A Separation of Concerns for Engineering Intelligent Support for Exploratory Learning Environments

Sergio Gutierrez-Santos, Manolis Mavrikis and George D. Magoulas

London Knowledge Lab
23–29 Emerald St
WC1N 3QS, London
Email: {sergut,gmagoulas}@dcs.bbk.ac.uk, m.mavrikis@lkl.ac.uk

Exploratory learning environments (ELEs) can have a positive effect on learning as long as they have proper support. However, the difficulty of supporting exploration and unstructured interaction means that there are few systems with intelligent computer-based support. The paper presents a divide-and-conquer strategy to approach the difficult tasks of development and evaluation of intelligent support in ELEs. The strategy considers three parts, each one focusing on the three most important questions related to support at any given moment: (i) what is the situation now? (evidence), (ii) which aspect needs support? (reasoning), and (iii) how should the support be presented for maximum efficacy? (presentation). The benefits of this approach include easier testing and validation, more reliable progress, and better communication and management of the members of the interdisciplinary team required to design and evaluate the system.

Keywords: *intelligent support, adaptive feedback, exploratory learning environment, microworld.*

ACM Classifications: *I.2.1, I.2.11, J.4, K.3*

1. Introduction

Exploratory Learning Environments (ELEs) and microworlds (μWs) in particular, have been shown to offer strong benefits to the learning process (Joolingen and Zacharia, 2009; Noss and Hoyles, 1996). However, their exploratory nature can be both a blessing and a curse (e.g. distraction, loss of focus) and research in the learning sciences (e.g. Mayer, 2004; Kirschner *et al*, 2006; Klahr and Nigam, 2004) suggests that freedom of exploration without a proper degree of support is problematic. Although ELEs in general are designed to react to students' actions, they usually provide only 'integrated' feedback. This integrated feedback is usually purely reactive. It is not adaptive nor supportive as it does not rely on an analysis of the situation and the student. This is the reason why most of the times students' exploration must be supported by humans who can structure the activity and provide individualised support. An illustrative example is a flight simulator: flight students can explore the world and the reactions of the plane, but an instructor must observe the students and provide feedback for the students to improve their skills. In the classroom, unfortunately, it is not possible to achieve this level of personalisation due to the intense pressure of supporting a high number of students at the same time.

This introduces a clear need for computer-based intelligent support in this kind of system, one that can take into account students' characteristics, individual differences, and their particular

Manuscript received: 1 January 2012
Communicating Editor: José-Luis Sierra-Rodriguez

interaction history with the environment. In other words: if microworlds and other exploratory learning environments are to be used widely in classrooms, computer-based intelligent support must be provided. This, however, is a very challenging task. There are two main problems: first, the interaction of learners is highly unstructured and difficult to model and prescribe (e.g. more than one "good" approach lead to a "correct" solution, and "wrong" approaches very often lead to good learning). Second, the expertise required to develop an intelligent environment is fragmented among several experts from very different disciplines (e.g. teachers, programmers, educational psychologists, etc). This fragmentation makes collaboration among these experts difficult due to their different backgrounds, expertise, and even vocabularies. As a result, there are few ELEs with explicit intelligent computer-based support, and there is a lack of a general methodology to approach the problem.

In this paper, we present an approach to implement intelligent support in ELEs which is driven by the following challenges: (i) the need to break down into tractable problems the challenge of monitoring and reacting to students' complex interactions in such an open environment, with an eye on scalability; (ii) the high cost of communication between several kinds of experts (e.g. computer and learning scientists) which are required to tackle this problem; and (iii) the difficulty of gradually validating a complex intelligent system that is developed in the context of an interdisciplinary team.

To convey our approach, the paper is structured as follows. Section 2 provides the theoretical background for the paper, clarifying the terminology and illustrating the challenge of designing and evaluating intelligent support for ELE. Section 3 presents our approach, using project MiGen as a case study. Section 4 discusses the main advantages of the approach while Section 5 closes the paper and hints lines of future work.

2. Background

2.1 Microworlds and other ELEs

ELEs are virtual environments that provide learners with opportunities to engage with a domain and explore a range of possibilities that would be otherwise difficult to experience directly. Such environments vary on the amount of exploration they afford and how they contextualise learning by means of tasks.

Simulators offer a (usually simplified) representation of the domain model, which allows inquiry-based exploring "what if" scenarios (Swaak, 1998). Learners can vary the parameters of the simulation or some of its elements and observe the effect of the changes. ACE (Bunt and Conati, 2003) is an example of a calculus simulator where students can modify the parameters of different functions; while TELEOS – described in Chieu *et al* (2010) – is an example of a medical simulator where the learner interacts with a virtual hip bone.

Microworlds also represent the domain model, but allow the learner to add or remove elements to it, as well as modifying its attributes. Examples include eXpresser, Scratch, and Cabri (see Figure 1); the first one is used later in the paper to illustrate our design and evaluation methodology (Section 2.3). It is obvious that the level of exploration granted to students is much higher in the case of microworlds and makes the problem of engineering intelligent support for this kind of ELE more challenging; this is why the approach presented in this paper takes the case of a microworld as a case study. Microworlds have been used in geometry and other topics in mathematics education (Noss and Hoyles, 1996) and inquiry learning (Joolingen and

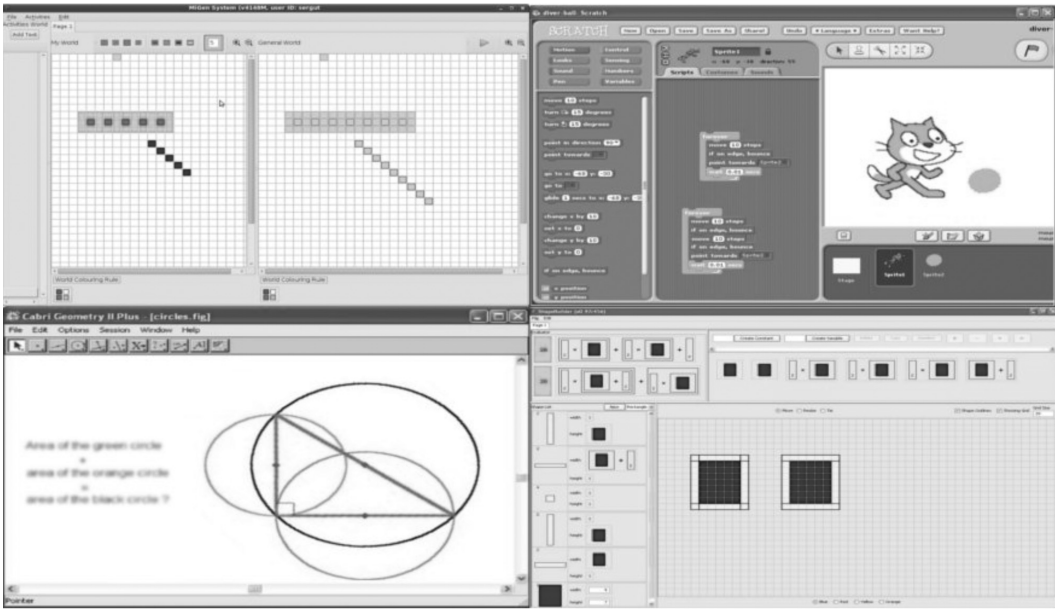


Figure 1: Four sample microworlds. eXpresser (top-left) allows students to create patterns and relate them with numbers (Noss *et al*, 2012). In Scratch (top-right), students create "programs" connecting boxes representing instructions, in order to produce a multimedia presentation (Maloney *et al*, 2008). The currency in the case of Cabri(bottom-left) are points, lines, figures, and other geometrical entities that can be combined and related to prove theorems (Balacheff and Sutherland, 1993). Lastly, in Shape-Builder (bottom-right) students create shapes, and they can relate their dimensions (e.g. make one shape as tall as another is wide) by using icon-variables (Pearce *et al*, 2008). Cabri and Scratch are currently supported by human facilitators. Computer-based support for eXpresser and ShapeBuilder was developed in the MiGen project.

Zacharia, 2009; de Jong, 2010). Figure 1 shows snapshots of four different microworlds for the sake of clarity. Although a static snapshot fails to capture the dynamic interaction with such a learning environment (let alone the provision of support), we have tried to give a glimpse by portraying different microworlds from different domains, like algebra, geometry, and programming.

2.2 Design and Evaluation of Intelligent Support in Educational Environments

Developing intelligent support in educational environments traditionally requires monitoring students' responses to particular questions or problem solving tasks, reason about them, and respond dynamically by providing feedback that facilitates the learning process. Such abilities allows personalization for each student, a goal shared in general by user-adaptive systems (Jameson, 2008). Several architectures have been proposed to deal with the requirements of such systems. The most relevant research is in the field of intelligent tutoring systems and adaptive hypermedia where intelligent systems are referred to as knowledge-centred (Woolf, 2008) or concept-based systems (Houben *et al*, 2005) respectively, since typically the domain is organised in terms of concepts that represent the content provided by the system or model the student knowledge.

In relation to feedback, research approaches are mainly focusing on the gradual provision of support and/or adaptation of feedback according to learners individual characteristics with the aim of providing guidance, tutoring or reflection (see a review in Gouli *et al*, 2006). Most intelligent learning environments that aim to provide personalised feedback accommodate mainly the learners knowledge level and typically rely on its representation in a user model (or in the case of educational applications, a student or learner model). Depending on the application most systems require an adaptation model (Houben *et al*, 2005) that requires the encoding of 'tutoring' or 'communication' knowledge to represent teaching strategies and communication methods (Woolf, 2008).

Regardless of the exact approach and components of a concept-based system, the relatively low level of interactivity facilitates the design and development of separate components and eases their integration. However, in the case of μWs and other ELEs, the unstructured interaction and the ill-defined nature of the knowledge that is supported by their design renders the integration of the various components of the system much more challenging. As it is argued in Mavrikis and Gutierrez-Santos (2010), providing intelligent support for a μW requires attention to its epistemology, i.e. how it provides scope for knowledge, or limits it; in other words, how it transforms the nature of the domain it represents (diSessa, 1995). For example, a midpoint in dynamic geometry microworlds, such as Cabri (Balacheff and Sutherland, 1993), is not necessarily the same as an Euclidean midpoint; gauges in science labs provide an operationalisation of concepts such as speed or voltage. This transformation of the domain requires identifying feedback strategies and interventions situated in the μW. This makes the typical knowledge elicitation process even more challenging than it usually is. For example, in the case of other intelligent learning environments, support is usually centred around domain concepts, prerequisites and the sequencing of educational material, e.g. (Papasalouros *et al*, 2004; Woolf, 2008). In contrast, in the case of μWs one needs to investigate the very nature of the interaction and how it can be supported in order to help students achieve the desired learning objectives.

The proposal presented here has been influenced by other divide-and-conquer approaches like the subsumption architecture used to develop AI for robots (Brooks, 1991), where complicated intelligent behaviour is divided into layers of simpler behaviour modules. The problem of robotic autonomous behaviour presents several similarities to that of intelligent support in microworlds: unstructured input data, difficulty of representation, and the requirements that emerge from students' real-time interaction. The main difference is our focus on facilitating the communication between learning and computer scientists which is crucial when engineering intelligent support in any learning application but especially in ELE.

In the field of ITS, there have been some initiatives that bear similarities to ours. For example Webber *et al* (2002) makes an effort to separate evidence collection from the decision on feedback, but there is no explicit reasoning or feedback adaptation, only a voting mechanism to coordinate agents. Another relevant work is Duquesnoy *et al* (2002), that separates the intelligent components that decides what feedback to provide and those that adapt the presentation of the feedback (defined by a barcode). Our approach can be seen as taking the best of each approach.

One benefit of our approach is that our separation of concerns facilitates testing. Several works have proposed other divide-and-conquer approaches in the field of adaptive hypermedia precisely to facilitate testing, (Tintarev and Masthoff, 2009; Weibelzahl and Weber, 2003; Paramythis and Weibelzahl, 2005). The particular case of microworlds, with their unstructured interaction and their complex relationship to learning, makes the problem even more challenging, and prevents automated test-driven approaches (Linillos *et al*, 2008).

2.3 The Case of eXpresser

In this paper we present our approach using examples from the interdisciplinary MiGen project, which set out to build a pedagogical and technical environment to help 11-14 year-old students' develop algebraic ways of thinking (Noss *et al*, 2012). The project is centred around the micro-world eXpresser (see Figure 2). Several components of the overal system take care of different aspects of the learning process, including collaboration activities, task sequencing, and intelligent support. A detailed description of the architecture of the MiGen system can be found in Pearce and Poulovassilis (2009) but below we provide a brief description of eXpresser to help the reader appreciate the examples in the next section.

The eXpresser microworld and the associated activities encourage students to construct animated models comprising patterns of repeated building blocks of tiles. Underlying this surface goal, the main objective is to promote students development of algebraic and other mathematical ways of thinking. In particular, students are required to find general algebraic expressions underpinning their models and in order to represent the generalities they perceive they can use special numbers that act like variables.

The microworld gives a lot of freedom to students to construct their patterns in a multitude of different but equivalent ways (for a detailed description of the eXpresser the interested reader is referred to Noss *et al* (2012). Students are given tasks in which they construct patterns and find expressions, but they can follow very different paths that lead to the same end: e.g. different combinations of parameters produce the same sub-pattern, and different sub-patterns can be

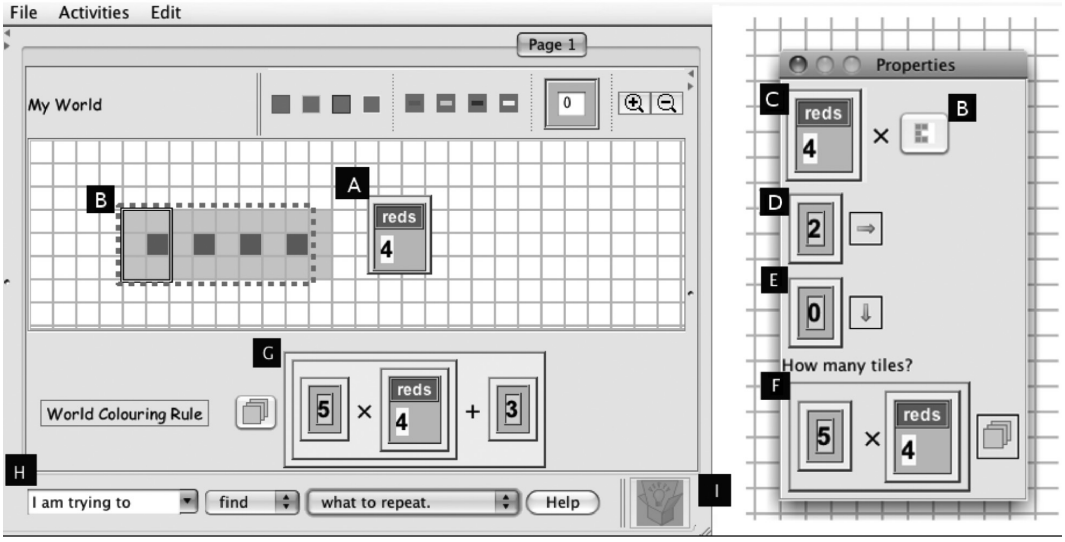


Figure 2: Constructing a pattern in eXpresser and describing it with a rule. Main features: (A) An 'unlocked' number that acts like a variable is given the name 'reds'and signifies the number of red (dark grey) tiles in the pattern. (B) Building block to be repeated to make a pattern. (C) Number of repetitions (i.e. the value of thevariable 'reds'). (D,E) Number of grid squares to translate B to the right and downafter each repetition. (F) Local rule: units of colour needed. (G) General rule: gives the total number of units of colour required to paint the whole pattern (if correct, solves the task). (H) Help-seeking area with drop down menus and (I) suggestion button for feedback provision.

combined to result in the same pattern. The need to understand all possible ways of approaching a task, and to apprehend the grade of correctness of each approach in order to provide intelligent support, result in a very complex endeavour.

3. Dividing and Conquering Support for ELE

This section presents the approach we have followed to compartimentalise the complex needs of the artificial intelligence aspects required to provide computer-based support on a microworld. This compartimentalisation of different concerns makes the overall problem easier to solve.

We consider the problem of development and evaluation of intelligent support in ELE as divided in three parts, each one focusing on the three most important questions related to support at any given moment: (i) what is the situation now? (evidence), (ii) which aspect needs support? (reasoning), and (iii) how should the support be presented for maximum efficacy? (presentation). Some of these questions depend logically on one another, i.e. we cannot answer how to present support and feedback on one aspect of the interaction if we do not know what needs to be supported, and we will not know that until the current situation is appraised. This compartimentalisation of the problem of support is depicted in Figure 3, and is orthogonal and complementary to the separation between domain model and learner model.

The three distinct aspects of the general problem of supporting exploratory learning can thus be attacked separately. Each of them requires different kinds of expertise; by separating clearly the scopes of different experts and defining clear communication interfaces between them, this approach facilitates the interaction in interdisciplinary teams such as those needed for the addition of intelligent support to an exploratory learning environment like a microworld or simulator. This separation of concerns also facilitates the testing and validation of different components at different levels by decoupling them from one another.

It is important to note at this point that the development of support starts taking place in the opposite direction of the flow of data. The first stage is to ascertain the kind of support that is needed for the learning environment, a work led by educators and teachers, although HCI experts may play an important role. The next step is finding regularities and similarities in the different instances of support and finding rules that can describe when to activate them, a work led by computer scientists and AI experts. The final step is finding the evidence needed by the reasoning machine to generate feedback strategies in the flow of information obtained from the

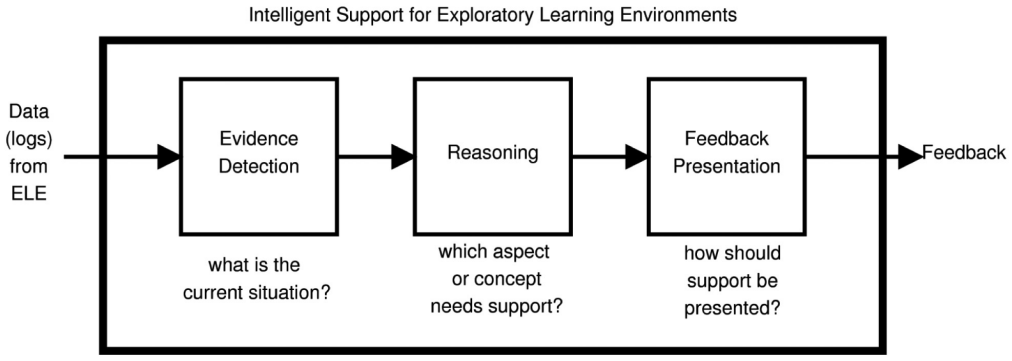


Figure 3: Conceptual data flow of support for exploratory learning.

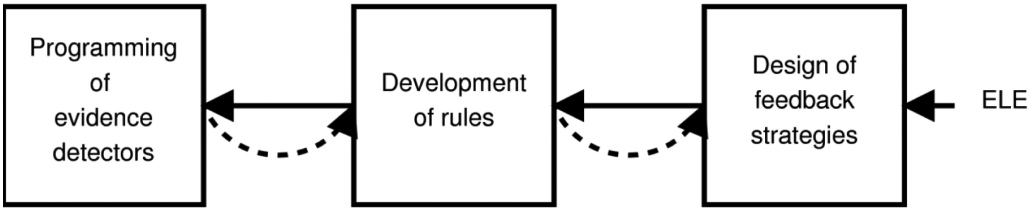


Figure 4: The first step is finding out which feedback strategies are needed, then how or when to fire them, and then looking for the evidence needed for this.

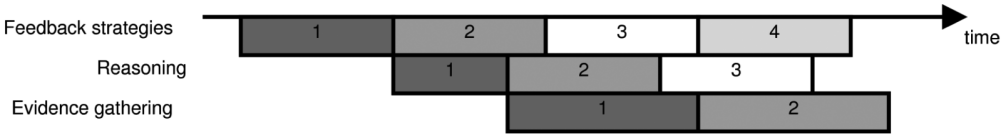


Figure 5: Design and development of feedback strategies, reasoning, and evidence collection are reponsibility of different researchers and can take place in parallel.

environment, a work for experts in machine learning. This is depicted in Figure 4. It is important to note that this is not a cascade model. First, because work at the three levels is done in parallel by different experts, e.g. while educators are still analysing the interaction of teachers and students for finding additional feedback strategies, AI experts are already designing the techniques that will generate already understood feedback strategies (a simplified depiction is shown in Figure 5). Sometimes work on evidence collection can uncover additional opportunities for reasoning and then feedback presentation; this is represented by the dotted lines in Figure 4. The three problems are described now in more detail, in this order.

3.1 Feedback strategies

Design: The first step is studying the support needed by students when interacting with the exploratory learning environment. This is a long and iterative process of requirement elicitation in which a learning ecosystem where all support is provided by a human is transformed into one in which most of the support is provided by the computer. We have described elsewhere (Mavrikis and Gutierrez-Santos, 2010) a methodology to undertake this transformation to which we refer to as 'Iterative Communication Capacity Tapering' (ICCT). In brief, ICCT recognises that in-depth understanding of user behaviour requires observing and analysing situations in their actual context. On the one hand, the communication bandwidth is gradually reduced: starting with a teacher interacting normally face to face with students, the situation evolves to remote interaction where teacher and student are physically separated (also known as 'wizard-of-Oz' studies). On the other hand, the capacity for improvisation is also reduced: at first the teacher can support the student with any words or actions at will, but gradually her options are limited until there is a script that describes when support is provided. For the sake of space, we do not go into details here but we refer interested readers to Mavrikis and Gutierrez-Santos (2010).

The rest of this section explains how the other layers are developed and integrated to obtain a full computer-based intelligent support system.

Evaluation: Feeback strategies are evaluated in context, and their identification and recording allows their evaluation from additional experts (e.g. teachers in focus groups), while various system components are being developed. As the possible feedback strategies are implemented, usability and pedagogy experts can evaluate the presentation of the feedback and other pragmatic aspects. Further wizard-of-Oz sessions can help evaluate the layer with students who are interacting with the system. The separation of concerns allows this process to be undertaken independently from the development of the other technical components of the system. When the other components become available and the whole system is put together, a more extended form of summative evaluation can take place (Mavrikis *et al*, in press).

The case of MiGen: At early stages, the ICCT process (Mavrikis and Gutierrez-Santos, 2010) allowed the identification of several pedagogic strategies through empirical studies with students and teachers. After the initial design, and the data collection, experts played the most important role in the formative evaluation of this layer. In particular, scenarios were presented to a 'teacher group' associated with the project, who commented on their initial reaction during a brainstorming session following well-documented approaches such as the ones presented in Kelley and Littman (2001). Scenarios were elaborated and the teacher group was requested to comment on the feedback. That was in the form of storyboards that provided experts with details about the context of the interaction and the actual feedback that the system would provide. Meetings with the teacher group were interleaved with wizard-of-Oz studies with students, where the requirements for intelligent feedback were gradually clarified, both in terms of the need for feedback and the graphical elements to be implemented and used.

The result of this process was a series of feedback strategies and requirements on how to implement them at the UI level. One very illustrative requirement elicited was the need to focus students' attention to portions of the screen by providing messages colocated with objects of the microworld. What teachers were found to do in face-to-face interactions (and was greatly needed during the wizard-of-Oz sessions) was implemented through a series of graphical widgets. Figure 6 shows an example. More examples of the UI and evaluation methodology in MiGen are provided in Mavrikis *et al* (in press).

3.2 Reasoning Layer

Design: The analysis of the interaction of students with the learning environment and the feedback interventions undertaken by teachers leads to the identification of high-level regularities. We call these repeated patterns feedback strategies. The same feedback strategy can be instantiated differently depending on student characteristics and the exact context. A reasoning component must be responsible for deciding when a specific feedback strategy must be followed – and, if many, prioritised properly. In the iterative process that drives the transition from a human-supported to a computer-supported system, these feedback strategies are progressively clarified and used as a script to guide the provision of feedback as the communication capacity of the system is reduced (Mavrikis and Gutierrez-Santos, 2010).

A requirements analysis of the feedback strategies shows what information is needed from the microworld (i.e. evidence). This also clarifies how a user model can be updated based on current data. Different techniques can be used to merge data from heterogeneous sources including rule-based systems (Hatzilygeroudis and Prentzas, 2004), Bayesian networks (Charniak, 1991), and case-based reasoning (Kolodner, 1993), to mention but a few.

Evaluation: Evaluation of the reasoning involves two different processes. First, the results of the reasoning mechanism have to be validated against the knowledge of experts (i.e. gold standard validation (Rusell and Norvig, 2003)). This involves presenting experts with the evidence (as calculated by the analysis modules, see below) in several scenarios, and comparing their recommendation of a feedback strategy with those proposed by the system. Additionally, it is important to check that the reasoning mechanism is robust. Depending on the reasoning mechanism used, the influence of the effect of changes in the evidence collected can be ascertained by following a form of sensitivity analysis (e.g. if Bayesian reasoning is used). This evaluation process can also demonstrate whether the reasoning lacks some pedagogical expertise.

The case of MiGen: In MiGen the reasoning layer is implemented using a knowledge-based approach, as it fits our requisites and it is easier to maintain and scrutinise, at this stage. The rules were developed in collaboration with pedagogical researchers within the team and specify situations where intelligent feedback is needed to help the student, bearing similarities to approaches based on constraints, c.f. (Mitrovic *et al*, 2002).

We have identified many strategies in MiGen, and will illustrate the concept focusing on a strategy we call "messing-up". This feedback strategy refers to a situation where a model in eXpresser is not coloured in a general way. The strategy is a way of provoking cognitive conflict and its use by teacher has been documented in the field of mathematics education (Healy *et al*, 1994). To automate the strategy and provide computer-based support requires (a) understanding whether a student has constructed a possible solution, (b) whether it is very similar (maybe equivalent) to one of the expected solutions to the task, (c) whether they have made an attempt to make it general and (d) whether the pattern is coloured in a general way. These are the pieces of evidence needed to "fire" this feedback strategy (see next Section). Once the strategy is selected, the feedback components developed by the pedagogical team decide on how to present it to students, e.g. selecting the right wording.

Evaluation of the reasoning components of MiGen had two main concerns: adequateness and completeness. First, and most important, we wanted to make sure that the reasoner was selecting the right feedback strategies given some evidence. In order to do this, we transformed the knowledge base into natural language descriptions (similar to the former paragraph) and showed them to the pedagogical experts to evaluate the appropriateness of a feedback strategy given some evidence. These descriptions were produced automatically (by XSLT tranformations) to prevent errors and misanderstandings due to human interpretation in the translation process.

In addition, the completeness of the rule base had to be evaluated. It would be virtually impossible to show all the rules to experts (and even so, they would not be able to assess whether a rule is missing). To alleviate this situation, experts were also asked to provide us with as many scenarios as they could (up to five) where they thought that feedback should be provided based on their experience. We then simulated their scenario and tested whether our rules captured it and whether the same feedback strategy was proposed by the system and the expert. This iterative process has revealed several missing rules.

3.3 Evidence Detection (Analysis Layer)

Design: As explained in the former section, the study of the feedback strategies reveals what information is needed. Information from the microworld is usually not provided in any appropriate form, but rather like low-level logging data at best. Producing high-level constructs –

that we call evidence – from the raw data generated by the microworld is done by many different computational modules, each of them pertaining to a particular and well-defined analysis problem. Each module produces as output one piece of evidence on which to perform the reasoning, each module filters and analyses the raw stream of data from the learning environment as appropriate.

Orthogonal to the scope separation feedback–reasoning–evidence, the clear separation of responsibilities among the analysis modules allows their development and validation to run in parallel. This in turn facilitates the work of several programmers in a scalable way without compromising the vision of the whole system.

Evaluation: Depending on the nature of the analysis modules, different evaluation techniques are required. The modules tackle specific problems, so in a first stage they can be tested using a combination of white-box and black-box techniques. In some cases, the modules will need to be fine-tuned according to the data by an iterative data-driven evaluation (i.e. cycles of training, validation, and improvement). Lastly, those modules that measure some subjective aspect need to be evaluated through a gold-standard process, where several experts are given the same inputs as the module and their answers are compared.

The case of MiGen: Some of the computational analysis modules that are used in MiGen are summarised in Table 1. Each module concentrates on one aspect of the interaction student–microworld. We have chosen a small selection that are illustrative of the needs of the system. MiGen uses more than thirty analysis modules to generate evidence about students' actions in eXpresser.

Some of the simpler modules attack a problem where no subjectivity is involved. In these cases, the validation was a purely technical process that was performed with a combination of unit testing, black-box testing, and a library of test cases. On the other hand, there are several modules that study aspects of the learners' interaction that involve a certain level of subjectivity. For example, different teachers may differ on what constitutes "rhythm" in the actions of the students, or how similar the student's construction is to one of the typical mistakes in the database. In these cases, the computational analysis modules were tested by presenting several scenarios to experts, asking them to choose a diagnosis from different options. The module outputs were then compared against a gold standard defined as the most voted option. During several iterations, the results of the computational component were compared with those of the

Name	Function
Apparent Solution Verifier	Returns true if the construction on the screen has the same appearance as a solution.
Construction Evaluator†	Given a set of expected solutions and common mistakes, returns a similarity measurement to each of them
Rhythm Detector	Detects rhythmic repetitions when students make patterns; this can be a symptom of implicit structures in the students' minds (Noss *et al*, 2012)

Table 1: A selection of computational analysis modules in MiGen (more than thirty). The module marked with † was originally developed to support students learning witha different microworld called ShapeBuilder (Cocea and Magoulas, 2008).

Function	Wizard-of-Oz	Gold-standard	Unit-testing
Feedback generation	X	X	
Reasoning		X	X
Evidence detection		X	X

Table 2: Decoupled testing and validation of different components of intelligent support. If there are several modules (e.g. evidence detection) all are independently evaluated.

experts, resulting in fine changes in the component, e.g. modification of weights in equations. After the last iteration, the decisions of the component were equivalent to those of the experts in average (individual experts agreed with the gold standard in between 70% and 80% of scenarios).

4. Discussion

The separation of responsibilities to layers helps the design and the development of the intelligent components, because it divides the complex and ill-defined problem of support for exploratory learning into smaller manageable well-defined subproblems (divide and conquer). The main benefits of this approach are:

Better management of interdisciplinary teams: The scopes of expertise and the communication interfaces within the research team are clearly set, which results in our experience in more focused work and more efficient communication in heterogeneous interdisciplinary TEL-research groups like the MiGen team.

Scalability: The approach encourages modular development: feedback strategies are independent from each other, and so are evidence detectors. This facilitates gradual improvement and formative evaluation of intelligent support.

Reusability: Computational analysis modules tackle specific and well-defined problems, interacting with the rest of the system through clearly defined interfaces; this has allowed the reuse of analysis modules for different microworlds (e.g. Shape-Builder and eXpresser).

Easier testing and validation: Clear separation of concerns results in lower coupling between elements and layers of computation, facilitating independent and parallel testing and validation. This is summarised in Table 2.

Early detection of errors: Related to the former point: a separated, focused, and early verification of the parts of the system results in earlier detection of problems, at a stage where the system components can still be modified and tuned.

More reliable progress: The more an intelligent system grows, the more difficult it can become to manage and scale. Clear separation and lower coupling, together with independent testing and evaluation, result in more reliable progress of the support system as new feedback strategies are identified and implemented into the system.

5. Conclusions and Future Work

Exploratory Learning Environments (ELEs) have the potential of enabling solid learning due to their exploratory nature, given adequate support. However, their openness is both a blessing and a curse: designing, implementing, and validating computer-based intelligent support for ELEs is

a daunting task. This is why support is almost always provided by human facilitators, but this solution does not scale well if such learning applications are to be used at the classroom level (let alone distance-learning scenarios). Finding a way to simplify the creation of intelligent support, and thus enable microworlds and other ELEs to be used in classrooms, has been one of our main motivation for the work reported here.

This paper proposes a division of the problem of providing computer-based support for an exploratory learning envioronment into three simpler subproblems: identification of feedback strategies, collection of evidence, and reasoning from the evidence to deduce when to follow a particular feedback strategy. The main advantages of following this approach in the MiGen project have been easier design of the different components, easier testing and validation of the different parts of the system, and improved collaboration between the varied members of our interdisciplinary team once the separation of concerns became clearer mid-project.

Our future work goes along the lines of improvement and extension. On the one hand, we plan to create mechanisms for future users of the MiGen system to improve (e.g. teachers generating new feedback strategies based on their experience). On the other hand, we plan to use this approach for supporting different simulators and microworlds in the future. Current ongoing development in the EU-funded Metafora project (www.metafora-project.org), where students are interacting with a computer-supported collaborative learning (CSCL) application that integrates several exploratory environments, relies on the same approach for the design and delivery of the feedback strategies. In the future, we envision employing the approach in order to provide support in domains that strongly benefit from exploratory learning like programming (e.g. on top of microworlds like Scratch or learning tools like Greenfoot). Our long-term aim is to facilitate the development of intelligent exploratory learning environments, thus encouraging their widespread use both in classrooms but also by individuals in 'anytime, anywhere' learning.

Acknowledgements

This work is funded in part from the MiGen project, grant TLRP RES-139-25-0381. Thanks to all our MiGen colleagues for their support and ideas.

References

BALACHEFF, N. and SUTHERLAND, R. (1993): Epistemological domain of validity of microworlds: The case of logo and cabri-gomtre. In *Proceedings of the IFIP TC3/WG3.3 Working Conference on Lessons from Learning.*

BROOKS, R.A. (1991): Intelligence without representation. In *Artificial Intelligence*. 47: 139–159.

BUNT, A. and CONATI, C. (2003): Probabilistic student modeling to improve exploratory behaviour. *User Modeling and User-Adapted Interaction* 13: 269–309.

CHARNIAK, E. (1991): Bayesian networks without tears. *AI Magazine* 12(4): 50–63.

CHIEU, V.M., LUENGO, V., VADCARD, L. and TONETTI, J. (2010): Student modeling in orthopedic surgery training: Exploiting symbiosis between temporal bayesian networks and fine-grained didactic analysis. *International Journal of Artificial Intelligence in Education* 20 (3).

COCEA, M. and MAGOULAS, G. (2008): Combining intelligent methods for learner modeling in exploratory learning environments. In *Proceedings of the International Workshop on Combinations of Intelligent Methods and Applications at ECAI-08*. 13–18.

DE JONG, T. (2010): Instruction based on computer simulations. In MAYER, R. and ALEXANDER, P. (Eds.), *Handbook of research on learning and instruction*. Routledge Press, 446–466.

DISESSA, A.A. (1995): Epistemology and systems design. In DISESSA, A., HOYLES, C., NOSS, R. and EDWARDS, L. (Eds.), *Computers and Exploratory Learning*. Springer, 15–29.

DUQUESNOY, L., BERGER, J-L., PREVOT, P. and SANDOZ-GUERMOND, F., (2002): SIMPA: A training platform in work station including computing tutors. In *ITS*. 507 – 520.

GOULI, E., GOGOULOU, A., PAPANIKOLAOU, K.A. and GRIGORIADOU, M. (2006): An adaptive feedback framework to support reflection, guiding and tutoring. In MAGOULAS, G.D. and CHEN, S.Y. (Eds.), *Advances in Web-Based Education: Personalized Learning Environments*. 178–202.

HATZILYGEROUDIS, I. and PRENTZAS, J. (2004): Using a hybrid rule-based approach in developing an intelligent tutoring system with knowledge acquisition and update capabilities. *Expert Systems with Applications* (26): 477–492.

HEALY, L., HOELZL, R., HOYLES, C. and NOSS, R. (1994): Messing up. *Micromath* 10: 14–17.

HOUBEN, G.J., AROYO, L., DICHEVA, D. and DE BRA, P. (2005): Adaptation engineering in adaptive concept-based systems. In CHEN, S.Y. and MAGOULAS, G. (Eds.), *Adaptable and Adaptive Hypermedia Systems*, 36–58.

JAMESON, A. (2008): Adaptive interfaces and agents. In SEARS, A. and JACKO, J.A. (Eds.), *The human-computer interaction handbook: Fundamentals, evolving technologies and emerging applications* (2nd ed.). 433–458.

JOOLINGEN, W.R. and ZACHARIA, Z.C. (2009): Developments in inquiry learning. In BALACHEFF, N., LUDVIGSEN, S., DE JONG, T., LAZONDER, A. and BARNES, S. (Eds.), *Technology-Enhanced Learning*. 2: 21–37.

KELLEY, T. and LITTMAN, J. (2001): The art ofinnovation: Lessons in creativity from IDEO, America's Leading Design Firm. Doubleday.

KIRSCHNER, P., SWELLER, J. and CLARK, R.E. (2006): Why minimal guidance during instruction does not work: An analysis of the failure of constructivist, discovery, problem-based, experiential and inquiry-based teaching. *Educational Psychologist,* 41(2): 75–86.

KLAHR, D. and NIGAM, M. (2004): The equivalence of learning paths in early science instruction – Effects of direct instruction and discovery learning. *Psychological Science*, 661–667.

KOLODNER, J. (1993): *Case-Based Reasoning*. Kluwer.

LINILLOS, P.P., GUTIERREZ-SANTOS, S., PARDO, A. and DELGADO KLOOS, C. (2008): Guaranteeing the correctness of an adaptive tutoring system. In NEJDL, W., KAY, J., PU, P. and HERDER, E. (Eds.), Adaptive hypermedia and adaptive web-based systems, 5th International Conference, AH 2008, *Lecture Notes in Computer Science*. Springer, 5149: 329–332.

MALONEY, J.H., PEPPLER, K., KAFAI, Y., RESNICK, M. and RUSK, N. (2008): Proceedings of the 39th SIGCSE technical symposium on computer science education. In *Proc. of the 39th SIGCSE technical symposium on Computer science education*.

MAVRIKIS, M. and GUTIERREZ-SANTOS, S. (2010): Not all wizards are from Oz: Iterative design of intelligent learning environments by communication capacity tapering. *Computers and Education* 54(3): 641–651.

MAVRIKIS, M., GUTIERREZ-SANTOS, S., GERANIOU, E. and RICHARD, N., (in press, available at http://dx.doi.org/10.1007/s00779-012-0524-3). Design requirements, student perception indicators and validation metrics for intelligent exploratory learning environments. *Journal of Personal and Ubiquitous Computing*.

MAYER, R.E. (2004): Should there be a three-strikes rule against pure discovery learning? – The case for guided methods of instruction. *American Psychologist*, 14–19.

MITROVIC, A., MARTIN, B. and MAYO, M. (2002): Using evaluation to shape its design: Results and experiences with SQL-tutor. *User Modeling and User-Adapted Interaction* 12(2–3): 243–279.

NOSS, R. and HOYLES, C. (1996): Windows on mathematical meanings: Learning cultures and computers. Dordrecht: Kluwer.

NOSS, R., POULOVASSILIS, A., GERANIOU, E., GUTIERREZ-SANTOS, S., HOYLES, C., KAHN, K., MAGOULAS, G.D. and MAVRIKIS, M., Aug. (2012): The design of a system to support exploratory learning of algebraic generalisation. *Computers & Education* 59(1): 63–81.

PAPASALOUROS, A., RETALIS, S. and PAPASPYROU, N. (2004): Semantic description of educational adaptive hypermedia based on a conceptual model. *Educational Technology and Society* 7(4): 129–142.

PARAMYTHIS, A. and WEIBELZAHL, S. (2005): A decomposition model for the layered evaluation of interactive adaptive systems. In ARDISSONO, L., BRNA, P. and MITROVIC, A. (Eds.), *10th International Conference on User Modeling (UM2005)*. Springer Verlag, 438–442.

PEARCE, D., MAVRIKIS, M., GERANIOU, E. and GUTIERREZ-SANTOS, S. (2008): Issues in the design of an environment to support the learning of mathematical generalisation. In *EC-TEL '08: Proceedings of the 3rd European conference on Technology Enhanced Learning*. Springer-Verlag, Berlin, Heidelberg, 326–337.

PEARCE, D. and POULOVASSILIS, A. (2009): The conceptual and architectural design of a system supporting exploratory learning of mathematics generalisation. In CRESS, U., DIMITROVA, V. and SPECHT, M. (Eds.), *Learning in the Synergy of Multiple Disciplines*. Springer Berlin Heidelberg, Berlin, Heidelberg, 5794(5): 22–36.

RUSELL, S. and NORVIG, P. (2003): Artificial intelligent: A modern approach. Prentice Hall.

SWAAK, J. (1998): What-if: Discovery simulations and assessment of intuitive knowledge. Ph.D. thesis, University of Twente.

TINTAREV, N. and MASTHOFF, J. (2009): Evaluating recommender explanations: Problems experienced and lessons learned for the evaluation of adaptive systems. In *6th Workshop on User-Centred Design and Evaluation of Adaptive Systems*, UMAP.

WEBBER, C., PESTY, S. and BALACHEFF, N. (2002): A multi-agent and emergent approach to learner modelling. In *Proceedings of the 8th Iberoamerican Conference on Artificial Intelligence*.

WEIBELZAHL, S. and WEBER, G. (2003): Evaluating the inference mechanism of adaptive learning systems. In BRUSILOVSKY, P., CORBETT, A. and DE ROSIS, F. (Eds.), *User Modeling. Proceedings of the Ninth International Conference, UM2003*. Springer, Berlin, 154–168.

WOOLF, B.P. (2008): Building intelligent interactive tutors, student-centered strategies for revolutionizing e-learning. Elsevier & Morgan Kaufmann.

Biographical Notes

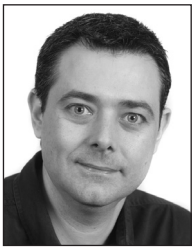
Sergio Gutierrez-Santos *received a telecommunications engineering degree (2002) and a PhD in computer science (2007) from University Carlos III of Madrid. He has worked at the London Knowledge Lab (Birkbeck College) ever since. His research interests centre on artificial intelligence (especially emergent technologies) and its application to problems in teaching and learning, in particular in relation to exploratory learning and creative thinking.*

Sergio Gutierrez-Santos

Manolis Mavrikis *is a research fellow in the London Knowledge Lab (Institute of Education). He holds a PhD and an MSc from the University of Edinburgh. He received his BSc in applied mathematics from the University of Athens where he undertook the additional specialty in mathematics education. His research interests lie in the design and evaluation of intelligent technology that provides opportunities for feedback directly to students and enables educators and researchers to better understand learning and answer questions about the settings and context in which it takes place.*

Manolis Mavrikis

George Magoulas *is a professor of computer science at Birkbeck, University of London. He received a BEng, MEng, and PhD in electrical and computer engineering from the University of Patras (Greece) and has been teaching and researching in the area of intelligent adaptive and learning systems since 1993. His research activities fall under the umbrella of intelligent technologies involving key information processing methods such as fuzzy systems, neural networks, and global search, in particular differential evolution and particle swarms. His research work has received awards from the ACM, the IEEE, the EUNITE, and the IADIS.*



George Magoulas