Probabilistic Student Modelling to Improve Exploratory Behaviour

ANDREA BUNT and CRISTINA CONATI
Department of Computer Science, University of Bristish Columbia, Vancouver, B.C., Canada, E-mail: bunt@cs.ubc.ca, conati@cs.ubc.ca

Abstract.
This paper presents the details of a student model that enables an open learning environment to provide tailored feedback on a learner's exploration. Open learning environments have been shown to be beneficial for learners with appropriate learning styles and characteristics, but problematic for those who are not able to explore effectively. To address this problem, we have built a student model capable of detecting when the learner is having difficulty exploring and of providing the types of assessments that the environment needs to guide and improve the learner's exploration of the available material. The model, which uses Bayesian Networks, was built using an iterative design and evaluation process. We describe the details of this process, as it was used to both define the structure of the model and to provide its initial validation.

Key words: Student Modelling, Open Learning Environments, Exploration, Bayesian Networks, Adaptive Feedback

1. Introduction

Over the past few years, open learning environments have been introduced as an alternative to the traditional tutor-controlled, computer-based, educational systems. While tutor-controlled systems follow the principle that the learner should be presented with a restricted set of activities to work on under the supervision of a computerized coach (e.g., Anderson et al., 1995; VanLehn, 1996), open learning environments place less emphasis on supporting learning through explicit instruction and more on providing the student with the opportunity to learn through free exploration of the instructional domain. Learners usually explore these environments by engaging in a variety of activities supported by the environment's interface. These activities often involve simulations where learners can experiment with different aspects and parameters of a given phenomenon to observe the effects these changes have on outcomes of the simulation (e.g., Smithtown (Shute and Glaser, 1990)). The idea is that through performing a meaningful set of experiments and by generalizing the results of those experiments, learners can acquire deeper, more structured understandings of concepts in the domain than they would from a less active style of learning (Shute and

Glaser, 1990; van Joolingen, 2000). These environments should also allow learners to practice and acquire the meta-cognitive skills associated with effective exploration (Njoo and de Jong, 1993). These meta-cognitive skills include hypothesis formation, the ability to construct meaningful experiments, self-monitoring (the ability to monitor one's progress and understanding), and self-explanation (the process of spontaneously generating explanations to oneself to better clarify and elaborate upon given instructional material (Chi, 2000)).

Several studies have shown, however, that open learning environments do not always achieve the expected learning objectives. In particular, it has been shown that the effectiveness of these tools strongly depends on a number of user-specific features, including the learner's activity level (Njoo and de Jong, 1993; Shute and Glaser, 1990; Shute, 1993), whether or not the learner already possesses the meta-cognitive skills necessary to learn from exploration (van Joolingen, 1999; Reiser et al., 1994; Shute and Glaser, 1990; de Jong and van Joolingen, 1998), general academic achievement (Recker and Pirolli, 1992; Reiser et al., 1994) and cognitive ability (Veermans et al., 2000). A mismatch between a particular user's features and those necessary to learn from open environments usually results in learning difficulties. Examples of these difficulties include failing to initiate experiments (Shute and Glaser, 1990), being unable to interpret and generalize results properly (van Joolingen and de Jong, 1991), and failing to perform experiments that cover all important domain concepts (Reiser et al., 1994).

In this paper, we propose a solution to the limitations of open learning environments that relies on a user model designed to assess the effectiveness of a learner's exploratory behaviour. The goal of the model is to understand both when a learner needs to be supported in the exploration process and how to provide this support in a tailored and timely manner, without taking away the sense of freedom and control that is a key asset of open learning environments. Thus, the primary objective of the work presented in this article (an extension of Bunt and Conati (2002)) is to investigate how to incorporate a student model into an open learning environment to enable the provision of intelligent support, making the environments beneficial for learners with a wider range of learning styles and abilities. To meet this objective, the work has the following goals:

1. To investigate what features are needed in a student model to assess the effectiveness of a learner's exploratory behaviour.
2. To create a student model that can accurately assess these features while the learner interacts with the environment.
3. To provide initial evidence on the accuracy of the model through empirical studies.

The Student Model for exploration that we present in this paper has been implemented in the context of the Adaptive Coach for Exploration

(ACE) (Bunt et al., 2001), an intelligent exploratory learning environment for the domain of mathematical functions. ACE uses the assessment of the Student Model to provide tailored hints targeted at guiding and improving the learners' exploration of the material provided throughout the course of the interaction.

The outline for the rest of the paper is as follows. Section 2 discusses the challenges of modelling effective exploratory behaviour and how the model proposed here addresses these challenges. Section 3 presents the ACE environment. Sections 4 and 5 describe the design and evaluation of the first version of the ACE Student Model. Section 6 describes how the first version has been revised in light of the evaluation results, and section 7 presents a small evaluation of the changes from the previous design. Section 8 is a discussion. Section 9 describes related work, and finally, section 10 concludes.

2. Challenges of Modelling Effective Exploration

Two main challenges make modelling exploratory behaviour in open learning environments an especially hard problem, which is why it has rarely been tackled, despite mounting evidence that adaptive support for exploration is needed.

The first challenge involves the type of skill being modelled. For more structured educational activities, such as problem solving and question answering, there is usually a clear definition of correct behaviour that allows this behaviour to be recognized and formalized in a model. Conversely, for exploration, there is still no clear understanding of exactly what effective exploration consists of or how to categorize learner behaviour as either good or bad exploration. We address this problem by designing the model in an iterative fashion. We begin by building an initial model using our intuition of what constitutes effective exploration in an environment such as ACE and evaluating its performance with a formal study. The results of this study are used to gain further insights on what types of exploratory behaviour enhance learning and how to change the model to capture these behaviours, or the lack thereof. We then evaluate the changes to complete the cycle.

The second challenge of modelling exploration is that even with a good definition of effective exploratory behaviour, it will be difficult to obtain reliable information on this behaviour, since open learning environments tend to provide only low bandwidth information on their users. The bandwidth issue relates to the amount and quality of information available to a user model (VanLehn, 1988). Low bandwidth means that there is limited explicit information on the user traits or behaviours that the model needs to assess, thus increasing the uncertainty involved in the modelling process.

Obtaining sufficient bandwidth in open learning environments is particularly problematic. The types of skills and behaviours relevant to learning effectively through exploration, such as self-monitoring and self-explanation, are difficult to observe unless the environment's interface is designed to force learners to make them explicit, as has been done by other researchers trying to monitor meta-cognitive skills (e.g., Conati and VanLehn, 2000). But such an interface clashes with the unrestricted nature of open learning environments. Thus, a model for exploratory behaviour is bound to deal with low bandwidth information, which introduces a high level of uncertainty to the modelling task. Our model uses Bayesian Networks (Pearl, 1988) to help manage this uncertainty.

3. The ACE Open Learning Environment - Overview of Components

The Adaptive Coach for Exploration (ACE) (Bunt et al., 2001) is an intelligent open learning environment for the domain of mathematical functions. ACE is typical of many open learning environments in that it provides learners with a set of activities to explore in a free and flexible manner. However, ACE was also designed to investigate if a purely exploratory learning environment could become beneficial for a wider range of learners by adding a minimum amount of tailored support. By providing this support only when necessary, this approach has the potential to retain all of the advantages of pure open learning environments for those learners possessing the skills necessary to learn effectively from free exploration, while still addressing the needs of learners who require more structure.

ACE consists of 4 main modules:

1. A GUI designed to allow learners to explore various aspects of mathematical functions (up to polynomial functions).
2. A Knowledge Base that contains information on function-related concepts and interesting cases for exploration.
3. A Student Model designed to assess the effectiveness of a learner's exploratory behaviour.
4. A Coach that provides tailored feedback on the student's exploration process using the assessment of the Student Model.

In this section we describe the aspects of ACE's interface, knowledge base and coaching components that are necessary to understand the design and application of the Student Model, which is described in the next section.

3.1. The ACE Interface

ACE presents the learner with various activities, divided into units and exercises. Units are collections of exercises with a common theme, while

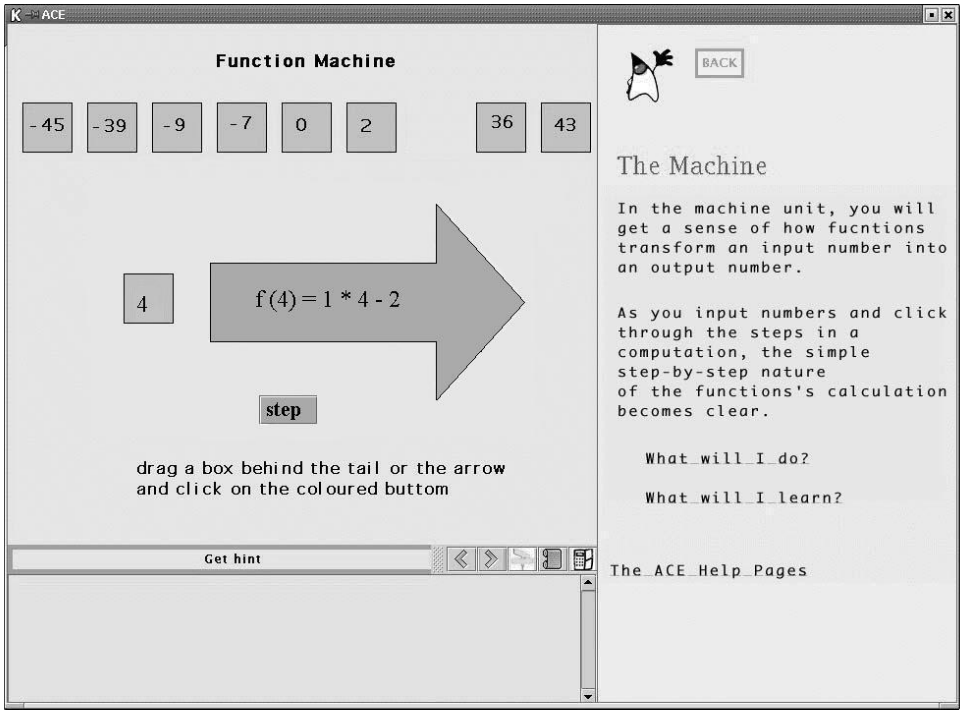

Figure 1. The ACE interface for the Machine Unit

exercises within the units differ in function type and equation. Figure 1 is a screen shot of the complete interface for one of ACE's units. The top-left window is the main interaction window, where the learner works on the exercises that ACE provides to explore various function concepts. The right panel is a set of hypertext help pages that contain instructions on how to use ACE and function-related definitions. The bottom panel displays messages containing the tailored feedback that can be obtained from the ACE's coaching component. A tool bar separates the main interaction panel from the feedback panel, containing access points to several help features and navigation tools.

3.1.1. *ACE Units*

Currently, ACE has three units: the Machine Unit, the Arrow Unit and the Plot Unit.

The Machine Unit (fig. 1) provides the learner with the opportunity to explore the relationship between the input and output of a given function.

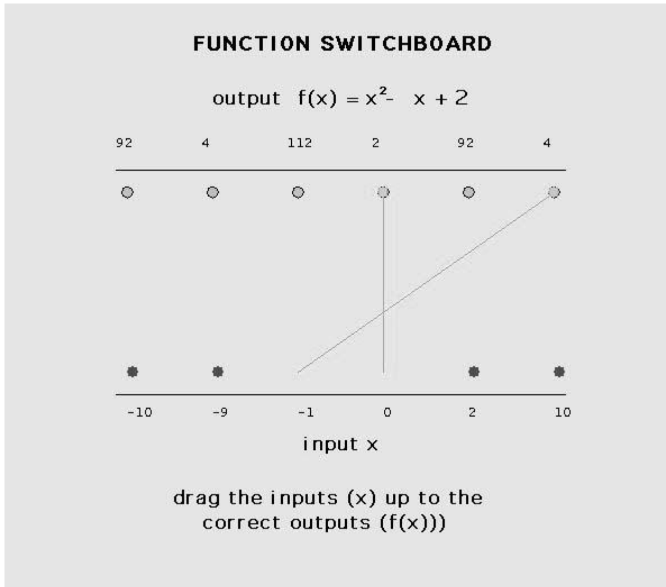

Figure 2. The Arrow Unit

The learner can explore this relationship by dragging any number of inputs displayed at the top of the screen to the tail of the function "machine" (the large arrow shown in fig. 1). The machine computes the output and spits it out at the other end of the arrow by encasing the output in an animated pink ball. If there are multiple steps involved in the computation (e.g., substitution and algebraic operations), the learner must repeatedly click the "step" button (found directly below the machine) to view the equation being resolved.

The Arrow Unit (fig. 2) is also designed to help the learner explore the input-output relationship, but requires more active thought on the part of the learner, who must both select which input to experiment with and connect it to the correct output. In the particular example shown in figure 2, the learner has connected the inputs -1 and 0 to the correct outputs 4 and 2. ACE gives feedback on the learner's actions by turning correctly connected arrows green and incorrectly connected arrows red. This is the only activity within ACE that has a clear definition of correct and incorrect knowledge application and for which ACE's Coach provides feedback on the correctness of the learner's actions.

The Plot Unit aims to help the learner gain an understanding of the relationship between the graph of a function and its equation, as well as

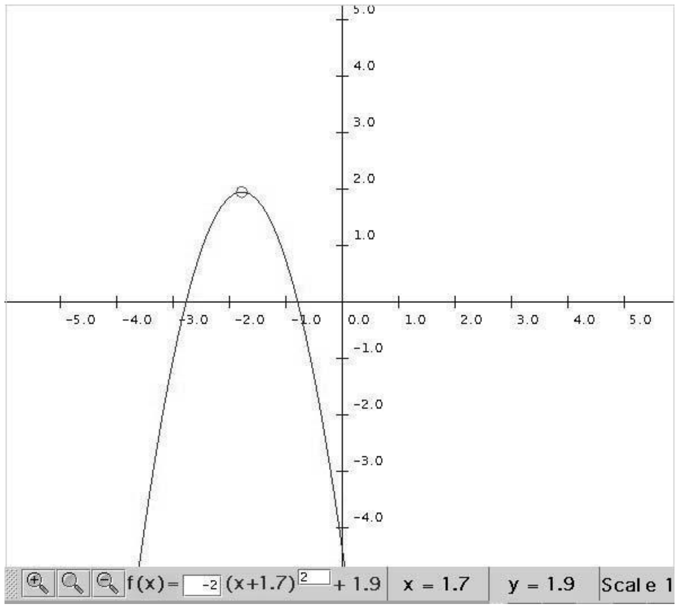

Figure 3. The Plot Unit

become familiar with different properties of graphs, including slopes and intercepts. The learner can manipulate a function's graph by either dragging it around the screen (using the mouse) or by editing the equation box (shown in the bottom-left hand corner of fig. 3). Changes in the position of the graph are immediately visible in the function's equation (also shown in a non-editable form in the top-right hand corner of the screen), and changes in the equation immediately update the graph. As an example, the learner could change the first coefficient in the equation from -2 to 2. This action would change the shape of the graph from concave upward to concave downward, illustrating the effect of positive and negative leading coefficients on the shape of the graph. The learner can also zoom in and out using the magnifying glass icons at the bottom of the panel to view the graph from a variety of different perspectives.

As the description above should show, the Plot Unit is ACE's most interesting unit in terms of the range of exploration that it permits. However, although the Machine and Arrow units may not permit the same degree of exploration, they are still exploratory in the sense that there are a number of inputs to choose from. This element of choice can cause a variety of learner behaviours, ranging from being too rushed or timid to explore throughly, to methodically selecting all of the inputs.

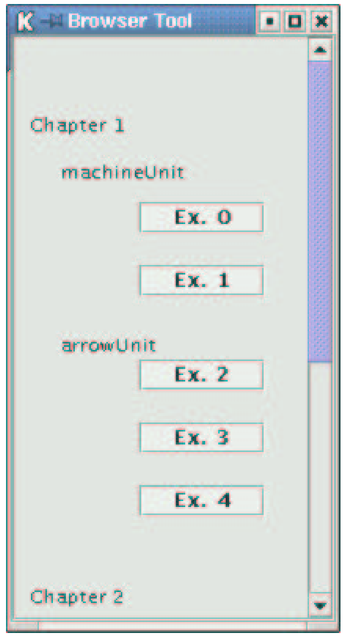

Figure 4. The Curriculum Browser

3.1.2. *Curriculum Navigation*

The learner has two ways to navigate through the curriculum. The first is by clicking on the forward and backward arrows on the tool bar (see fig. 1). The second way is by using the curriculum browser shown in figure 4. This tool lets the learner jump to any exercise in the curriculum and is built at run-time using information stored in a text-file. Clicking on the icon in figure 1 that looks like a scroll activates this tool.

3.2. ACE's Coach

ACE's coaching component provides learners with tailored feedback on their exploration process through hints, which can be obtained by clicking on the "Get Hint" button (located on the tool bar in fig. 1). Hints are supplied to the learner at increasing levels of detail, ranging from a generic suggestion to explore the current exercise more thoroughly to exactly what things to explore. The Coach uses the Student Model's assessment to determine if the learner needs a hint, and if so, which concept the hint should target.

If the learner tries to move on to a new exercise (using one of the navigation mechanisms) before the Coach feels that the learner has explored the current exercise sufficiently, the Coach will generate a warning, suggesting better exploration of the current exercise. These warnings also remind the learner of the availability of hints. Since the system is designed to give the learner as much control as possible, the learner can either choose to follow the Coach's advice or to move on. As with hints, the Coach decides whether or not to generate this warning based on information obtained from the Student Model.

3.3. The Knowledge Base

ACE's Knowledge Base contains representations of functions and related concepts, such as substitution and arithmetic, as well as a representation of the interesting elements of exploration within a given activity. Since the exploration-related part of the Knowledge Base is used extensively by the Student Model to produce its assessment, a description follows. Further details on the knowledge base can be found in Bunt (2001).

In the Knowledge Base, *relevant exploration cases* are the salient concepts found in each exercise that should be explored by the learner to gain a thorough understanding of the target material. The set of concepts considered to be salient for a particular exercise depends on both which unit the exercise belongs to and the type of function the learner is exploring within the exercise.

The rules given in figure 5 define the *relevant exploration cases* for each function type within each unit. The first rule (R1 in fig. 5) states that, in the Machine Unit, if the function is a constant function then the learner should explore how the function behaves with small positive numbers (S+), small negative numbers (S-), zero (Z), large positive numbers (L+) and large negative numbers (L-). The rule is the same for a linear function (R2 in fig. 5). With power and polynomial functions (R3 and R4 in fig. 5), the learners should also see how the function behaves with a positive and negative version of the same number (Pos_Neg in fig. 5), since a function equation with even exponents generates the same output in both cases. Large numbers are not included as relevant exploration cases for either of these functions since they could cause some learners to get fixated on complicated arithmetic unnecessarily. The Arrow Unit has the same set of rules (R5-R8 in fig. 5) since it also deals with the relationship between input and output.

In the plot unit, the rules specify which function manipulations the learners should explore. For a constant function (R9 in fig. 5), they should view the graph with both positive and negative intercepts. The rule for linear functions (R10 in fig. 5) builds on this, adding also positive and negative slopes along with the zero slope, which turns the function back to a constant one. The relevant exploration cases for a power function (R11 in fig. 5) include shifting (resulting in the learner viewing the graph at a variety of

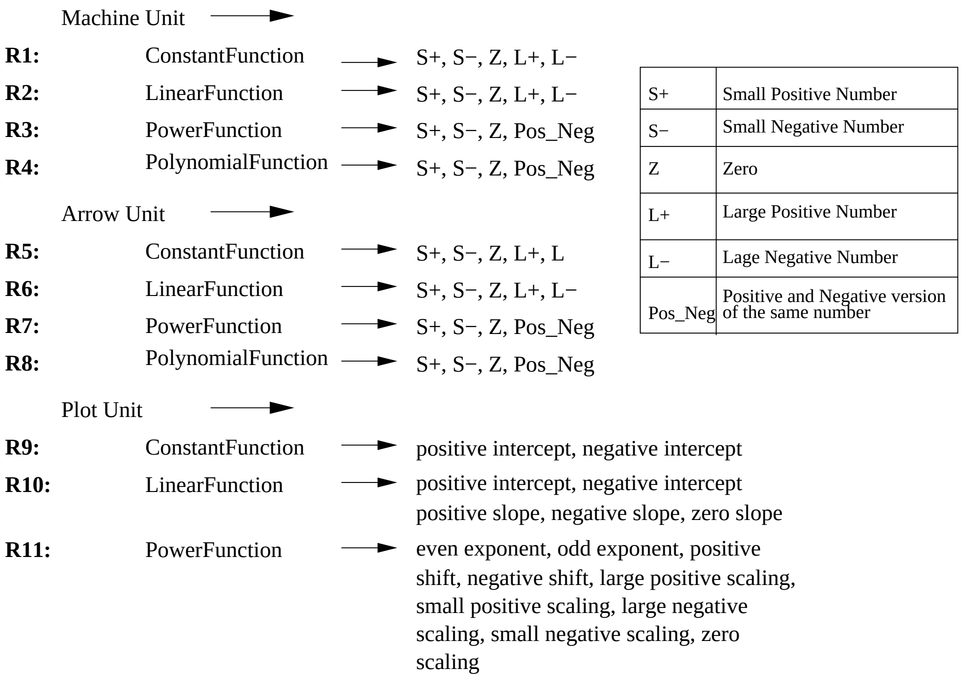

Figure 5. The Knowledge Base rules for the relevant exploration cases

intercepts), scaling (changing the width and orientation of the graph) and odd and even exponents (changing the shape of the graph). For this unit, Polynomial functions were not implemented in the version of ACE described here, but are planned for future versions.

4. ACE's Student Model - Version I

As stated in the introduction, the main objective of the Student Model is to generate an assessment of a learner's exploration of the ACE environment that allows the coaching component to provide tailored feedback aimed at guiding and improving this exploration. However, because the ACE interface maintains a natural, unrestricted interaction style, which gives the learner full control of the exploration process, the Student Model assessment is riddled with uncertainty. ACE gives the learners the freedom to choose which exercises to explore and which elements within an exercise to experiment with, without requiring them to explain the rational behind their actions. Furthermore, since there are very few activities with any notion of correct and incorrect behaviour, learners are free from having their skills and knowledge tested and evaluated. These freedoms contribute to lowering the bandwidth of information available to the Student Model, since the interface provides few indicators of the learner's actual reasoning and understanding. In particular, the Student Model can view which items in the interface the learner experiments with, but this is not always sufficient to determine what the learner is trying to accomplish with these actions. In addition, it is hard to infer directly if the learner's actions are contributing to exploration that is effective enough to learn the material. For example, the learner may perform many experiments, but unless they are well selected and unless the learner is able to draw the right conclusions from them, they will not necessarily contribute to an overall understanding of the concepts targeted by ACE. Thus, the assessment task of ACE's Student Model entails a high level of uncertainty. We use Bayesian Networks to clearly formulate and process this uncertainty in a principled way.

This section describes the first iteration of the model's design and construction, while section 6 discusses a second iteration. The main difference between the two versions is the treatment of knowledge. The first version uses evidence of effective exploration in conjunction with direct skill application to update beliefs of the learner's knowledge. The second version is geared more toward incorporating the learner's explicitly demonstrated knowledge into the model's assessment of effective exploration. We describe both the initial model and its refinement for two reasons. The first reason is that the design and evaluation of the second model depend on the results of the first evaluation. The second, more compelling reason, is that discussing the details of the refinement process helps answer one of the open questions that this work aims to address: what constitutes effective exploration in an open environment such as ACE?

4.1. High Level Description of the Model's Bayesian Networks

ACE's Student Model is separated into two Bayesian Networks: one for the input/output units (Machine and Arrow) and another for the graph manipulation unit (Plot). Separating the model into two networks helps to both divide the initial design task into two more manageable components and address some performance concerns. The two networks have a common high-level design, which this section describes.

The Student Model's networks formalize a method for obtaining evidence of good exploration using the *relevant exploration cases* discussed in section 3.3. Figure 6 provides a high-level description of the interactions among the different types of nodes in the networks. Each network in ACE's Student Model consists of two types of nodes: exploration nodes and knowledge nodes.

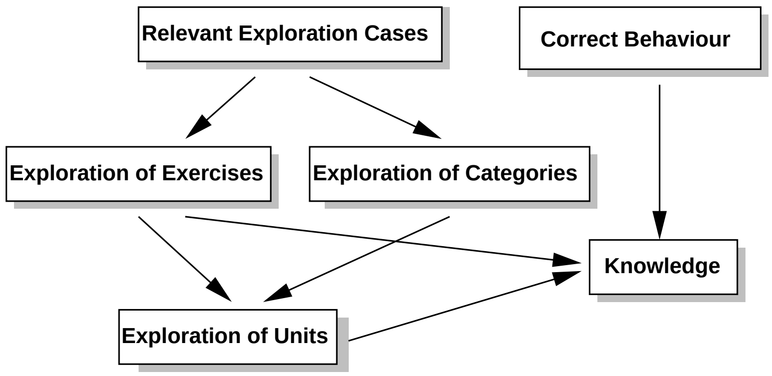

Figure 6. A high-level description of the Bayesian Networks

Exploration nodes represent the effectiveness of the learner's exploratory behaviour, while knowledge nodes represent the learner's understanding of relevant domain concepts. Several classes of exploration nodes are present in the network:

– *Relevant Exploration Cases*: the exploration of individual exploration cases in an exercise (e.g., dragging the number 3, a small positive input, to the back of the function machine in the Machine Unit).
– *Exploration of Exercises*: the exploration of individual exercises (e.g., exercise #1 in the Machine Unit).
– *Exploration of Units*: the exploration of groups of related exercises (e.g., all of the exercises in the Plot Unit).
– *Exploration of Categories*: the exploration of groups of relevant exploration cases that appear across multiple exercises (e.g., all of the relevant exploration cases involving small positive inputs).

Each exploration node can take on the value of either True of False. A True value means that the learner has sufficiently explored the item associated with the node (i.e., the relevant exploration case, exercise, unit or category), where sufficient means that the learner has performed enough meaningful interface actions to indicate thorough exploration.

Knowledge nodes in figure 6 also have binary values, where a True value represents the probability that the learner understands the domain concepts covered by the curriculum. Two types of exploration nodes (exploration of categories in the Machine/Arrow Units and the exploration of units in the Plot Unit) are used in conjunction with evidence of correct behaviour (when observable) to assess the learner's knowledge (see fig. 6). Evidence of good exploratory behaviour carries less weight than does evidence of correct skill application, as is reflected in the Student Model's Conditional Probability

Tables (CPTs) for knowledge nodes (which we describe in more detail in sec. 4.4). In ACE's initial design, however, evidence of correct skill application currently comes only in the Arrow Unit. Thus, most of the knowledge assessment is done using the learner's exploratory behaviour.

Assessment at different levels of granularity allows the coaching component to provide a wide range of tailored feedback. First, the assessment of individual exploration cases and categories of cases can be used to select the content of hints, which aim to improve the exploration of individual exercises. Second, the assessment of individual exercises allows the Coach to generate a warning when the learner attempts to leave an exercise that has not been thoroughly explored. Finally, the assessment of how well the learner has explored categories and related exercises can be used to adapt the curriculum and to provide suggestions as to which exercises to explore next. Currently, ACE's Coach does not fully support the latter type of feedback but, because the relevant assessments are already provided by the Student Model, it could easily be extended to do so.

4.1.1. *Static and Dynamic Parts of the Student Model*

Each Bayesian Network in the Student Model has a static portion and a dynamic portion. The static portion is completely hand-specified ahead of time, and its structure remains identical for every learner and session. In addition to this static component, each network has a dynamic portion that is automatically constructed during a particular interaction.

The reason for having part of the network be dynamic is that there are components of the modelling task that vary from session to session. First, the curriculum is generated automatically at run-time from a text file. This text file contains a list of exercises for each unit, including the specifics of each exercise's function. Although ACE has a recommended curriculum, instructors can modify the text file to contain a different set of exercises. Thus, each session could potentially have a different curriculum. Adding nodes dynamically to the network removes any curriculum-dependent information from the static portion, avoiding situations where the model designer has to build a separate network for every conceivable curriculum. In addition to the structure of the curriculum, the exact nature of each exercise is unknown ahead of time since the coefficients and exponents of the function equations are randomly generated. Finally, using ACE's navigation tools, each learner may take a different path through the curriculum, choosing to explore some of the exercises and ignore others.

Most of these issues could be addressed by automatically constructing the networks to contain all exercises, rather than building the networks dynamically throughout the interaction. This would result, however, in networks that are unnecessarily large. Furthermore, it is not immediately obvious how

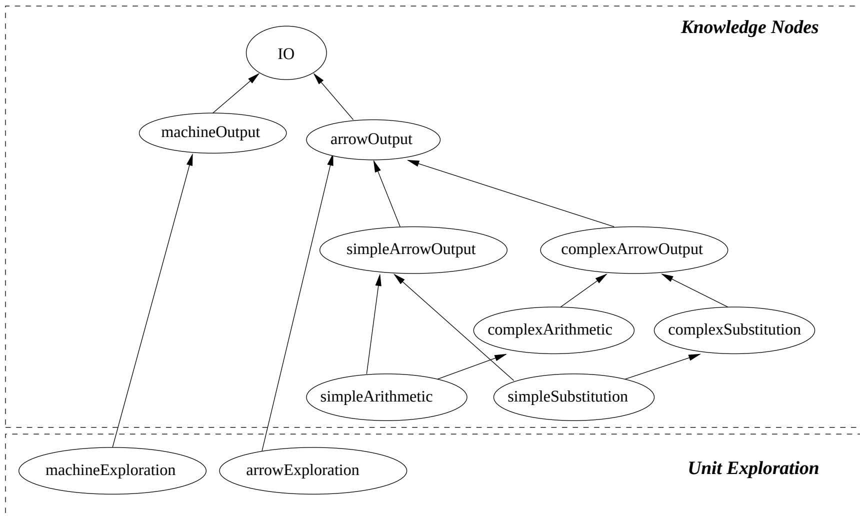

Figure 7. The static portion of the Bayesian Network for the Machine and Arrow Units

to define the semantics of nodes representing items that the learner has never been exposed to (e.g., particular exercises). Since the learner has yet to explore these items, their values should somewhat represent the learner's predicted performance in future explorations, rather than the learner's assessed behaviour and knowledge up to this point. However, representing this distinction requires either substantially increasing the network complexity, as done in Murray and VanLehn (2000), or adding an interpretation procedure external to the network, as done in Conati et al. (2002). We circumvent the problem by adding only nodes relevant to the exploration that the student has actually performed - an approach similar to that adopted by the student model in POLA (Conati and VanLehn, 1996).

In the following subsections we provide more detail on both the static and dynamic parts of the Student Model's networks.

4.2. Static Portions of the Student Model

Figure 7 shows the static portion of the network for the Machine and Arrow Units. This network contains nodes representing the learner's knowledge of concepts and skills involved in generating the input and output of a function, such as substitution and arithmetic operations (see nodes "simpleArithmetic," "simpleSubstitution," "complexArithmetic," and "complexSubstitution" in fig. 7). In addition, the network contains nodes that represent

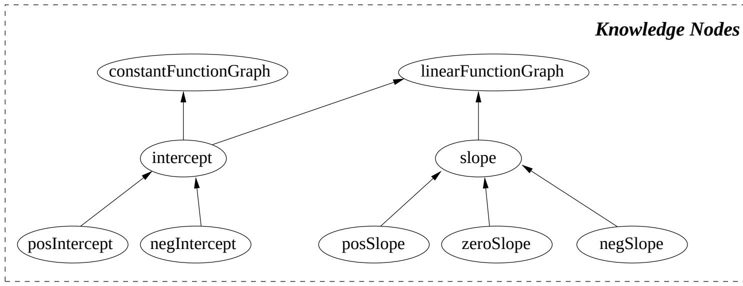

Figure 8. Example of knowledge nodes in the network for the Plot Unit

higher-level effective exploratory behaviour of individual units (e.g., "arrowExploration" in fig. 7). In all subsequent figures, the nodes that are labeled with "exploration" in conjunction with those labeled "e_i" and "e_iCase$_i$" refer to exploration nodes, while all of the others refer to knowledge nodes or instances of correct behaviour.

Figure 8 depicts some of the knowledge nodes in the static part of the network for the Plot unit pertaining to the graphs of constant and linear functions. Understanding the graph of a constant function involves understanding intercepts (positive and negative), while a linear function also involves understanding different slopes (positive, zero and negative). There are also nodes in the static part of the Plot Unit's network for the knowledge related to power functions, including nodes for concepts such as shifting, scaling and exponents. The general idea is that more complicated functions have more complicated knowledge hierarchies in the network.

4.3. Dynamic Portions of the Student Model

The portions of the networks that are added dynamically contain nodes representing the exploration of individual exercises (see "Exercise Exploration" layer in figs. 9 and 10), the exploration of relevant exploration cases (see "Relevant Case Exploration + Time" layer in figs. 9 and 10), and the exploration of categories (see "Category Exploration" layer in figs. 9 and 10). The effectiveness of a learner's exploration of an individual exercise is represented by the nodes labeled "e_i" (e.g., "e_1" in fig. 9). These nodes are added to the network every time a learner visits a new exercise. For instance, the network in figure 9 shows that the learner has visited three exercises: "e_1", "e_2" and "e_3". As exercise nodes are added to the network, they are linked to their corresponding unit nodes. As illustrated by figure 9, the nodes "e_1" and "e_2" represent exercises in the Machine Unit and the node "e_3" represents an exercise in the Arrow Unit.

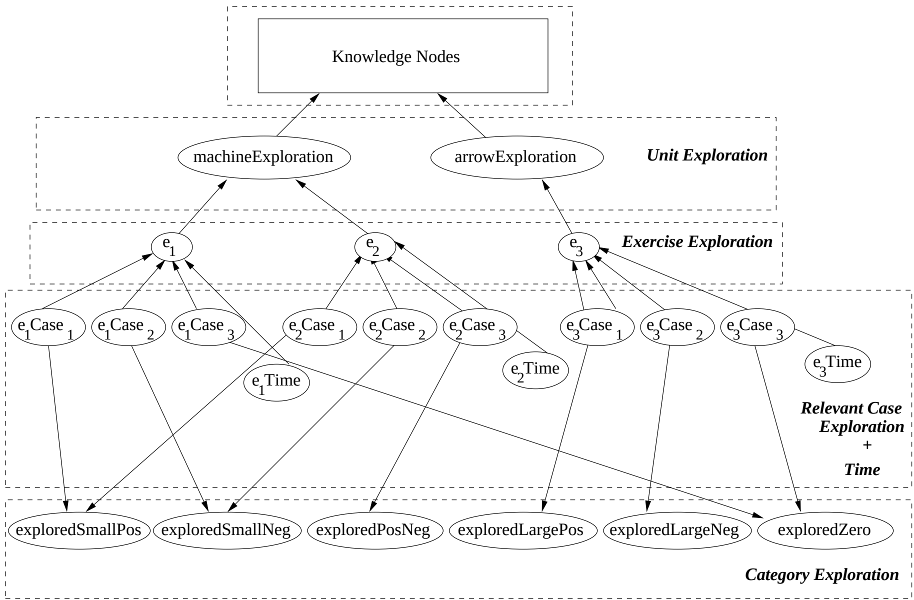

Figure 9. Exploration nodes in the dynamic portion of the Bayesian Network for the Arrow and Machine Units

The degree to which the learner is considered to have explored an individual exercise is influenced by whether or not she has explored all of the salient cases within that exercise in addition to how long she spent exploring the exercise. This relationship is represented in the network by adding nodes for each of the relevant exploration cases associated with the function presented in the exercise (obtained from the knowledge base) and a node representing exploration time. The relevant exploration case nodes are labeled "e_iCase$_i$" (e.g., "e_1Case$_1$" in fig. 9) and the time node is labeled "e_iTime" (e.g., "e_1Time" in fig. 9). The time node represents a measure of how long the learner spent exploring the exercise as a whole, not each individual exploration case. A measure of how long the learner spent exploring each case would be more informative, but ACE's current interface cannot reliably tell how long the user has spent focusing on individual items. How to measure time spent exploring a particular case is discussed at the end of this paper.

Since the relevant exploration case nodes are instances of exploration categories of concepts that can appear across multiple exercises, nodes are added to the network representing the more general effectiveness of the learner's exploration of these categories (e.g., "exploredSmallPos" in fig. 9). If an

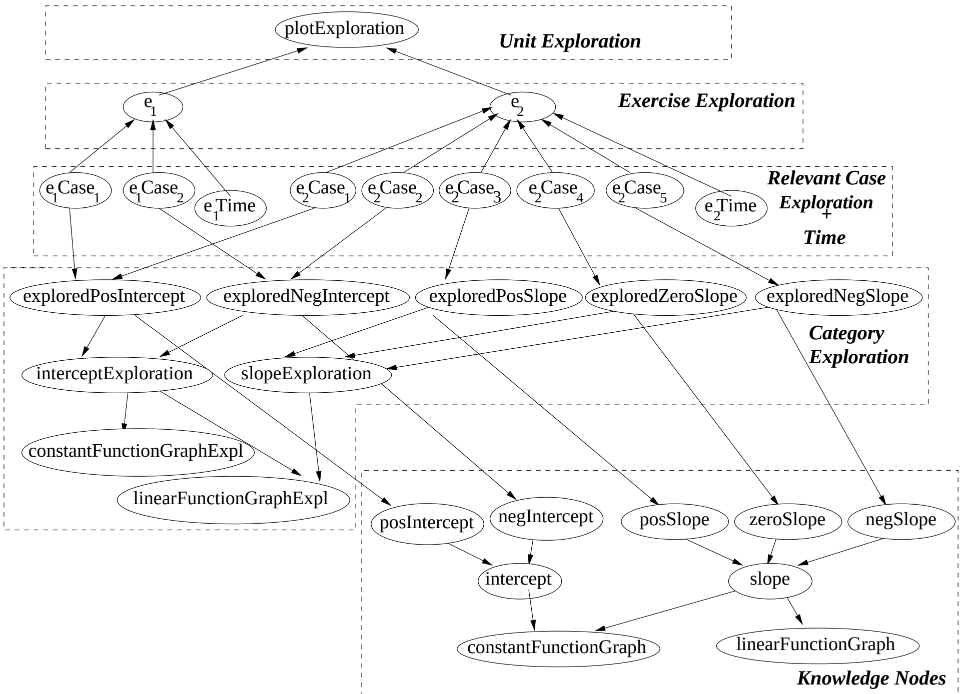

Figure 10. An example dynamic portion of the Bayesian Network for the Plot Unit

exploration category is not in the network already, it is added dynamically at the same time as the corresponding relevant exploration case.

Figure 10 is an example of the dynamic portion of the network for the Plot Unit. There are two differences between the two networks. The first is that the exploration categories in the Plot Unit (e.g. "exploredPosIntercept") form the base of a hierarchy that mirrors the knowledge hierarchy. The second difference is the manner in which the knowledge nodes are updated. In the Plot Unit, there is a meaningful relationship between the exploration categories and the knowledge targeted by the unit. Therefore, the knowledge nodes are influenced directly by the appropriate categories (see the "Knowledge Nodes" layer in fig. 10). Knowledge the learners should be acquiring through exploration in the Machine and Arrow units, on the other hand, corresponds better to the learner's exploration of the units as a whole, than it does to the particular exploration categories, which represent different types of inputs. Thus, in this network, the knowledge nodes (abstracted by the box labeled "Knowledge Nodes" in fig. 9) are influenced by the unit exploration nodes (see "Unit Exploration" layer in fig. 9).

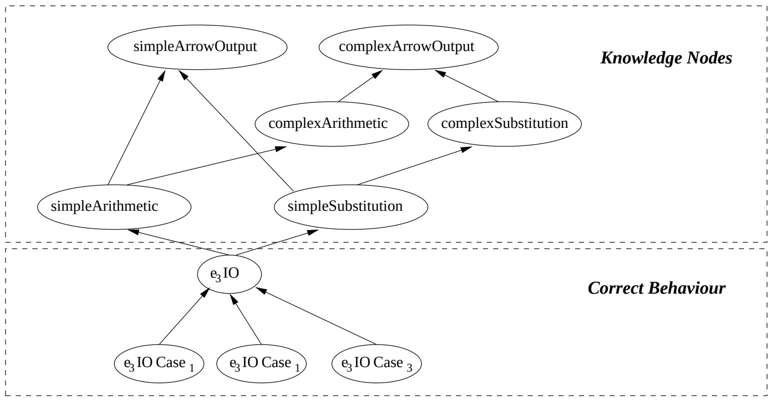

Figure 11. Part of the dynamic portion of the Bayesian Network for the Arrow and Machine Units including knowledge nodes

4.3.1. *Correct Behaviour*

The final part of the network added dynamically (shown in fig. 11) deals with the Arrow Unit - the only unit that gives learners the opportunity to demonstrate their knowledge directly. In this unit, nodes representing proficiency in correctly mapping the input and output of a function are added to the network (e.g., e_3IO in fig 11). These "e_iIO" nodes directly influence knowledge nodes that represent the type of substitution and arithmetic knowledge needed to generate the correct output for the function in question. When a learner connects an input to an output in the Arrow Unit, a new node labeled "e_iIOCase$_i$" (e.g., "e_3IOCase$_1$" in fig. 11) is attached to the "e_iIO" node and set to "True" if the answer is correct and to "False," otherwise.

4.4. Conditional Probability Tables

The Conditional Probability Tables (CPTs) and prior probabilities in the Student Model's Bayesian Networks were defined using our best estimates of the corresponding probabilistic dependencies and prior beliefs. As is the case with all human-designed CPTs, such estimates need to be refined further through empirical evaluations. This section highlights some of their key features.

As illustrated in figure 12, the CPTs for the exercise exploration nodes (e.g., "e_1" in fig. 9) are set so that the more relevant exploration cases the learner explores, the higher the probability of having effectively explored that function. Spending a sufficient amount of time on the exercise further strengthens this belief, although sufficient time by itself results in a very

e_n Case 1	e_n Case 2	e_n Case 3	e_n Time	e_n
T	T	T	T	0.97
			F	0.75
		F	T	0.75
			F	0.50
	F	T	T	0.75
			F	0.50
		F	T	0.50
			F	0.25
F	T	T	T	0.75
			F	0.50
		F	T	0.50
			F	0.25
	F	T	T	0.50
			F	0.25
		F	T	0.25
			F	0.03

Figure 12. The Conditional Probability Table for an example exercise exploration node e_n.

simpleArrowOutput	complexArrowOutput	arrowExploration	arrowOutput
T	T	T	0.97
		F	0.8
	F	T	0.7
		F	0.2
F	T	T	0.8
		F	0.4
	F	T	0.6
		F	0.03

Figure 13. The Conditional Probability Table for a knowledge node (arrowOutput) given explicit evidence (simpleArrowOutput and complexArrowOutput) and the exploration of that concept (arrowExploration).

low probability of effective exploration. The same technique (minus the time nodes) is used to update the CPTs for the unit and category exploration nodes. The CPT for a knowledge node (see fig. 13 for an example) defines the combined influence of explicit evidence of the learner's understanding of the corresponding concept and the learner's exploration of that concept. If the learner has only explored the concept well and there is no explicit evidence of understanding, the model is less confident that the learner understands it, but still considers good exploration to be an indication of some degree of understanding.

4.5. The Model's Operation

Probabilities in the networks change as learners explore relevant exploration cases, since as they cover these cases, new evidence is introduced. This evidence then updates the model's assessment of the learner's exploratory behaviour and knowledge.

The exploration case nodes are all set to "False" when they are initially added to the network and are not set to "True" until the learner performs interface actions that the system considers to be indications that the learner has explored the associated cases. Each unit has a different interpretation of which interface actions provide such evidence. In the Machine Unit, dragging an input that is an instance of a relevant exploration case (e.g., a small negative number) to the Machine is considered evidence of the learner having explored that case. In the Arrow Unit, corresponding evidence is introduced when the learner drags a relevant input to an output. Finally, the learner is considered to have explored a relevant exploration case in the Plot Unit when he either drags and drops the graph to an interesting position belonging to a particular case (e.g. dragging the graph to a location where it has a positive y-intercept) or edits the function equation to change the graph in a desired way.

In addition to the relevant exploration case nodes, evidence is also introduced into the network through the time nodes (e.g., "$\mathrm{e_1Time}$" in fig. 9), which are observed as "True" when the time that the learner has spent exploring an exercise exceeds a pre-specified threshold. Finally, information regarding correct and incorrect knowledge application is gathered in the Arrow Unit by setting the values of the nodes for the generation of input and output (e.g., "$\mathrm{e_3IOCase_3}$" in fig. 11).

The time threshold for each unit is the minimum length of time necessary to explore an exercise in that unit in a careful manner. We determined the threshold for each unit by timing one of the system designers as they used ACE, followed by pilot testing with two users. These thresholds are not meant to be definitive measures of the amount of time needed to explore

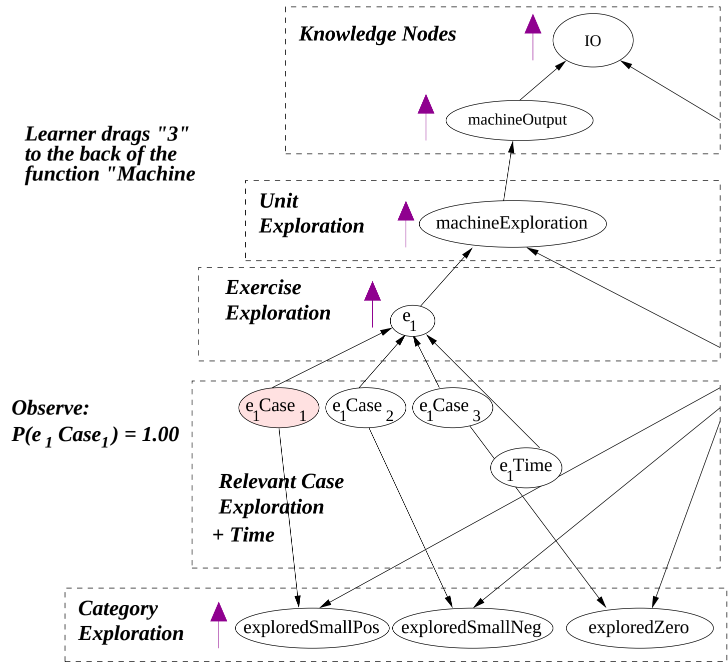

Figure 14. An example assessment

each exercise. Instead, they aim to provide another source of information that the model can use as a sign of good exploration.

4.6. An Example Interaction

As a concrete example (depicted in the example portion of the network in figure 14), consider the case where a learner is exploring exercise 1 ("e_1" in fig. 14) in the Machine Unit. The learner's first action is to drag the input "3" to the back of the function machine. At this point, since 3 is a small positive input, the node "e_1Case_1" in figure 14 is set to "True," increasing the probability that the learner has effectively explored the exercise, along with the probabilities that the learner has effectively explored the unit ("machineExploration" in fig. 14") and the category of small positive inputs ("exploredSmallPos" in fig. 14). These probabilities remain the same until the learner covers a different case (a small negative input or the zero input).

As the learner covers the relevant exploration cases, the probabilities for the pertinent knowledge nodes also increase. These increases, however, are not as substantial as those for the exploration nodes, unless the model has some explicit evidence of correct behaviour. In this example interaction, the probability that the student understands I/O (the "IO" node in fig.14) increases as the learner covers the relevant exploration cases, but only slightly, since the model does not yet have any explicit evidence of correct behaviour. If the learner goes on to exhibit good exploration in the Arrow Unit and correctly connects the inputs to the outputs, the "IO" node then reaches a very high probability. If the learner simply continues to explore thoroughly in the machine unit, the probability that the learner understands I/O will continue to increase (up to a certain point), but never achieves a very high probability.

We will now illustrate how the Coach generates its feedback using the example described above. More detail on the Coaching algorithms can be found in Bunt et al. (2001). As mentioned in section 3.2, ACE's coach generates two kinds of feedback. The first type of feedback is in the form of on-demand hints, which the learner can obtain at any time during the interaction. The Coach determines the focus of the hint by querying the exploration categories that are relevant to the current exercise, and hints on a category which has a probability below a pre-defined threshold (but has not been explored in the current exercise). Following the example interaction described above, if the learner asks for a hint after performing the first action, the Coach would hint on either small negative inputs or on the zero input, but not on small positive inputs.

The second type of feedback is generated when the learner tries to leave an exercise. At this point, the Coach queries the exercise exploration node. If the probability that the learner has explored the exercise is below a predefined threshold, the Coach queries the relevant category nodes to see if the learner has thoroughly explored the corresponding categories over the course of the interaction. If the Coach can find one category with a probability that is too low, the Coach generates a warning suggesting that the student explore the current exercise more thoroughly, and to ask for a hint if needed. Thus, in the previous example, if the student tries to move on to a new exercise after performing the described action, the Coach generates a warning since the exercise node and the categories of small negative inputs and the zero input all have low probabilities.

5. Evaluation of the First Version of the Student Model

In November 2000, we conducted a user study with ACE to verify the hypothesis that intelligent support in an open learning environment can help to make the experience more beneficial for all types of learners. While analyzing the learners' behaviour as they used the system gave insight on the general effectiveness of ACE's components, this section focuses on quantitative and qualitative results that pertain to the accuracy and assessment capabilities of the Student Model.

5.1. Participants

The subjects were first-year university students taking a computer literacy course. ACE's target population is actually high school students, but because of unforeseen difficulties in coordinating with a local high school, we were unable to gain access to this population. The university students we used, however, are still considered suitable since the teaching assistants for the course felt that many of their students had very limited math knowledge. To further ensure that subjects did not have too much knowledge we only accepted subjects who had not taken a university math course. A total of 14 subjects participated in the study.

5.2. Experiment Design

In the experiment, all subjects used the full ACE environment described in section 3. We decided not to divide the subjects into a control and an experimental group because the small subject pool, coupled with the fact that several subjects turned out to have better knowledge than expected, would have made it very hard to gain any significant information with a two-group design.

One to three subjects participated in each session, positioned so that they could not see each other. Each session was observed by one researcher who recorded informal observations. Sessions lasted between 60 and 80 minutes and consisted of a pre-test phase, a session with ACE and a post-test phase. The pre-test phase included a paper and pencil test designed to gauge learners' knowledge of the topics targeted by ACE. The test consists of 39 questions, divided equally into questions on function output recognition and generation, graph property recognition, equation property recognition and graph-equation correspondence. The post-test phase consisted of a test analogous to the pre-test and of a nine item questionnaire targeted at the learners' subjective view of their ACE experience. Several sessions were also captured on videotape. In addition, ACE was instrumented to produce log files that capture the sessions at the level of basic interface actions. Table I summarizes the key interaction events captured in the logs.

5.3. Study Results

Table II displays the average per student and the standard deviation for each of the event counts described in table I. In addition to these basic, descriptive statistics, linear regression analysis was used to verify whether there is any correlation between different aspects of ACE usage and student learning. In

Table I. Information recorded in the log files

Interaction Event	Description
Number of exercises visited	The number of exercises the learner chose to explore.
Number of exercises passed	An exercise is passed if ACE let the learner leave an exercise without a warning to stay and explore further.
Number of stay events	A stay event occurs when a learner follows the Coach's suggestion to explore further and remains in the current exercise.
Number of leave events	A leave event occurs when the learner does not follow the Coach's warning and chooses to move on.
Number of hints	The total number of hints accessed by the learner.
Number of exploratory actions	Actions performed by the learner that the Student Model uses as evidence of the exploration of a relevant exploration case.

Table II. Statistics on information recorded in the log files.

Interaction Event	Average per Learner	Standard Deviation
Number of exercises visited	19.86	6.64
Number of exercises passed	12.71	5.36
Number of stay events	5.85	2.71
Number of leave events	5.79	4.14
Number of hints	5.07	3.41
Number of exploratory actions	168.07	60.18

each regression analysis described in this section, post-test scores are used as the dependent variables and pre-test scores are added to the regression equation to control for incoming knowledge.

The following event counts are found to be significant positive predictors of post-test scores (after controlling for pre-test scores):

1. Total number of hints accessed [$p = 0.0406$, $R^2 = 84.6\%$]
2. Number of exercises passed [$p = 0.0093$, $R^2 = 87.9\%$]

The first result provides an initial indication that ACE's support of the exploration process, in terms of hints generated using the Student Model's assessment, improves learning. The second result provides encouraging evidence that the Student Model's assessment accurately reflects students' learning in the environment because the Coach determines that a learner has passed an exercise when the Student Model indicates that it is likely the learner has sufficiently explored and understood the relevant skills and concepts.

The above results may also be attributable to additional factors that were not controlled for in the study, including the learner's general academic ability and conscientiousness. This seems not to be the case, however, since there is no significant correlation between the two event counts. A study that controls for these factors is required to definitely rule out this possibility.

The total number of exploratory actions that learners performed was not found to be a significant predictor of learning. This may be because of how the exploratory actions were recorded in the log files. Every interface action that indicated exploration of a relevant exploration case was counted, even multiple actions repeated for the same case. The Student Model, however, uses only one such action to set a relevant exploration case node to "True." Subsequent exploratory actions related to the same case are considered by the Student Model to be redundant. In fact, a few cases of learners performing redundant explorations are observed, which is consistent with findings showing that the inability to self-monitor is one of the common problems in open learning environments (van Joolingen, 1999). If learners don't have the self-monitoring skills necessary to understand when they have explored a concept sufficiently, they likely begin to over-explore that concept rather than move on.

The learners' answers to the post-test questionnaires indicate that they enjoyed using ACE and found it effective. For example, 79% of the learners found it useful when the Coach suggested that they continue to explore a certain exercise and 71% prefered to learn in this manner over a textbook (21% were neutral).

5.4. Utility of the Student Model

The work presented here relies on the assumption that it is beneficial to include a student model that can assess exploratory behaviour in an open learning environment because of the tailored support that it permits. Although this study does not provide conclusive evidence of the validity of this assumption, anecdotal evidence of subjects' interactions with ACE does highlight the importance of developing a model to assess exploration. To illustrate this point, the following subsections summarize a few key examples of learners using the system.

5.4.1. *Case 1: Succeeding in Uncovering Concepts*

This learner scored poorly on the pre-test questions involving negative slopes. He did not experiment with negative slopes in the Plot Unit until he received a navigation hint from the Coach. At this point he asked for more hints and obtained one targeted at negative slopes. As a consequence, he performed the appropriate explorations and improved on the post-test. The benefit of tailored hints was also very noticeable in two other participants.

5.4.2. *Case 2: Passive/Timid Learner*

On the pre-test, this learner exhibited a lack of understanding of constant functions. Her exploratory behaviour was adequate in the Machine Unit, but very poor in the Arrow Unit. In this unit, when she was presented with a constant function, she tried one input, connected it to the incorrect output, stared at the screen for a length of time and then tried to leave. When the Coach issued a warning she remained in the exercise, stared at the screen for another length of time and then chose to move on despite the Coach's second suggestion to stay.

This example illustrates that some learners are passive and/or timid in an open learning environment, and as a result, fail to improve. Providing hints upon request may not be the solution for these learners since some do not always access available help when needed (Aleven and Koedinger, 2000) (this subject did not access a hint despite the Coach's suggestions to do so). Thus, it is even more important to have a student model that can identify passive learners and allow the system to provide more proactive, unsolicited guidance when necessary.

5.4.3. *Case 3: Learner Failing to Uncover Concepts*

This learner was very active in the Plot Unit, but failed to improve on post-test questions targeted at the zero slope for linear functions. He experimented with all the relevant exploration cases in the Plot Unit except this one. The Coach warned him to stay when he tried to leave the exercise. He chose to stay in the exercise, but did not request a hint from the Coach. Although he remained an active explorer, he did not end up experimenting with a zero slope. There were three other participants who showed similar behaviours. These subjects show the potential benefit of having a model that can assess the details of a user's exploratory behaviour. This assessment would have allowed the Coach to provide hints targeting concepts these learners didn't cover, however, had the learners requested them.

The behaviour of these subjects, as well as the behaviour described in the previous section, confirms that user-initiated hints are not always adequate to elicit more thorough exploration. Furthermore, determining the appropriate hint level should be informed by the Student Model, since a generic suggestion to explore further may not be helpful to learners who have explored all but the more subtle concepts. This is also consistent with the finding that learners do not always benefit from high-level hints (Aleven and Koedinger, 2000).

5.5. Student Model Limitations

The log files also captured many of the Bayesian Network probabilities after a learner visited an exercise, including those for exercise exploration, unit exploration, exploration of any relevant categories, and for any pertinent knowledge nodes. None of these probabilities were significantly correlated with post-test scores. Observing learners during the sessions and a subsequent manual analysis of the log files uncovered some limitations of the first version of the Student Model that could potentially explain the absence of these correlations. The more significant limitations include the Student Model's treatment of knowledge levels and how it assesses whether or not a learner has effectively explored a relevant exploration case, as we discuss in the next subsections.

5.5.1. *Knowledge Levels*

Having several highly knowledgeable learners among our subjects uncovered one significant flaw in the model. Learners who already understand certain concepts should not be expected to explore these concepts as thoroughly as someone with less knowledge. In fact, observations of the subjects who appear to be able to self-monitor show that most of these subjects often explore only concepts they do not completely understand. The first version of the Student Model had little evidence on the learner's knowledge with which to gauge the sufficiency of the learner's exploration. Since the Machine and Plot units do not test a learner's knowledge of relevant concepts directly, in these units the only actions that can provide evidence of the learner's knowledge are exploration actions. Thus, in our study, the Student Model interpreted signs of the knowledgeable learners' inactivity in certain exercises as poor exploratory behaviour and consequently poor knowledge of related concepts, causing the Coach to intervene when the learners tried to leave. These unnecessary interventions do not follow ACE's guiding principle of intervening only when necessary. In addition, they may be a cause of the lack of correlation between some of the log data with the post-test scores. For instance, some knowledgeable learners did follow the Coach's advice and stayed in the current exercise. Also, because the unnecessary interventions were the result of incorrect low probabilities, and often the knowledgeable students who stayed did not explore enough to increase them, it would be equally difficult to correlate these probabilities with improvements on the post-tests.

5.5.2. *Exploration of Relevant Exploration Cases*

Currently, the Student Model looks only at interface actions to determine if the learner has explored a particular exploration case and doesn't incorporate more sophisticated factors, such as the learner's ability to learn from those actions. This limitation is especially apparent in the Plot Unit. There were a few learners who performed several of the desired interface actions in this unit, but did not end up learning the underlying concepts, while others learned these concepts after minimal exploratory actions. For example, some learners experimented with several negative and positive slopes, but did not learn the difference between the two, while others caught on after experimenting with only one of each. This observation suggests that effective exploration is more than just performing the appropriate interface manipulations; the learners must also interpret and reason about the results of those actions. For example, in the Plot Unit it is possible that learners who did not learn from their actions were moving the function graph around, but did not look at the function equation, or looked but did not build a connection between the graph shape and the equation. Lacking the skills or motivation to self-explain seems to be a plausible cause for this ineffective behaviour, but ACE's Student Model currently has no way to monitor these factors.

5.5.3. *Other Limitations*

The evaluation also uncovered other limitations of the Student Model that do not indicate problems with its general design but contributed to reducing the model's accuracy.

The first of these limitations is that the model is missing a certain type of long-term assessment. As we described in section 4.1, the Student Model assesses exploratory behaviour at several levels of granularity, including exercises exploration, exercise-specific exploration cases, unit exploration, and categories of exploration cases appearing in multiple exercises. The Student Model does not, however, provide a long-term assessment of concepts that span a number of exercises and/or units but do not correspond to specific exploration cases. For example, a number of subjects thoroughly explored most of the exercises in the arrow and machine units with the exception of those targeting constant functions. This type of information, however, could not be obtained from the original network, since constant function exploration was not one of the relevant exploration cases, which, in these two units, corresponded to trying different categories of inputs. This limitation was not as apparent in the network for the Plot Unit, because in that unit, exploration of different types of functions did correspond to the relevant exploration categories (see fig. 10).

Second, the Student Model did not take into account the similarity of different exercises. If two exercises are similar the learner should not be expected to explore the second exercise as thoroughly.

Finally, the time nodes did not have their intended influence on the assessment of effective exploration. Almost all learners stayed in the exercises for

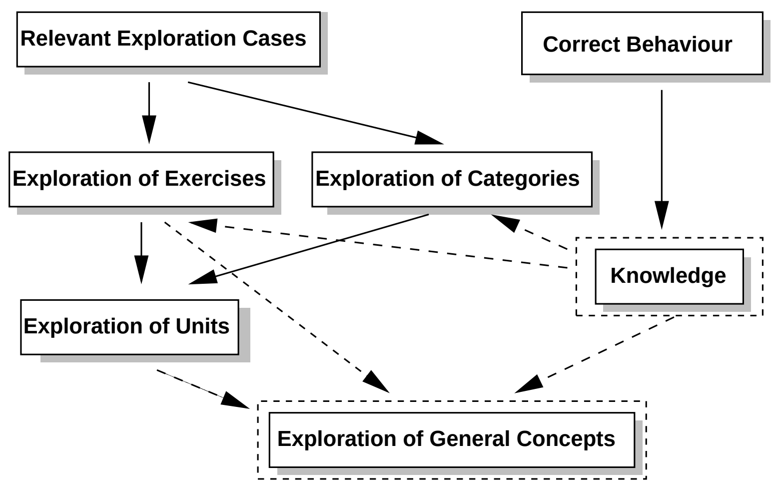

Figure 15. A high-Level description of the second version of the Student Model, with the changes marked by dashed lines

appropriate lengths of time, increasing the probability that they had effectively explored the exercises. As discussed in the previous section, however, it is unclear whether or not the time spent contributed to effective exploration. Occasionally, the time was spent staring at the interface, other times learners were performing many exploratory actions, yet they still failed to learn. We can provide only a subjective evaluation of the time nodes since the log files did not record how often they were set to True.

In the next section we discuss a second version of the Student Model that we built to address some of the limitations uncovered by the first evaluation.

6. Student Model - Version II

The major changes to the network structure (shown in fig. 15) include the incorporation of knowledge into the assessment of effective exploration, and the inclusion of another category of exploration nodes that assess how effectively the learner is exploring general concepts across exercises not covered by the exploration cases. This includes the relationship between the input and output for each of the different types of functions (constant, linear, power and polynomial). In addition, the similarity of exercises is now taken into account and the time nodes have been removed. We now discuss each of these changes in detail.

6.1. Knowledge Nodes

As discussed in section 5.5.1, the first version of the Student Model does not take the learner's knowledge into consideration when assessing the quality of learners' exploration. As a consequence, learners who already had high knowledge received unnecessary warnings to explore more. The proposed solution to this problem integrates learner knowledge into the exploration assessment so that a learner with high knowledge of a concept is not expected to explore that concept as thoroughly as a learner with low knowledge.

To achieve this behaviour in the second version of the Student Model, components of the Bayesian Networks that represent exploration nodes are now influenced by the learner's knowledge (see fig. 15). More specifically, most exploration nodes are now influenced by one or more knowledge nodes representing understanding of related concepts. These knowledge nodes have a different meaning than those in the first Student Model. In that model, knowledge nodes represent a general definition of concept understanding, comprising both performance in more traditional problem-solving tasks and performance in exploration activities. The second version of the Student Model separates the representation of student knowledge related to problem-solving activities from that of understanding gained through exploration. Thus, in the new model, knowledge nodes are updated only through non-exploratory actions (which includes evidence from exercises in the Arrow Unit, and when available, from pre-test activities) and directly contribute to the assessment of how much further understanding the learner can gain from exploratory actions.

Consequently, the semantics of the exploration nodes also need to change in the new model. A "True" value for an exploration node still represents the probability that the learner has effectively explored that item, but the definition of effective exploration is now broader than in the previous model. A learner has effectively explored a concept when the combination of the learner's exploration actions and existing knowledge has caused the learner to understand that concept. A low probability means that the learner should explore that concept more thoroughly to increase her understanding.

Figure 16 illustrates the influence of knowledge nodes in the new model. In this simplified example, "e_1" is an exercise in the Machine Unit involving a constant function. Thus, whether or not the learner has effectively explored the exercise is influenced by the learner's coverage of the relevant exploration cases ("e_1Case_1" and "e_1Case_2") and any existing knowledge of the concepts related to that exercise ("simpleArithmetic," "simpleSubstitution" and "constantFuncIO").

The CPTs for exploration nodes are defined so that if the student has high knowledge of the related concepts, the probability for that exploration node is also high. Otherwise, the probability of the learner having effectively

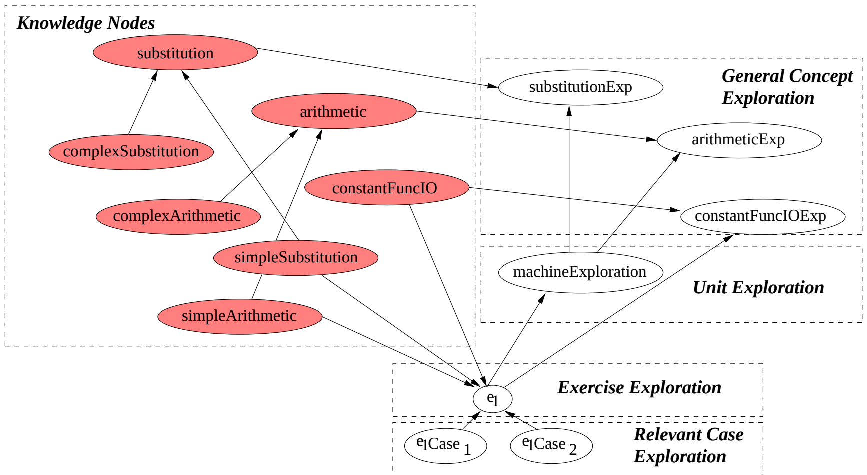

Figure 16. The influence of knowledge nodes (shaded nodes) in the second version of the Student Model

explored the item increases as the learner explores the relevant exploration cases, as in the first version of the model.

6.2. Additional General Exploration Concepts

The first version of the Student Model assesses learners' exploratory behaviour within particular exercises with the nodes representing individual exercises and their relevant exploration cases. It also assesses the learner's behaviour over a longer period of time with the nodes representing unit exploration and categories of exploration cases appearing in multiple exercises (e.g., small negative inputs and small positive inputs). However, as we describe in section 5.5.3, the study also uncovers the need to maintain a long-term assessment of concepts that span a number of exercises and/or units, but do not correspond to the specific exploration cases, as is the case for the input and output of different types of functions in the Machine and Arrow Units.

The model now has additional nodes that represent the learner's exploration of these additional concepts; they include the exploration of the input/output relationship for each type of function and how effectively the learner is exploring the concepts of arithmetic and substitution. These exploration concepts are not associated with particular exploration cases, but with exercises or units. Whether these exploration concepts are influenced by exercise exploration nodes or unit exploration nodes depends on the generality of the particular concept. For instance, as shown in the example portion

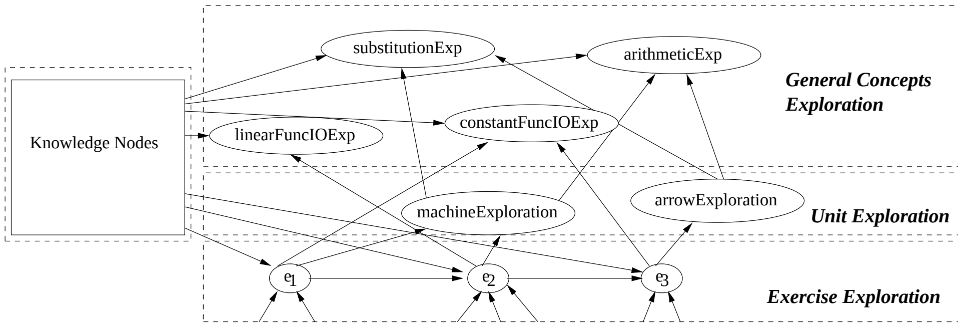

Figure 17. Example of nodes representing the exploration of general concepts in the second version of the Student Model

of the network in figure 17, the exploration of the input/output of constant functions ("constantFuncIOEx") relates to certain exercises in which constant functions input/output can be explored. Therefore, it is influenced by only those exercises ("e_1" and "e_3" in fig.17) and by knowledge of the related concepts. On the other hand, exploring the concept of substitution ('substitutionEx" in fig. 17) is covered by all exercises in certain units, and so is influenced by the related unit exploration nodes (e.g., "machineExploration" and "arrowExploration" in fig. 17).

The CPTs for the new concept nodes are designed (as discussed in the previous section) so that high knowledge of the concept results in a high probability of the "True" value regardless of the learner's exploration activities. If low knowledge exists, the more relevant exercises (or units) the learner has effectively explored, the higher the probability that the learner has effectively explored the related general concept.

6.3. Similarity of Exercises

The first version of the Student Model did not take the similarity of exercises into account in its assessment of effective exercise exploration. If a learner explores one exercise thoroughly and then chooses to explore a similar exercise, the learner should not be expected to explore this second exercise as thoroughly. To address this concern, arcs were added between exercise exploration nodes that have some degree of similarity (see "e_1", "e_2", and "e_3" in fig. 17). The strengths of the arcs are determined by similarity scores, which are computed based on similarity of both the units the exercises belong to and their function types. For instance, two linear functions in the Machine Unit have a higher similarity score than one linear function in the Machine Unit and another in the Arrow Unit.

6.4. Time Nodes

The final change to the new version of the Student Model consists of removing the time nodes that, in the previous model, contribute to the assessment of exercise exploration. How well learners explored exercises did not seem to depend on time spent in each exercise but on how long they spent reasoning about their actions and how long they spent exploring each individual exploration case. Thus, for the time being, the time nodes have been removed from the network and will be added again as part of the future research that incorporates a more sophisticated view of what it means to explore a relevant exploration case effectively.

6.5. Conditional Probability Tables

Other than those discussed in sections 6.1 and 6.2, no major changes were made to the CPTs as a result of the first evaluation. The problems uncovered in the first evaluation did not appear to be related to the CPTs, but rather to the model's structure. Some of the CPTs were tuned slightly, but did not deviate from their original design. Exploration is still assessed in an incremental manner. As learners explore more relevant items, the probability that they have explored effectively increases.

7. Version II - Evaluation

7.1. Study Goal

In this second evaluation, the new Student Model is evaluated with a small group of subjects. Before going to a full scale evaluation, we wanted to investigate how the major changes in the model influence the subjects' interactions with the system. A larger, more complete evaluation is needed, but will be left for future work. Ideally, the changes we made should have the following effects. First, the Coach was giving unnecessary warnings to high ability learners (as we discussed in section 6.1) to explore further. Both the new influences of the knowledge nodes and the addition of the exercise similarity should contribute to lowering the number of unnecessary warnings. Thus, the new model should result in the Coach intervening less frequently with the high ability learners, without causing the Coach to ignore the learners who are experiencing difficulty. Second, the removal of the time nodes might reduce some of the model's tendency to overestimate exploratory behaviour. Third, the addition of the new general exploration concepts described in section 6.2 should allow the model to generate appropriate assessments for learners who continually fail to explore general concepts, such as the input and output of specific function types.

Table III. Unnecessary warnings

	Version 1	Version 2
# Subjects	14	5
# Unnecessary Warnings	62	2
Total # warnings	163	42
% of warnings that were unnecessary	38%	5%
Average # of warnings per person	4.4	0.4

7.2. Study Participants

The subject pool for this follow-up study consists of five participants. As in the previous study, the subjects are first-year university students who have not taken a university math course.

7.3. Experiment Design

The second study uses essentially the same experimental design as the first study (see sec. 5.2). The only difference is that we use the results from the pre-tests to set the values of the knowledge nodes before the subjects start using ACE, because the second version of the Student Model performs better if it has some initial indication of the learners' knowledge.

7.4. Results

The log files of the learners' interactions with ACE are analyzed to see how the changes made to the model influence the number of navigation hints generated by the Coach. We count the number of times the Coach asks the learner to explore an exercise further despite clear indications from the pre-tests that the learner understands the concepts associated with the exercise. We refer to this event count as the number of *unnecessary warnings*. Since some of the changes are intended to reduce these unnecessary interruptions without the model becoming too lenient, we also count the number of times the Coach allows the learner to leave an exercise without warning, even though the learner has not learned the associated concepts (as indicated by the post-test). We refer to this event count as the number of *premature passes*. Since the only modification that addresses the problem of overestimating the learners' exercise exploration is the elimination of the time nodes, we do not expect the count of premature passes to change significantly from the previous experiment, but we want to at least make sure that it does not worsen.

Table III shows that the new Student Model does, in fact, result in a sizable reduction in the number of unnecessary warnings. While with the pre-

Table IV. Premature passes

	Version 1	Version 2
# Subjects	14	5
# Premature passes	6	5
Total # exercises to be assessed	154	55
% of premature passes	4%	9%
Average # of premature passes per person	0.4	1

Table V. Hint seeking

	Version 1	Version 2
Average per person	5.07	4.2
Standard Deviation	3.41	3.35

vious version of the model 38% of the Coach's warnings are unnecessary, the precentage drops to 5% with the new model. This is an important reduction, since with the previous version, learners received an average of 4.4 unnecessary interuptions per session, while they received an average of less than one with the new model. As shown in table IV, there is a small increase in the number of premature passes (from 4% to 9%), indicating that, as expected, the second model still overestimates the learners' exploratory behaviour: when learners perform a large number of exploratory interface actions, the model assesses this to be good exploratory behaviour, even though some of these learners do not learn from these actions.

The results of this study provide further evidence that the model needs to include the capability of assessing the learner's ability to self-explain the phenomena observed in the environment to accurately assess effective exploration. The elimination of the time nodes, the incorporation of exercise similarity, and the possibly incorrect assessment of the subjects' prior knowledge also may have caused part of the increase in premature passes, since all of these changes may have resulted in the model becoming more lenient. Finally, the study did not provide an evaluation of the change involving the addition of the general exploration concept nodes, since none of the participants in this study chose to ignore any particular high-level concepts, as was the case in the first study.

Since one of the encouraging results from the first evaluation is that the number of hints requested by the learners is positively correlated with post-test improvements, table V shows how the new model affects the rate of hint seeking. While table V shows the rate decreasing, the sample size is too small to draw conclusions on aspects of ACE usage that are influenced by both the behaviour of the system and individual learner characteristics.

Hint seeking is an example of such an aspect, as some learners have a general tendency not to ask for help (Aleven and Koedinger, 2000).

8. Discussion

The results of the two evaluations provide encouraging support for the framework of ACE's Student Model. The fact that the Student Model continues to over-estimate effective exploratory behaviour, however, means that improvements still need to be made. First of all, to fully assess effective exploration, the model needs to perform a more sophisticated assessment of what it means for a learner to explore a relevant exploration case effectively. Currently, the model bases this assessment only on a combination of the learner's interface actions and any explicit evidence of the learner's knowledge. As our evaluations have shown, however, some students perform the right set of actions, yet still fail to learn.

Our observations lead us to believe that the model needs to incorporate additional factors to allow it to better assess whether or not the learner's actions are a sign of good exploration. In particular, whether or not learners are self-explaining the outcomes of their exploratory actions appears to be a key component of the ability to learn from exploratory actions. Support for this hypothesis comes from our observations of one learner in particular who, in the Plot Unit, moved the function graph rapidly around the screen, without pausing long enough to permit any kind of self-explanation of how his movements were updating the function equation. According to the post-test, this learner failed to learn the concepts targeted by this unit.

Factors that the model could use to assess the learner's self-explanation behaviour include the amount of time spent reasoning about each case (as done in Conati et al. (2002)), whether or not the learner is observing the results of the experiments, and the learner's tendency to spontaneously self-explain. Consider, for example, a learner who is continually altering a graph in the Plot Unit. The model could be more confident that this is a sign of good exploration if the learner looks at both the graph and the equation as they change, and leaves the graph at the key positions for long enough to self-explain the correspondence with the current equation. Knowing a priori that the learner has a tendency to self-explain (most learners do not, as shown by many studies summarized in Chi (2000)) could further increase the model's confidence that the learner's behaviours are conducive to learning. Otherwise, it is possible that the learner is simply dragging the graph around the screen for entertainment purposes.

We didn't address these issues after the first evaluation because ACE's current interface provides little evidence of the factors indicating self-explanation. For instance, we could potentially measure time spent on each case by considering the interval between the action initiating two consecutive cases, but we would not be able to tell if, during that interval, the learner is actually attending to the current case. The use of eye-tracking could help widen the bandwidth of information coming from the interface, since knowing where users are focusing their attention provides a more accurate assessment of how much time they spend on each case and whether or not they pay attention to the outcomes of their exploration.

What about situations where the users are inactive? Currently the model can detect these situations, but does not know the cause. The current model can highlight which concepts need further exploration, but is limited in its ability to diagnose why these concepts haven't been explored. There are a number of possible causes of passivity in open learning environments, including the learner's motivation, personality traits, learner's knowledge of exploration strategies, as well as domain knowledge. Incorporating these factors, along with the aforementioned factors affecting self-explanation behaviour, is expected to improve both the model's assessment and its diagnostic capabilities, thus allowing the implementation of coaching strategies that can directly address the causes of poor exploration.

Other directions of future work include opening up the Student Model to the learners and adding mechanisms for them to indicate the reasoning behind their actions. Both of these extensions would allow learners to reflect more on their own exploration process as well as self-explain the results. Opening up the Student Model would encourage learners to reflect on how thoroughly they have been exploring and which exercises or concepts they have tended to ignore. Mechanisms for learners to explain their exploratory actions, including why they have chosen to explore certain cases and not others, would not only promote reflection and self-explanation, but would also provide an additional source of information for the Student Model. However, without natural language processing capabilities, special care will have to be taken so that these mechanisms do not cause the interface to become overly restricted.

9. Related Work

9.1. User Models for Open Learning Environments

Few other researchers have tried to model user performance in open learning environments. The user model in Veermans and van Joolingen (1998) focuses on a sub-skill of exploration by assessing a user's ability to either confirm or reject a hypothesis. Unlike the ACE's Student Model, this approach assumes that users are active enough to generate hypotheses and perform the experimentation necessary to verify these hypotheses. Since the system waits for the learner to submit work to be evaluated before providing feedback, it is unable to model learners who do not submit hypotheses. In addition, the system cannot support users who have difficulty covering the exploration space, where effective coverage involves exploring a wide range of hypotheses.

In Smithtown (Shute and Glaser, 1990), users can perform experiments involving variables in the domain of microeconomics. In the exploration phase, the user gathers information and makes observations about variables in the particular example economy. In the subsequent experimentation phase, the user forms hypotheses, manipulates variables and draws conclusions pertaining to the laws of supply and demand. In this latter phase, the system supports the user's exploration by monitoring the user's behaviour and interrupting if the user consistently violates one of the rules of good experimentation (e.g., manipulates more than one variable at a time). The user model in this situation, however, is essentially a set of counters that record the number of times a user has violated the rules in the knowledge base. Like the previous approach, this type of modelling relies on the assumption that learners are active experimenters. Furthermore, unlike ACE, Smithtown does not assist the learner in choosing which experiments to perform.

The user model in the ALI system (D'Souza et al., 2001) shares many aspects in common with the ACE model. Although the techniques are different (ALI has a overlay model while the ACE student model uses Bayesian Networks), the model in ALI contains information on whether or not the learners encounter all of the important domain concepts, in addition to whether or not they are able to understand them through their experimentation. The main difference, however, is that instead of making inferences through the learner's behaviour in the environment, ALI generates its assessment based on explicit evidence obtained by engaging the learner in dialogues and having the learner complete small quizzes. Thus, ALI departs from the open-ended and learner-directed style of interaction typical of pure open learning environments, which ACE's Student Model aims to preserve.

Other user models in open learning environments have focused less on assessing the user's exploration of the available material and more on assessing the user features that permit the environment to preemptively guide the user's exploration in a tailored manner. For example, the user model in Hypadapter (Hohl et al., 1996), a hypertext system, includes information on user characteristics such as curiosity level and material presentation preferences, which are obtained from a questionnaire. The model also assesses users' knowledge of topics as they explore, although it relies on the strong assumption that following a link for a topic results in increased knowledge of the topic. The model is used by the system to restrict the amount of information and links available for the user to explore. The user model in ELM-ART (Weber and Brusilovsky, 2001), an adaptive hypermedia system designed to support learning LISP programming, also tries to assess the users' evolving knowledge as they explore the available lessons and exercises. The authors, however, do not agree with the assumption that knowledge can be inferred from followed links, and instead assess knowledge through more explicit activities and tests.

9.2. Probabilistic User Models

Bayesian Networks have become a popular formalism in user modelling because of their ability to model uncertainty (Jameson, 1996). They have been used extensively for knowledge assessment (e.g., Mayo and Mitrovic, 2000; Millan and de-la Cruz, 2002; Mislevy and Gitomer, 1996; VanLehn and Martin, 1998), plan recognition (e.g., Albrecht et al., 1998; Charniak and Goldman, 1993; Horvitz et al., 1998; Huber et al., 1994) or both (e.g, Conati et al., 2002; Conati and VanLehn, 1997; Murray and VanLehn, 2000). There have also been some interesting applications of Bayesian Networks that have extended beyond knowledge assessment and plan recognition, including a model to assess how effectively the learner is self-explaining (Conati and VanLehn, 2001), models of collaborative behaviour (Singley et al., 1999; Zapata-Rivera and Greer, 2001) and models of users' emotions (Conati and Zhou, 2002; Ball and Breese, 1999). A model of attention is proposed in Horvitz et al. (1999), and a model that detects when a user is experiencing cognitive overload is discussed in Muller et al. (2001).

Recently, systems have also begun to use an extension to Bayesian Networks known as Dynamic Decisions Networks (Russell and Norvig, 1995), a framework that supports rational decision making. Examples of systems that have employed this decision-theoretic approach to tailored interaction can be found in Murray and VanLehn (2000), Mayo and Mitrovic (2001), Horvitz and Paek (2001), and in Jameson et al. (2001).

10. Conclusions

In this paper, we have presented research aimed at creating a student model that can assess the effectiveness of a learner's exploratory behaviour in an open learning environment. This work addresses a substantial limitation of open learning environments: their effectiveness for learning strongly depends on learner-specific features that influence the learners' ability to explore adequately. The Student Model we present is designed to support the provision of tailored feedback on a learner's exploration process in ACE, an open learning environment for mathematical functions. The aim is to make the open learning environment beneficial even for those learners who do not already possess the skills relevant to effective exploration. The model is also designed to monitor the learners' actions in the environment unobtrusively, in order to maintain the unrestricted nature of the interaction that is one of the key assets of open learning environments.

Our model relies on the probabilistic framework of Bayesian Networks to handle the large amount of uncertainty involved in monitoring the learner's exploratory behaviour. This uncertainty stems from the freedom the environment gives the learner, which makes it difficult for the model to assess the reasons behind a learner's interface actions and whether or not these actions contribute to exploration that is truly effective. The model performs its assessment by relying on a Knowledge Base that defines the relevant exploration cases within each activity that ACE supports and by monitoring the learner's exploration of these cases. Nodes in the Bayesian Networks that correspond to these relevant exploration cases form the basis of the rest of the Student Model's assessment, including the effectiveness of the learner's exploration of exercises, of units and of concepts. Exploration nodes are influenced by a set of related knowledge nodes, incorporating explicit evidence of the learner's knowledge into the assessment of effective exploration.

One of the major challenges in building the model was the lack of a clear definition of what constitutes effective exploratory behaviour. We have described how the model's representation of effective exploration has evolved through an iterative design process. The first version of the model followed the assumption that effective exploration can be assessed based solely on interface actions. But the empirical study we performed on the model showed that monitoring sheer activity level is most likely not sufficient to accurately assess effective exploration. The learners' existing knowledge and the amount of reasoning they do as they perform the exploratory actions also appeared to be crucial elements for assessing exploratory behaviour and for diagnosing the causes of poor exploration. Thus, in the second model we present here, we made learner knowledge one of the predictors of effective exploration. We discussed what additional factors (e.g., the learner's self-explanation behaviour) could be added to improve the assessment of learner reasoning, but we have yet to perform the actual modifications to the model.

Since our model uses Bayesian Networks, an alternative to the iterative design and evaluation approach would be to use machine learning techniques. This would also involve building and evaluating an initial version of the model. But, rather than using human judgment to refine the model, machine learning algorithms would be run on traces of the learners' interactions with ACE to learn the network structure and/or the CPTs. While this is an attractive alternative, since it removes a great deal of the subjectivity, it requires a large amount of data, and dealing with the additional challenge due to the fact that our model has many hidden variables.

The empirical study also revealed that relevant cases could not form the basis of assessing all types of exploration. There are certain concepts that appear across multiple exercises and units, but that do not correspond directly to relevant exploration cases. These types of nodes have been added to the second version. The first version did not take exercise similarity into account, which has been addressed in the second model. Finally, the evaluation demonstrated that the way in which time spent exploring was used to assess effective exploration was not sufficient, thus, the time nodes were removed.

In addition to giving insight into how to improve the model, the evaluation provided initial indications of the Student Model's accuracy and the benefits of having tailored support. As far as accuracy is concerned, there are two statistically significant correlations with improvement on post-tests. The first is that the more hints the learners access, the more they improve. Since the content of the hints was selected based on the assessment of the Student Model, this result is an initial indication that the Student Model generates an accurate assessment of which concepts need further exploration. The second is that the number of exercises the Student Model assessed as being effectively explored is positively correlated with the learners' improvements from pre-tests to post-tests, also an indication of the model's assessment accuracy.

The evaluation also provides support for the idea of including a model that can assess effective exploration in an open learning environment. Observations from the study illustrate that it is important for an open learning environment to identify learners who are not actively exploring and to give them specific suggestions targeted at improving their exploration. These observations show that some learners are inactive despite having low knowledge of the concepts available to be explored, and others need guidance to completely cover the exploration space effectively. Without such guidance they are unable to learn important domain concepts.

The Student Model that we presented here was specifically tailored to the ACE environment, but its high-level design has the potential to generalize to other open learning environments. Whether or not the approach can be generalized is highly dependent on the environments' definition of "good exploration". If the definition is similar to the one discussed here, then the high-level structure of the model can be applied to any open learning environment with a set of activities and exploration concepts of interest within those activities. This generalizable high-level structure includes the representation of exploration using the classes of nodes described here, as well as the high-level dependencies between exploration and knowledge.

Assessing effective exploratory behaviour has applications beyond educational settings. It is applicable in any system that is rich with features or information, but relies on the user's ability to explore. Examples include the World Wide Web and large word processing systems. Since some users have difficulty coping with the lack of direction that these systems provide, modelling effective exploration and providing tailored feedback is important to their usefulness. We believe that the research we have presented in this paper helps to highlight both how to approach the solution to the problem and future directions that this work can take.

Acknowledgments

The work presented here is supported by an NSERC PGS B post-graduate scholarship. We would like to thank Kasia Muldner and Michael Huggett for their contributions to the ACE environment. Kasia Muldner designed and implemented the Coach and helped organize and conduct the first evaluation. Michael Huggett worked on the interface and was also involved in the first evaluation. In addition, we would like to thank Giuseppe Carenini and the anonymous reviewers for their many helpful comments and suggestions.

References

Albrecht, D., I. Zukerman, and A. Nicholson: 1998, 'Bayesian Models for Keyhole Plan Recognition in an Adventure Game'. *User Modeling and User-Adapted Interaction* **8**, 5–47.

Aleven, V. and K. Koedinger: 2000, 'Limitations of Student Control: Do Students Know When They Need Help?'. In: *Proceedings of the Fifth International Conference on Intelligent Tutoring Systems (ITS 2000)*. Montreal, Canada, pp. 292 – 303.

Anderson, J., A. Corbett, K. Koedinger, and R. Pelletier: 1995, 'Cognitive Tutors: Lessons Learned'. *Journal of the Learning Sciences* **4**(2), 167–207.

Ball, G. and J. Breese: 1999, 'Modeling the Emotional State of Computer Users'. In: *Workshop on Attitude, Personality and Emotion in User-Adapted Interaction, UM'99*. Banff, Canada.

Bunt, A.: 2001, 'On Creating a Student Model to Assess Effective Exploratory Behaviour in an Open Learning Environment'. Master's thesis, University of British Columbia.

Bunt, A. and C. Conati: 2002, 'Assessing Effective Exploration in Open Learning Environments using Bayesian Networks'. In: *Proceedings of ITS 2002, 6th International Conference on Intelligent Tutoring Systems*. Biarritz, France, pp. 698–707.

Bunt, A., C. Conati, M. Huggett, and K. Muldner: 2001, 'On Improving the Effectiveness of Open Learning Environments Through Tailored Support for Exploration'. In: *Proceedings of AIED 2001, 10th International Conference on Artificial Intelligence in Education*. San Antonio, TX, pp. 365–376.

Charniak, E. and R. Goldman: 1993, 'A Bayesian Model of Plan Recognition'. *Artificial Intelligence* **64**(1), 53–79.

Chi, M.: 2000, 'Self-Explaining Expository Texts: The Dual Processes of Generating Inferences and Repairing Mental Models'. In: R. Glaser (ed.): *Advances in Instructional Psychology*. Mahwah, NJ: Lawrence Erlbaum Associates, pp. 161–238.

Conati, C., A. Gertner, and K. VanLehn: 2002, 'Using Bayesian Networks to Manage Uncertainty in Student Modeling'. *User Modeling and User-Adaptive Interaction* **12**(4).

Conati, C. and K. VanLehn: 1996, 'POLA: a Student Modeling Framework for Probabilistic On-Line Assessment of Problem Solving Performance'. In: *Proceedings of UM-96, Fifth International Conference on User Modeling*. Kailua-Kona, Hawaii, pp. 75–82.

Conati, C. and K. VanLehn: 1997, 'Probabilistic Plan Recognition for Cognitive Apprenticeship'. In: *Proceedings of the 18th Annual Conference of the Cognitive Science Society*. San Diego, CA.

Conati, C. and K. VanLehn: 2000, 'Toward Computer-Based Support of Meta-Cognitive Skills: a Computational Framework to Coach Self-Explanation'. *International Journal of Artificial Intelligent in Education* **11**.

Conati, C. and K. VanLehn: 2001, 'Providing Adaptive Support to the Understanding of Instructional Material'. In: *Proceedings of the International Conference on Intelligent User Interfaces*. Santa Fe, NM, pp. 41–47.

Conati, C. and X. Zhou: 2002, 'Modeling Students' Emotions from Cognitive Appraisal in Educational Games'. In: *Proceedings of ITS 2002, 6th International Conference on Intelligent Tutoring Systems*. Biarritz, France, pp. 944–954.

de Jong, T. and W. van Joolingen: 1998, 'Scientific Discovery Learning With Computer Simulations of Conceptual Domains'. *Review of Educational Research* **68**(2), 179–201.

D'Souza, A., J. Rickel, B. Herreros, and W. Johnson: 2001, 'An Automated Lab Instructor for Simulated Science Experiments'. In: *Proceedings of AIED 2001, 10th International Conference on Artificial Intelligence in Education*. San Antonio, TX, pp. 65–75.

Hohl, H., H. Bocker, and R. Gunzenhauser: 1996, 'Hypadapter: An Adaptive Hypertext System for Exploratory Learning and Programming'. *User Modeling and User-Adapted Interaction* **6**(2-3), 131–156.

Horvitz, E., D. Herckerman, D. Hovel, and R. Rommelse: 1998, 'The Lumiere Project: Bayesian User Modeling for Inferring the Goals and Needs of Software Users'. In: *Proceedings of the 14th Conference on Uncertainty in Artificial Intelligence*. Madison, WI, pp. 256–265.

Horvitz, E., A. Jacobs, and D. Hovel: 1999, 'Attention-Sensitive Alerting'. In: *Proceedings of UAI'99, Conference on Uncertainty and Artificial Intelligence*. Stockholm, Sweden, pp. 305–313.

Horvitz, E. and T. Paek: 2001, 'Harnessing Models of Users' Goals to Mediate Clarification Dialog in Spoken Language Systems'. In: *UM 2001, 8th International Conference on User Modeling*. Sonthofen, Germany, pp. 3–13.

Huber, M., E. Durfee, and M. Wellman: 1994, 'The Automated Mapping of Plans for Plan Recognition'. In: *Proceedings of the 10th Conference on Uncertainty in Artificial Intelligence*. Seattle, WA, pp. 344–351.

Jameson, A.: 1996, 'Numerical Uncertainty Management in User and Student Modeling: An Overview of Systems and Issues'. *User Modeling and User-Adapted Interaction* **5**(3-4), 193–251.

Jameson, A., D. Großmann-Hutter, L. March, R. Rummer, T. Bohnenberger, and F. Wittig: 2001, 'When Actions Have Consequences: Empirically Based Decision Making for Intelligent User Interfaces'. *Knowledge-Based Systems* **14**, 75–92.

Mayo, M. and A. Mitrovic: 2000, 'Using a Probabilistic Student Model to Control Problem Difficulty'. In: *Proceedings of the Fifth International Conference on Intelligent Tutoring Systems (ITS 2000)*. Montreal, Canada, pp. 524–533.

Mayo, M. and A. Mitrovic: 2001, 'Optimising ITS Behaviour with Bayesian Networks and Decision Theory'. *International Journal of Artificial Intelligence in Education* **12**, 124–153.

Millan, E. and J. P. de-la Cruz: 2002, 'A Bayesian Diagnostic Algorithm for Student Modeling and its Evaluation'. *User Modeling and User-Adapted Interaction* **12**(2-3), 281–330.

Mislevy, R. and D. Gitomer: 1996, 'The Role of Probability-Based Inference in an Intelligent Tutoring System'. *User Modeling and User-Adapted Interaction* **5**(3-4), 253–282.

Muller, C., B. Großmann-Hutter, A. Jameson, R. Rummer, and F. Wittig: 2001, 'Recognizing Time Pressure and Cognitive Load on the Basis of Speech: An Experimental Study'. In: *UM 2001, 8th International Conference on User Modeling*. Sonthofen, Germany, pp. 24–33.

Murray, R. and K. VanLehn: 2000, 'DT Tutor: A Decision-Theoretic, Dynamic Approach for Optimal Selection of Tutorial Actions'. In: *Proceedings of the Fifth International Conference on Intelligent Tutoring Systems (ITS 2000)*. Montreal, Canada, pp. 153–162.

Njoo, M. and T. de Jong: 1993, 'Exploratory learning with a Computer Simulation for Control Theory: Learning Processes and Instructional Support'. *Journal of Research in Science Teaching* **30**(8), 821–844.

Pearl, J.: 1988, *Probabilistic Reasoning in Intelligent Systems*. Los Altos, CA: Morgan Kaufmann.

Recker, M. M. and P. Pirolli: 1992, 'Student Strategies for Learning Programming from a Computational Environment'. In: *Proceedings of the Second International Conference on Intelligent Tutoring Systems*. Montreal, Canada, pp. 382–394.

Reiser, B., W. Copen, M. Ranney, A. Hamid, and D. Kimberg: 1994, 'Cognitive and Motivational Consequences of Tutoring and Discovery Learning'. Technical report, The Institute for the Learning Sciences.

Russell, S. and P. Norvig: 1995, *Artificial Intelligence: A Modern Approach*. Morgan-Kaufman.

Shute, V.: 1993, 'A Comparison of Learning Environments: All That Glitters...'. In: S. P. Lajoie and S. J. Derry (eds.): *Computers as Cognitive Tools*. Hillsdale, NJ: Lawrence Erlbaum Associates, pp. 47–74.

Shute, V. and R. Glaser: 1990, 'A Large-Scale Evaluation of an Intelligent Discovery World: Smithtown'. *Interactive Learning Environments* **1**, 55–77.

Singley, M., P. Fairweather, and S. Swerling: 1999, 'Team Tutoring Systems: Reifying Roles in Problem Solving'. In: *CSCL'99*. Stanford, CA, pp. 538–548.

van Joolingen, W.: 1999, 'Cognitive Tools for Discovery Learning'. *International Journal of Artificial Intelligence and Education* **10**, 385–397.

van Joolingen, W.: 2000, 'Designing for Collaborative Discovery Learning'. In: *Proceedings of the Fifth International Conference on Intelligent Tutoring Systems (ITS 2000)*. Montreal, Canada, pp. 202–211.

van Joolingen, W. and T. de Jong: 1991, 'Supporting Hypothesis Generation by Learners Exploring an Interactive Computer Simulation'. *Instructional Science* **20**, 389–404.

VanLehn, K.: 1988, 'Student Modeling'. In: M. C. Polson and J. J. Richardson (eds.): *Foundations of Intelligent Tutoring Systems*. Hillsdale, NJ: Erlbaum, pp. 55–78.

VanLehn, K.: 1996, 'Conceptual and Meta Learning during Coached Problem Solving'. In: *Proceedings of Intelligent Tutoring Systems, 3rd International Conference (ITS'96)*. Montreal, Canada, pp. 29–41.

VanLehn, K. and J. Martin: 1998, 'Evaluation of an Assessment System Based on Bayesian Student Modeling'. *International Journal of Artificial Intelligence in Education* **12**, 179–221.

Veermans, K., T. de Jong, and W. van Joolingen: 2000, 'Promoting Self-Directed Learning in Simulation-Based Discovery Learning Environments Through Intelligent Support'. *Interactive Learning Environments* **8**(3), 229–255.

Veermans, K. and W. van Joolingen: 1998, 'Using Induction to Generate Feedback in Simulation Based Discovery Learning Environments'. In: *Proceedings of Intelligent Tutoring Systems, 4th International Conference (ITS'98)*. San Antonio, TX, pp. 196–205.

Weber, G. and P. Brusilovsky: 2001, 'ELM-ART: An Adaptive Versatile System for Web-Based Instruction'. *International Journal of Artificial Intelligence in Education* **10**, to appear.

Zapata-Rivera, J. and J. Greer: 2001, 'SModel Server: Student Modelling in Distributed Multi-Agent Tutoring Systems'. In: *Proceedings of AIED 2001, 10th International Conference on Artificial Intelligence in Education*. San Antonio, TX, pp. 446–455.

Author's vitae

Andrea Bunt

Department of Computer Science, University of British Columbia, Vancouver, BC, V6T1Z4, Canada

Andrea Bunt is a Ph.D. student in the Computer Science Department at the University of British Columbia. She received her B.Sc (Honours) in Computing and Information Science from Queen's University in 1999 and her M.Sc. in Computer Science from the University of British Columbia in 2001. Her interests lie in the areas of artificial intelligence, intelligent learning environments, user modeling and adaptive interfaces.

Cristina Conati

Department of Computer Science, University of British Columbia, Vancouver, BC, V6T1Z4, Canada

Dr. Conati is an Assistant Professor of Computer Science at the University of British Columbia. She received her "Laurea" degree (M.Sc. equivalent) in Computer Science at the University of Milan, and her M.Sc. and Ph.D. in Artificial Intelligence at the University of Pittsburgh. Her research interests include user modeling, reasoning under uncertainty, adaptive interfaces and intelligent learning environments. Her current research focuses on extending the range of user's features that can be captured in a user model - from purely cognitive features (knowledge, goals, preferences), to meta-cognitive skills (such as learning capabilities), personality traits and emotional reactions, in order to widen the spectrum of information that an interactive system can use to adapt its behavior to a user's needs.