

# Assessing Effective Exploration in Open Learning Environments using Bayesian Networks

Andrea Bunt, Cristina Conati

Department of Computer Science, University of British Columbia  
201-2366 Main Mall, Vancouver, B.C, V6T 1Z4  
bunt@cs.ubc.ca, conati@cs.ubc.ca

**Abstract.** Open learning environments provide a large amount of freedom and control, which can be beneficial for students who are able to explore the environment effectively, but can also be problematic for those who are not. To address this problem, we have designed a student model that allows an open learning environment to provide the students with tailored feedback on the effectiveness of their exploration. The model, which uses Bayesian Networks, was created by an iterative design and evaluation process. The successive evaluations were used to improve the model and to provide initial support for its accuracy and usefulness.

## 1 Introduction

Open learning environments are computer-based educational systems that place less emphasis on supporting learning through explicit instruction and more on providing the learner with the opportunity to explore the instructional domain freely, acquiring knowledge of relevant concepts and skills in the process [4][7]. In theory, this type of active learning should enable students to acquire a deeper, more structured understanding of concepts in the domain [7]. Also, owing to the unguided nature of the interaction, the hope is that, in addition to skills in the target instructional domain, the learner can practice and acquire meta-cognitive skills associated with effective exploration [4].

Empirical evaluations, however, have shown that a student's ability to benefit from interacting with open learning environments depends on a number of student-specific features, including activity level [4][7], whether or not the student already possesses the meta-cognitive skills necessary to learn from exploration [7] and general academic achievement [6]. Students who are inactive or lack the necessary cognitive skills often fail to initiate enough meaningful experiments; they can have difficulty interpreting and generalizing the results of the experiments that they do initiate [7][8], thus incurring sub-optimal learning.

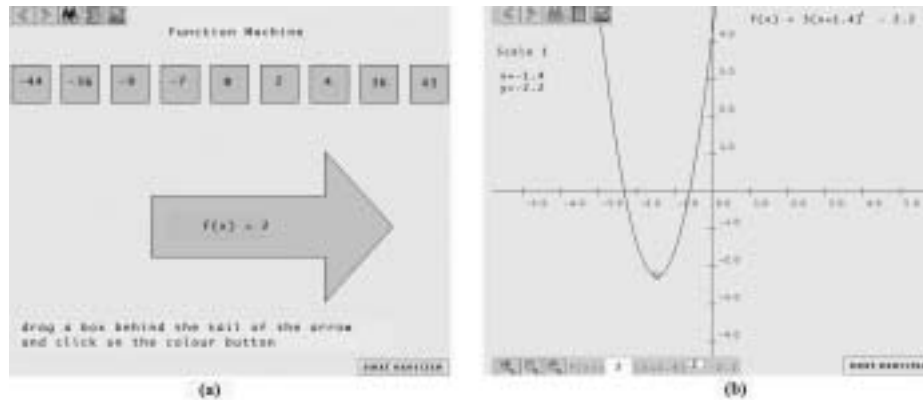
The above findings indicate that the effectiveness of open learning environments could be improved by providing real-time support for the exploration process, tailored to each student's individual needs. Such support should be provided only when necessary, to avoid interfering with the unrestricted nature of this learning activity. Therefore, having a student model that can detect when a

student is not benefitting from the exploration process is crucial. In this paper, we describe such a model. Our model relies on Bayesian Networks to assess the effectiveness of the student's exploration process and was built using an iterative design and evaluation process. Two successive user evaluations were conducted to define the structure of the model and to validate our approach. The model has been implemented in the Adaptive Coach for Exploration (ACE) [2]. Using the Student Model's assessment, ACE provides tailored hints to guide and improve the students' exploration of mathematical functions.

Modelling students' exploration presents unique challenges, for two main reasons. First, in more structured educational activities, such as problem solving and question answering, there is usually a definition of correct behaviour, which allows this behaviour to be represented and recognized in a formal model. In contrast, in open learning environments there is no clear understanding of what constitutes successful exploration in general. Second, it is hard to obtain reliable information on the student's exploratory behaviour. The amount and quality of information available to a user model to perform its assessment is referred to as the bandwidth issue [9]. The less explicit information on the user's relevant traits or behaviours the model is able to obtain, the higher the uncertainty in the modelling process. The bandwidth problem is especially difficult for student modelling in open learning environments. Both exploratory behaviour and related meta-cognitive skills necessary for effective learning are not easily observable unless the environment's interface forces students to make them explicit. However, forcing students to articulate their exploration steps 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 into the modelling task.

Because of these challenges, there has been little work on how to monitor and assess student behaviour in open learning environments. ALI's student model [3] indicates whether or not the student has encountered the important concepts in the target environment and understood them through experimentation. This model, however, deals with quite high bandwidth information obtained directly by both engaging students in tutorial dialogs and requiring them to take quizzes. Other systems tackle a restricted version of the problem of modelling exploration. A student model that can assess a student's ability to either confirm or reject hypotheses is described in [10]. Smithtown [7] tracks when a student violates rules of good experimentation, such as manipulating more than one variable at a time. Neither model, however, addresses the issue that some students are inactive in open learning environments and can have difficulty initiating a set of experiments that effectively covers the exploration space.

The rest of the paper describes how we tackle the challenges of modelling exploratory behaviour in ACE's student model. We formalize effective exploration behaviour by designing the student model iteratively using the results of an evaluation with human subjects. We address the uncertainty owing to the low bandwidth problem using Bayesian Networks.



**Fig. 1.** The Machine Unit (a) and the Plot Unit (b).

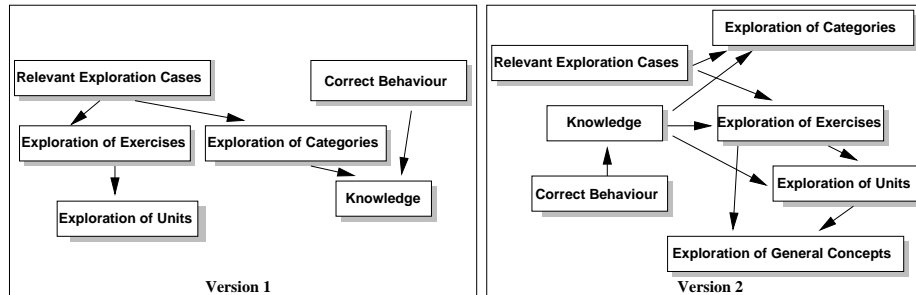
## 2 The ACE Open Learning Environment

ACE [2] is an intelligent open learning environment for the domain of mathematical functions. ACE's activities are divided into units and exercises. Units are collections of exercises whose material is presented with a common theme and mode of interaction. Exercises within the units differ in function type and equation. Currently, ACE supports only different types of polynomial functions.

Figure 1 shows the main interaction window for two of ACE's units: the *Machine Unit* and the *Plot Unit*. ACE also has a third unit, the *Arrow Unit*, not displayed for lack of space. We have also omitted the help pages and the feedback panel, which normally appear to the right of and below the main window, respectively. The *Machine Unit* (fig. 1(a)) and the *Arrow Unit* allow the student to explore the relationship between an input and the output of a given function. In the Machine Unit, the exploration consists of 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(a)), which then computes the corresponding output. The Arrow Unit allows the student to match a number of inputs with the correct outputs and is the only activity within ACE that has a clear definition of correct and incorrect behaviour. In the *Plot Unit* (fig. 1(b)), the student can explore the relationship between the graph of a function and its equation, as well as graph properties, such as slopes and intercepts. The student can manipulate a graph either by dragging it around the screen (using the mouse) or by editing the equation box.

## 3 The Student Model

The Student Model aims to generate an assessment of the student's exploration of the ACE environment that the system can use to support the student's exploration. As the description of the ACE interface in section 2 shows, the student's



**Fig. 2.** A High-Level Description of the First and Second Versions of the Student Model

actions within the available activities have low bandwidth because, although they are easy to track, they provide limited information about the student’s reasoning. The Student Model uses Bayesian Networks to model and manage the resulting uncertainty[5].

One of the main challenges in using Bayesian Networks is to define the network’s structure to accurately represent the probabilistic dependencies among the variables of interest. In our model, this problem is exacerbated by the difficulty of defining correct exploratory behaviour (see sec. 2). Our approach was to use an iterative design process, which we describe in this section. We first built a version of the Student Model using our intuition of what constitutes effective exploration in ACE and evaluated the model in a formal study. We then used the study results to redesign the model and conducted an evaluation of the changes. By building and evaluating two versions of the model, we were able to gain valuable insight into (i) what factors contribute to effective exploration of the ACE environment and (ii) how to formalize these factors in the Student Model. Because of space limitations we are not able to describe all the details of this iterative process. Further information can be found in [1] and [2].

### 3.1 Version 1 of ACE’s Student Model

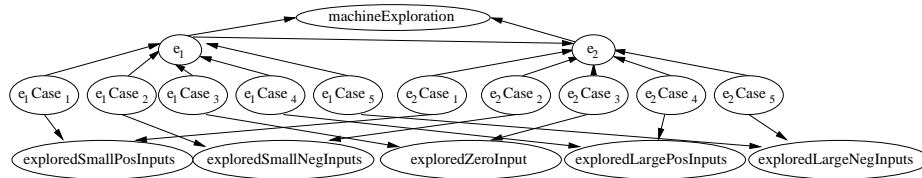
Figure 2 (Version 1) shows a high-level description of the types of nodes in the first version of the model’s Bayesian Network and the influences among them. There are two classes of nodes: exploration nodes and knowledge nodes. To assess exploration, the model uses the student’s coverage of the *relevant exploration cases*. Relevant exploration cases represent the salient function-related concepts that should be explored in each exercise to gain a thorough understanding of the target material. Exploratory behaviour is modelled at different levels of granularity, including the exploration of individual exercises (“Exploration of Exercises” in fig. 2), of groups of related exercises (“Exploration of Units” in fig. 2) and of exploration cases that appear across multiple exercises (“Exploration of Categories” in fig. 2). The set of relevant exploration cases for a particular function pre-

sented in the exercise. For example, in the Plot Unit, if the student is exploring a constant function, she should experiment with positioning the graph at both positive and negative intercepts, while a linear function would also require experimenting with different slopes. Exploration cases in the Machine and Arrow units involve different categories of inputs, such as small positive numbers and large negative numbers. All exploration nodes are binary variables where a True value represents the probability that the student has effectively explored the corresponding item (i.e., an exploration case, exercise, unit or category). Knowledge nodes (“Knowledge” in fig. 2) represent the knowledge of function related concepts. These nodes are updated using a combination of student actions that indicate exploration and any explicit evidence of their knowledge (“Correct Behaviour” in fig. 2). In this version of ACE, explicit evidence of knowledge is available only in the Arrow Unit. A True value for a knowledge node means that the student understands the related material. The Conditional Probability Tables (CPTs) in the Bayesian Network were constructed using our best estimates of the corresponding probabilistic dependencies, which were tested and refined through the evaluation described later.

Figure 3 shows a more detailed portion of the network representing the exploration of two exercises in the Machine Unit (nodes “ $e_1$ ” and “ $e_2$ ”). The exploration case nodes for each exercise (nodes “ $e_i$ Case $_i$ ” in fig. 3) influence both their corresponding exercise node and the exploration category that they belong to. For example, “ $e_1$ Case $_1$ ” is a relevant exploration case in exercise “ $e_1$ ” and is an instance of the exploration category consisting of small positive inputs (“exploredSmallPosInputs” in fig. 3).

Direct evidence of the students’ exploration is introduced into the network through the relevant exploration case nodes. The value of a relevant exploration case node is changed to True when the student performs interface actions that the system considers to be an indication of the student having explored that case. Each unit in ACE has a different interpretation of what this entails. For example, in the Machine Unit, a case is considered explored when the student drags an input to the “machine” that is an instance of that case. For instance, in figure 3, the node “ $e_1$ Case $_1$ ” would be set to true when the student drags a small positive input. In the Plot Unit, a case is considered explored when the student either drags and drops the graph to a position belonging to a particular case (e.g., drags the line to a location where it has a positive y-intercept) or edits the function equation to change the graph in a meaningful way (e.g., changes the equation so that the slope of the graph is negative).

The Student Model’s assessment is used by ACE’s coaching component [2] to support the student’s exploration in two ways. The first is through tailored hints, which the student can obtain on demand. The Coach determines the focus of a hint (i.e., what aspect of the current exercise should be explored further) by searching the Bayesian Network for a relevant exploration concept node that has a low probability. The second kind of support is provided when a student tries to move on to a new exercise. When this happens, the Coach examines the probability that the student has effectively explored the current exercise. If that



**Fig. 3.** A Portion of the Network Related to Exercise Exploration in the Machine Unit

probability is too low, the Coach generates a warning to explore the exercise better and to ask for a hint if needed.

The first version of ACE’s Student Model was evaluated in a formal study described in [2]. Subjects in the study were first-year university students who had not taken a university math course. Two of the statistically significant results from the study provided initial support for the Student Model’s accuracy and usefulness:

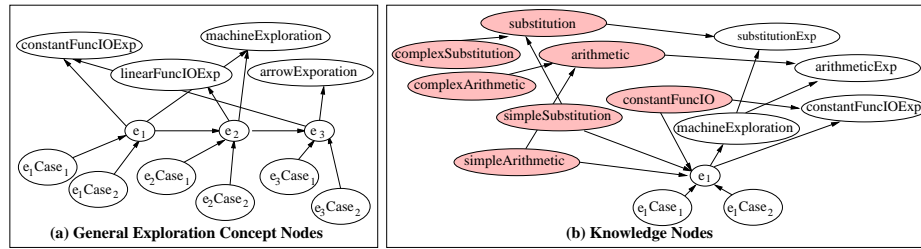
1. The more exercises the students effectively explored according to the Student Model, the more they improved on the post-test.
2. The more hints the students asked ACE for, the more they improved on the post-tests. Since the concepts targeted by the hints are based on the Student Model’s assessment, this is an indication that the model can enable the Coach to provide relevant feedback that helps guide the students’ exploration.

Analyzing students’ behaviour during the study also uncovered two main problems with the model’s assessment. First, a category of exploration assessment was missing from the model. Second, the model under-estimated the exploratory behaviour of knowledgeable students. In the next section, we describe these inadequacies and the new version of the model that we built to address them.

### 3.2 Version 2

Figure 2 (Version 2) shows the high-level structure of the second version of the Student Model. The differences between this model and the previous version are the addition of “General Exploration Concepts” nodes and the dependencies between the knowledge nodes and exploration nodes. We now describe both of these changes.

**General Exploration Concepts** The study uncovered that some students thoroughly explored many exercises but consistently exhibited poor exploration of exercises that targeted the same general concepts (e.g., exercises with constant functions). The Coach could determine that the student had exhibited poor exploratory behaviour in a couple of exercises. However, since the first version of the model maintained an assessment of cross-exercise exploration only through categories of relevant exploration cases (see fig. 2 (Version 1)), the Coach had



**Fig. 4.** Example Portions of the Bayesian Network in the Second Version of the Model Showing (a) the General Exploration Concepts and (b) the Knowledge Nodes

no way of knowing that both exercises targeted the same general concept. We addressed this Student Model inadequacy by adding a new type of node to the network, which represents the exploration of general concepts (“Exploration of General Concepts” in fig. 2 (Version 2)). These nodes allow the model to maintain an assessment of how effectively students are exploring concepts that appear in multiple exercises or units, but that do not correspond to any of the relevant exploration cases within the exercises. Figure 4(a) illustrates how these general exploration concepts appear in the network, which in this example involve the exploration of input/output in constant and linear functions (“constantFuncIOExp” and “linearFuncIOExp” in fig. 4(a)).

**Knowledge Nodes** The study also revealed that several highly-knowledgeable students (identified via a pre-test) rightfully ignored unnecessary warnings from the Coach to explore an exercise further, and chose not to explore concepts they already understood. The model’s inadequacy in these situations was that its exploration assessment was based solely on how active students were during the interaction, captured by the relevant exploration case nodes. The model did not take students’ knowledge into account in deciding if and how much they actually needed to explore to improve their understanding.

We modified the second version of the Student Model to incorporate information on the student’s knowledge into the model’s assessment of effective exploration. Thus, in the new model, low exploration activity on certain concepts is only a sign of poor exploration if the student does not already know those concepts. Student knowledge is incorporated into the assessment by having each exploration node influenced by a set of related knowledge nodes.

The semantics of both knowledge nodes and exploration nodes also changed in the new model. In the previous version, knowledge nodes represented general knowledge of material assessed through both exploratory behaviour and explicitly correct behaviour, when demonstrated (e.g., in the Arrow Unit). In the new model, knowledge nodes represent only knowledge related to non-exploratory activities and receive evidence only from the students’ explicitly correct behaviour (including pre-test results, when available). Similarly, while in the previous version probabilities for exploration nodes were simply a quantitative measure of

**Table 1.** Results from the Evaluation of the Second Version of the Student Model

	<b>Version 1</b>	<b>Version 2</b>
<b># Subjects</b>	14	5
<b># Unnecessary warnings</b>	62	2
<b>Total # warnings</b>	163	42
<b>% of warnings that were unnecessary</b>	38%	5%
<b># Premature passes</b>	6	5
<b>Total # exercises to be assessed</b>	154	55
<b>% of premature passes</b>	4%	9%

the student’s coverage of the relevant exploration cases, they now represent the likelihood that the student has explored the related concepts sufficiently to understand them. This assessment depends on both the student’s exploratory actions and knowledge of the related topic.

Figure 4(b) illustrates an example of the relationship between knowledge nodes and exploration nodes. Exercise 1 involves knowledge of simple substitution, simple arithmetic and the input/output of constant functions. Therefore, the corresponding exercise node “e<sub>1</sub>” depends on knowledge nodes for those concepts (the shaded nodes “simpleSubstitution”, “simpleArithmetic” and “constantFuncIO” in fig. 4(b)), as well as the relevant exploration cases (e.g., “e<sub>1</sub>Case<sub>1</sub>” in fig. 4(b)). The CPT for each exploration node is designed so that if the student has high knowledge of the related concepts, the probability that the student needs to explore more thoroughly is low. When there is a low probability that the student knows the corresponding concepts, the probability that the student needs to explore more thoroughly is based on the adequacy of the student’s coverage of the relevant exploration cases.

## 4 Evaluation

After revising ACE’s Student Model in light of the observations made during the first evaluation, we ran a second study to determine the effectiveness of the changes. In particular, before going to a fuller scale evaluation we wanted to verify that the changes in the model actually improve the way ACE issues warnings. As discussed in section 3.1, the Coach issues a warning to explore further when a student tries to leave an exercise before the Student Model assesses that the exercise was explored effectively. Ideally, the assessment of this Student Model compared with that of the previous version should result in the Coach intervening less frequently with the high ability students, while still supporting the students who are experiencing difficulty with the exploration process. In addition, the model should be able to detect situations in which students are systematically failing to explore general concepts, such as the input/output of specific function types.

A total of five subjects participated in the study. As in the previous study, the subjects were first-year university students who had not taken a university



math course. Subjects participated in one session that lasted at most 80 minutes and consisted of a pre-test, a session with ACE and a post-test. The pre-test and post-test consisted of 39 questions, divided equally into questions on i) function output recognition and generation, ii) graph and equation property recognition and iii) graph-equation correspondence. One researcher observed each session and ACE was instrumented to produce log files of the students' interface actions. The pre-test scores were used to set the values of the knowledge nodes in the model before the students started using ACE.

To verify that the new model reduced unnecessary interruptions for more knowledgeable students, without becoming too lenient with less knowledgeable ones, the logs from both the old and new study were analyzed by hand for two event counts: the number of unnecessary warnings that ACE generated and the number of premature transitions to new exercises that ACE allowed. Unnecessary warnings were considered to occur when the Coach generated a warning to explore an exercise further, despite it being clear from the pre-test that the student already understood the concepts associated with that exercise. A premature pass occurred when the Student Model determined that the student had effectively explored the exercise, but the student did not appear to understand the associated concepts on the post-test.

Table 1 shows the results of the analysis. As desired, the new Student Model resulted in a substantial reduction in the percentage of warnings that were deemed to be unnecessary (from 38% with the old model to 5% with the new model). The number of premature passes did rise slightly (from 4% to 9%). This indicates that the model, at times, overestimates the students' exploratory behaviour. Instances of the Student Model over-estimating exploratory behaviour were also observed in the first evaluation. Although knowledge is now taken into account together with the students' actions to assess effective exploration, the model still has no way to assess whether or not the students actually use this knowledge to reason about the outcome of their exploratory actions. We plan to address this problem in future versions of the model.

Unfortunately, we could not evaluate accuracy of the second change (the addition of general exploration concept nodes) since, unlike in the first study, none of the subjects poorly explored any of the more general concepts.

## 5 Conclusions and Future Work

There is mounting evidence that students need support in open learning environments (e.g., [7]). This support cannot be provided in a tailored and timely manner without knowing when and why a student is having difficulty exploring. This paper has presented the details of a Student Model that assesses the effectiveness of a student's exploration in ACE, an open learning environment for mathematical functions.

We built two versions of the Student Model. Results from the evaluation of the initial version provided confirmation of the model's accuracy and usefulness, but also uncovered some problems. The second version considerably improved

the model's performance since knowledgeable students received fewer unnecessary interventions, and yet ACE continued to support students who were experiencing difficulty with the exploration process. These results support our belief that having a Student Model assess the effectiveness of a student's exploratory behaviour can help provide the tailored support that many students need to learn from open learning environments.

Future work on the ACE Student Model will be to continue improving the model's assessment of effective exploration. Our studies showed that while it is important for students to perform the right set of exploratory actions, these actions alone are not always sufficient to learn the targeted material. The students need to proactively reason about their actions. Thus, we plan to include in our model additional factors that would permit a more accurate assessment of the student's exploration. These factors include the student's tendency to self-explain, whether or not the student is attending to results of her exploration, and the time spent on each relevant exploration case.

## References

1. A. Bunt. On creating a student model to assess effective exploratory behaviour in an open learning environment. Master's thesis, University of British Columbia, 2001.
2. A. Bunt, C. Conati, M. Huggett, and K. Muldner. On improving the effectiveness of open learning environments through tailored support for exploration. In J.D. Moore, C.L. Redfield, and W.L. Johnson, editors, *AIED 2001*, pages 365–376, San Antonio, TX, 2001.
3. A. D'Souza, J. Rickel, B. Herreros, and W.L. Johnson. An automated lab instructor for simulated science experiments. In J.D. Moore, C.L. Redfield, and W.L. Johnson, editors, *AIED 2001*, pages 65–75, San Antonio, TX, 2001.
4. M. Njoo and T. de Jong. Exploratory learning with a computer simulation for control theory: Learning processes and instructional support. *Journal of Research in Science Teaching*, 30(8):821–844, 1993.
5. J. Pearl. *Probabilistic Reasoning in Intelligent Systems*. Morgan Kaufmann, Los Altos, CA, 1988.
6. B.J. Reiser, W.A. Copen, M. Ranney, A. Hamid, and D.Y. Kimberg. Cognitive and motivational consequences of tutoring and discovery learning. Technical report, The Institute for the Learning Sciences, 1994.
7. V.J. Shute and R. Glaser. A large-scale evaluation of an intelligent discovery world: Smithtown. *Interactive Learning Environments*, 1:55–77, 1990.
8. W.R. van Joolingen and T. de Jong. Supporting hypothesis generation by learners exploring an interactive computer simulation. *Instructional Science*, 20, 1991.
9. K. VanLehn. Student modeling. In M. C. Polson and J. J. Richardson, editors, *Foundations of Intelligent Tutoring Systems*, pages 55–78. Erlbaum, Hillsdale, NJ, 1988.
10. K. Veermans, T. de Jong, and W.R. van Joolingen. Promoting self-directed learning in simulation-based discovery learning environments through intelligent support. *Interactive Learning Environments*, 8(3):229–255, 2000.