				designed		teacher		not designed				
				req	unreq	req	unreq	req	unreq			
1	20	4	4	3	3	9	0	3	0	0.83	0.57	0.90
2	15	3	3	2	1	5	0	4	1	0.83	0.70	0.83
3	17	3	3	1	3	8	0	2	0	0.84	0.59	0.91
4	19	4	4	3	0	3	0	4	0	0.84	0.58	0.87
5	17	3	3	3	1	3	4	3	0	0.85	0.59	0.88
6	14	2	2	3	3	5	4	0	3	0.87	0.47	0.91
total	102	19	19	15	11	33	8	16	4	0.84	0.58	0.88

Table 6 A sample of the results of a small-group formative evaluation for relevance, coverage and scope.

Former studies have allowed us to identify issues particularly with the relevance of the support provided due to the fact that the system was ignoring a lot of context related to the latest accomplished goal. We consider the current implementation of the intelligent support subsystem adequate for the identified needs particularly as there was a low number of errors (most non-relevant system interventions were the cause of "not implemented yet" not "new feedback strategy we hadn't thought of"). On close examination, however, the main technical concern is the need for a more robust prioritisation mechanism that assigns more weight to strategies that relate to the object that was affected from the latest students' actions rather than the violated constraints and unaccomplished goals as the system does at the moment. This requires more

planning and problem solving capabilities on behalf of the system. The coverage is around 58% because in the MiGen system several of the identified strategies are left for the teacher who, informed from the Teacher Assistance Tools, remains the best support for some of the thorny conceptual problems that students face in the domain of algebra.

The high *scope* indicates that the pedagogical analysis has identified the majority of feedback instances required in the microworld (at least for the particular tasks we are investigating). Most of the “not designed (new)” interventions relate to the need of supporting students’ help-seeking behaviour i.e. the actual use of the help features presented in Section 4.2. This is related to a research question outside the scope of this paper but worth discussing. Relevant research suggests that, when interacting with intelligent learning environments, many aspects of students’ affect and motivation come into play leading to help abuse or help avoidance [2]. However, little is known about the actual use of students’ help-seeking behaviour in ELEs and its relation to learning and the few studies that investigate the issue report conflicting results or are targeted to systems in well-defined domains [34, 4].

6.2 Classroom studies

As part of our effort to try the system in situations as ecologically valid as possible, we have also conducted larger classroom studies. We have tried the system in actual classroom sessions mainly in two schools (one in London and one outside London) over a period of 2-3 hours spread in different days of the week. Apart from the opportunity to test the pedagogical validity of the system, these studies allow collecting data about the overall students’ perception of the system. It would have been difficult and time consuming to do that only from oth

Despite the fact that, to some extent, the data from one-to-one and small-scale studies provide more detailed (mostly qualitative) data about the use of the system, having a measure of the overall classroom perception would be difficult without the questionnaires. As an example, Figure 7 presents boxplots of the answers to the questionnaires in Table 5 after our latest classroom studies with 12-13 year-old students. In the case of the second school we conducted a study with the same class after 6 months and with more difficult activities (2 students were absent and therefore only 21 students answered the questionnaire).

There are several things that can be gleaned or analysed using the surveys. We have found them useful mostly as a means of identifying next steps and prioritising our efforts. For example in early versions of the system in Nov 2010 we decided to dedicate more resources in improving the comprehension of the system’s feedback as a large proportion of the answers (47.4%) were below the neutral response in the Likert scale (a coarse interpretation of the results also shows that the average score was 2.58 marked with a point in the boxplot in Figure 7). This provided strong evidence about the need to pay more attention on the comprehension of the messages first. It would have been difficult to obtain this information since in one-to-one studies students have a tendency to avoid providing negative feedback about the system and do not easily admit that they do not understand something — even if that is clear from their actions. Accordingly, the questionnaire responses forced us

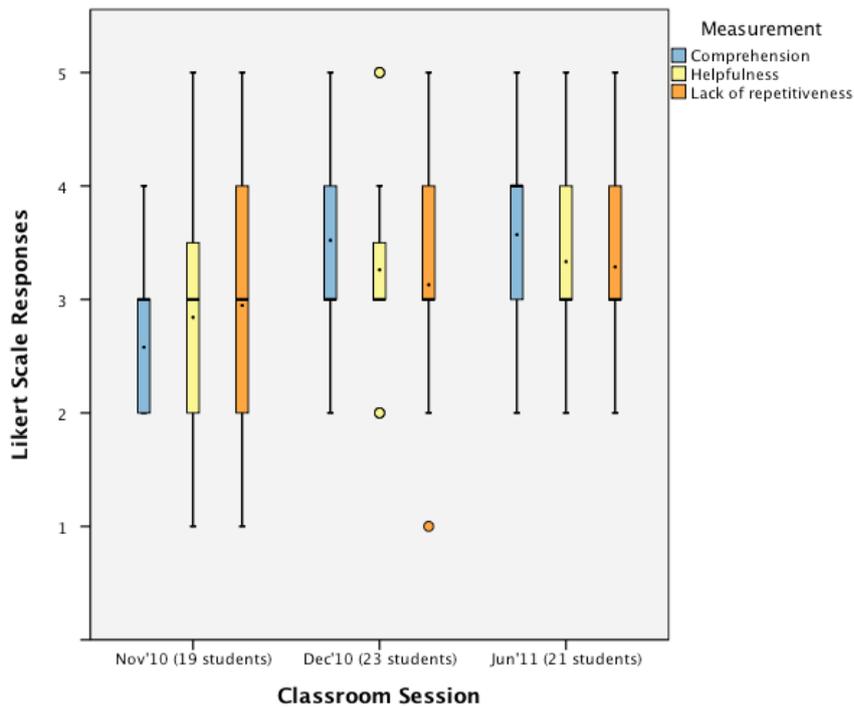


Fig. 7: Boxplots for comprehension, helpfulness, and lack of repetitiveness, across three different sessions as perceived by the students after three classroom studies.

to pay more attention to the help messages in further design sessions with the help of the teachers and based on one-to-one interviews with some of the students with low answers on this item. The answers to the questionnaires also acted as a reminder to student and together with their recorded interaction helped during retrospective walkthroughs to have discussions about the system and how to improve it. Without claiming any undue statistical rigour behind the results, the perceived comprehension at the next study seems to suggest that our effort was effective. A large proportion of the students (43.4%) provided an answer above the neutral response. Regardless, analysis of the questionnaires helped identify the individuals that seemed to have issues with understanding the messages (especially the outliers) and another iteration of improvements was performed. Most of the problems with comprehension related to long texts that were subsequently shortened. Other problems related to terminology that suggested that we should pay more attention to the way the system is introduced in the classroom. It is encouraging to observe that in the next iteration (either because of familiarity with the system and/or because of the care that the teacher took in introducing some misunderstood concept) a larger percentage of the same students (67%) answered above the neutral response.

We used the other two indicators in a similar manner. In early versions of the systems repetitiveness was considered an important limitation. Even if a feedback

message was relevant or pedagogically valid the fact that it was repeated was clearly affecting students' perception of the feedback functionalities and a deterioration of its usage, even in occasions where the logs suggests that an intervention would have been quite relevant. We attempted to improve perceived repetitiveness by enabling the provision of the same feedback message in different wordings, mimicking human teachers. Repeating a message in a different wording has the potential to be understood better and, even if semantically the same, students do not easily perceive the similarity. With regards to helpfulness, we discerned the initial high proportion of responses below neutral (47.3%) as a call for making more options available. Students who replied that they do not find the help of the system helpful also reported that their main problem was that they could not identify which option to select from the dropdowns on Figure 3. Although we simplified the list of options, pragmatic and resource limitations have not allowed us at the time of this writing to enable the system with more reactive capabilities on students' requests. Despite the fact that after the latest study a large proportion of answers is above neutral (42.8%), the low average still acts as a reminder for future work.

7 Conclusion

In this paper, we argue that it has become paramount to understand how to design and evaluate intelligent exploratory learning environments. Despite the growing interest in the use of artificial intelligent techniques to support learning with and from exploratory environments, we believe that computational analysis and reasoning will be of little help when designing exploratory environments for children without a nuanced understanding of the child-computer interaction and the types of support they need. Our focus has been secondary education students but this point is more general and perhaps even more pertinent for younger children. Designing and developing support in an exploratory environment is a challenging and costly task: it is essential to engage in such an endeavour with a clear idea of the goal and indicative signposts that help the design as it unfolds.

Building on the literature on exploratory learning on the one hand and observations from experts' interactions with 11-13 year-old students in face-to-face and wizard-of-Oz studies, we have identified three general requirements that acted as signposts and which we believe are general enough to help other designers of computer-based intelligent support: (i) students should be interrupted rarely in their interaction, (ii) support and feedback messages must be co-located with the objects that require student action, and (iii) support should be goal-oriented and in-line with the student's approaches and interaction. Contradicting these requirements by eager interventions, lack of attention-focusing means, or by failing to understand the purpose of students' actions and rationale for help, tends to result in less effective learning due either to students' disengagement and frustration or to the provision of unsolicited answers on inopportune interventions. The paper depicts specific ways in which these three requirements were taken into account in the design of support for the eXpesser mathematical microworld.

Design requirements, however, can only bootstrap the design and implementation of a system; they do not guarantee its efficacy. Intelligent systems are still difficult

to create, and require the combination of varied types of expertise along an extended period of time. The danger of finding out that a system is not really useful enough in the end is non-negligible, and this acts as a powerful deterrent against many efforts to provide intelligent computer-based support in exploratory environments. In order to reduce this risk and to facilitate the implementation of future intelligent exploratory learning environments, a design team requires qualitative and quantitative methods that can indicate progress and provide insights into students' view of the system. To this end, the paper presents three indicators of student perception of helpfulness, repetitiveness and comprehension, that provide useful insights into where to focus attention and complement any qualitative research on the effectiveness of the technical approach taken. In addition, the three metrics of relevance, coverage, and scope can act as early warning signs of problems in the implementation or the pedagogical analysis, providing ample opportunity for correction and improvement at every point in the development.

We advocate following a similar procedure after each iteration of the system and its usage in classroom or other activities. Even a crude analysis can provide indications of students' perception and inform design decisions and priorities. The set of metrics relevance, coverage and scope can be used both within successive iterations of a design project and across projects as a means of comparison. With regard to the three indicators we employed, further investigation is needed to establish whether they are sufficiently different and whether they provide a coherent picture of students' perception. It is important to determine whether what they measure is valid across student cohorts and across design projects. Other teams may want to use the same student perception indicators to further this work or employ the method and instruments described here to introduce new ones. In our case, although there are of course several factors that influence the results, the process of establishing values for the set of metrics and the analysis of the questionnaire responses after each iteration of the system indicated that our efforts between successive iterations were not in vain. In future work it would be interesting to investigate gender or ability differences in the way students perceive helpfulness or repetitiveness. A preliminary analysis we have conducted begs the hypothesis that weak students find the help less repetitive and more helpful than higher ability students. Such hypotheses, once confirmed, can lead to a better understanding of the interaction between students and system with respect to help-seeking behaviours.

References

1. Alevan V (2002) An effective metacognitive strategy: learning by doing and explaining with a computer-based Cognitive Tutor. *Cognitive Sci* 26(2):147–179
2. Alevan V, McLaren B, Roll I, Koedinger K (2006) Toward Meta-cognitive Tutoring: A Model of Help Seeking with a Cognitive Tutor. *International Journal of Artificial Intelligence in Education* 16(2):101–128
3. Arroyo I, Ferguson K, Johns J, Dragon T, Meheranian H, Fisher D, Barto A, Mahadevan S, Woolf BP (2007) Repairing Disengagement With Non-Invasive Interventions. In: *Proceeding of the 2007 conference on Artificial Intelligence in Education*, IOS Press, Amsterdam, pp 195–202

4. Baker RS, Corbett AT, Koedinger KR, Wagner AZ (2004) Off-task behavior in the cognitive tutor classroom: when students "game the system". In: CHI '04: Proceedings of the SIGCHI conference on Human factors in computing systems, ACM, New York, NY, USA, pp 383–390
5. Bark I, Folstad A, Gulliksen J (2005) Use and usefulness of HCI methods: results from an exploratory study among nordic HCI practitioners. In: International Conference on HCI
6. Bransford JD, Franks JJ, Vye NJ, Sherwood RD (1989) New approaches to instruction: because wisdom can't be told. In: Vosniadou S, Ortony A (eds) Similarity and analogical reasoning
7. Bunt A, Conati C (2003) Probabilistic Student Modelling to Improve Exploratory Behaviour. *User Modeling and User-Adapted Interaction* 13:269–309
8. Chevallard Y (1991) *La transposition didactique - Du savoir savant au savoir enseigné*. Grenoble: La Pensée sauvage (1re éd., 1985).
9. Dempsey JV, Wager SU (1988) A Taxonomy for the Timing of Feedback in Computer-Based Instruction. *Educational Technology* 28(10):20–25
10. Disessa AA, Cobb P (2004) Ontological Innovation and the Role of Theory in Design Experiments. *Journal of the Learning Sciences* 13(1):77–103
11. Do-Lenh S, Jermann P, Cuendet S, Zufferey G, Dillenbourg P (2010) Task performance vs. learning outcomes: A study of a tangible user interface in the classroom. In: Wolpers M, Kirschner P, Scheffel M, Lindstaedt S, Dimitrova V (eds) *Sustaining TEL: From Innovation to Learning and Practice*, Lecture Notes in Computer Science, vol 6383, Springer Berlin / Heidelberg, pp 78–92
12. Gutierrez-Santos S, Mavrikis M, Magoulas G (2010) Layered development and evaluation for intelligent support in exploratory environments: The case of microworlds. In: Aleven V, Kay J, Mostow J (eds) *Intelligent Tutoring Systems*, Lecture Notes in Computer Science, vol 6094, Springer Berlin / Heidelberg, pp 105–114
13. Healy L, Hoelzl R, Hoyles C, Noss R (1994) Messing up. *Micromath* 10:14–17
14. Hoyles C, Sutherland R (1989) *Logo Mathematics in the Classroom*. Routledge
15. de van Jong T, Joolingen WR (1998) Scientific discovery learning with computer simulations of conceptual domains. *Review of Educational Research* 68:179–201
16. Joolingen WR, Zacharia ZC (2009) Developments in Inquiry Learning. In: Balacheff N, Ludvigsen S, de Jong T, Lazonder A, Barnes S (eds) *Technology-Enhanced Learning*, chap 2, pp 21–37
17. Kelleher C, Pausch R (2005) Stencils-based tutorials: design and evaluation. In: Proceedings of the SIGCHI conference on Human factors in computing systems, ACM, NY, CHI '05, pp 541–550
18. Kirschner P, Sweller J, Clark RE (2006) Why minimal guidance during instruction does not work: An analysis of the failure of constructivist, discovery, problem-based, experiential and inquiry-based teaching. *Educational Psychologist* 41(2):75–86
19. Kynigos C (1992) Insights into Pupils' and Teachers' Activities in Pupil-Controlled Problem-Solving Situations. In: *Information Technology and Mathematics Problem Solving: Research in Contexts of Practice*, NATO ASI Series, vol 2, Springer Verlag, pp 219–238

20. Lesh R, Kelly AE (1996) A constructivist model for redesigning AI tutors in mathematics. In: Laborde JM (ed) *Intelligent learning environments: The case of geometry*, New York:Springer Verlag
21. Mason J (2008) Being Mathematical With & In Front of Learners: Attention, Awareness, and Attitude as sources of Differences between Teacher Educators, Teachers & Learners. In: Wood (ed) *International handbook of mathematics teacher education*, vol 4, Sense Publishers
22. Mavrikis M (2004) Improving the Effectiveness of Interactive Open Learning Environments. In: 3rd Hellenic Conference on Artificial Intelligence - companion volume, pp 260–269
23. Mavrikis M, Gutierrez-Santos S (2010) Not all wizards are from Oz: Iterative design of intelligent learning environments by communication capacity tapering. *Computers & Education* 54(3):641–651
24. Mayer RE (2004) Should There Be a Three-Strikes Rule Against Pure Discovery Learning? - The Case for Guided Methods of Instruction. *AmPsychol* pp 14–19
25. Noss R, Hoyles C (1996) *Windows on mathematical meanings: Learning cultures and computers*. Dordrecht: Kluwer
26. Noss R, Poulouvassilis A, Geraniou E, Gutierrez-Santos S, Hoyles C, Kahn K, Magoulas GD, Mavrikis M (2012) The design of a system to support exploratory learning of algebraic generalisation. *Computers & Education* 59(1):63–81
27. Paramythi A, Weibelzahl S, Masthoff J (2010) Layered evaluation of interactive adaptive systems: framework and formative methods. *User Modeling and User-Adapted Interaction* 20(5):383–453
28. Pontual Falcão T, Price S (2010) Interfering and resolving: How tabletop interaction facilitates co-construction of argumentative knowledge. *International Journal of Computer-Supported Collaborative Learning* pp 1–21
29. Read J, Macfarlane S (2002) Endurability, Engagement and Expectations: Measuring Children's Fun. In: *Interaction Design and Children*, Shaker Publishing, vol 2, pp 1–23
30. Read JC (2008) Validating the Fun Toolkit: an instrument for measuring children's opinions of technology. *Cognition, Technology & Work* 10(2):119–128
31. Rounding K, Tee K, Wu X, Guo C, Tse E (2011) Evaluating Interfaces with Children. In: *Child Computer Interaction. 2nd Workshop on UI Technologies and Educational Pedagogy*. At ACM CHI 2011, Conference on Human Factors in Computing Systems
32. Scardamalia M, Bereiter C (1991) Higher Levels of Agency for Children in Knowledge Building: A Challenge for the Design of New Knowledge Media. *Journal of the Learning Sciences* 1(1):37–68
33. Tse E, Schöning J, Huber J, Marentette T, Beckwith R, Rogers Y, Mühlhäuse M (2011) *Child Computer Interaction. 2nd Workshop on UI Technologies and Educational Pedagogy*. At ACM CHI 2011, Conference on Human Factors in Computing Systems
34. Wood H, Wood D (1999) Help seeking, learning and contingent tutoring. *Computers & Education* 33:153–169
35. Zayas Pérez B, Cox R (2009) Teaching safety precautions in a laboratory DVE: the effects of information location and interactivity. *Computacion y Sistemas*

(special issue on 'Innovative Applications of Artificial Intelligence') 13:96–110

Appendix

Scenario	Feedback strategy description and messages	Feedback Characteristics
<p>A student has made a pattern but has not coloured it at all or the colouring is wrong (e.g. usually the same number of tiles as the number of repetitions). The student continues working ignoring this and therefore the suggestion button is enabled, indicating the existence of feedback. The student clicks help.</p> <p>A student has constructed and coloured a pattern. They unlock the number used in the repetitions of the building block. They think they have accomplished the goal but haven't checked whether their pattern is general.</p> <p>A student has made the pattern and has coloured it correctly for the specific number of repetitions of the building block. For example for 4 repetitions of the building block of the green pattern, they have put 20 green tiles. The student probably fails to see the lack of generality and a suggestion is generated.</p> <p>Continuing from the scenario above, the student does not realise the specificity of their answer. As soon as they provide the correct (albeit specific) number instead of a general expression the system provides a reflection prompt.</p> <p>A student places single tiles on the canvas in an effort to make the given pattern. They have already placed 10 tiles in the correct place and they don't seem to have an intention to use a building block to repeat them.</p> <p>A student has cluttered their canvas with tiles and numbers and struggles to construct a general rule. A clutter detection algorithm identifies possible unwanted elements of the student's construction.</p>	<p>Feedback strategy description and messages</p> <p>The specificity of messages increases progressively from just indicating the problem to providing a suggestion on how to fix. Messages point to the pattern and the number of building block: This pattern is not coloured, "How many tiles make a building block in your pattern? How many tiles does the whole pattern need to be coloured?". "Enter here the number of tiles you need to colour the pattern".</p> <p>The system points to the animation button. "Have you tried animating your pattern. Is it always coloured?"</p> <p>A transparent layer appears on top of eXpresser and only allows the animation button to be clicked.</p> <p>Upon the student's request the system suggests to the student to change manually the number of repetitions of the building block. The student changes it to 6 and the visual feedback (lack of colouring) in combination with the repetitive action (establishing that the number of tiles now is 30) provides the student with an opportunity to start recognising variants and invariants.</p> <p>This is one of the few occasions that the system provides an intervention without an explicit or implicit request. The prompt point to the corresponding property list and reads 'How did you calculate the correct number of tiles here'.</p> <p>The system has detected a rhythm in the way the student places tiles and prompts them to focus/find what they repeat and use this to make a pattern.</p> <p>Upon student's failed attempt to construct a general rule, the system points and highlights the detected elements suggesting to the student to consider whether they are necessary.</p>	<p>explicitly requested, passive, no change, co-located text</p> <p>implicitly requested, active, no change, text</p> <p>low, active, no change, co-located text</p> <p>interruption, passive, no change, co-located text</p> <p>low, passive, no change, text + image</p> <p>interruption, passive, view, co-located text</p>

Table 7 Example scenarios, corresponding feedback strategies and feedback characteristics (as per Section 3.2) for providing support in eXpresser.