

Building and Evaluating an Intelligent Pedagogical Agent to Improve the Effectiveness of an Educational Game

Cristina Conati
University of British Columbia
2366 Main Mall
Vancouver, B.C., V6T1Z4, Canada
conati@cs.ubc.ca

Xiaohong Zhao
Simon Fraser University.
8888 University Drive
Burnaby, B.C., V5A1S6
xzhao2@cs.sfu.ca

ABSTRACT

Electronic educational games can be highly entertaining, but studies have shown that they do not always trigger learning. To enhance the effectiveness of educational games, we propose intelligent pedagogical agents that can provide individualized instruction integrated with the entertaining nature of the games. In this paper, we describe one such agent, that we have developed for Prime Climb, an educational game on number factorization. The Prime Climb agent relies on a probabilistic student model to generate tailored interventions aimed at helping students learn number factorization through the game. After describing the functioning of the agent and the underlying student model, we report the results of an empirical study that we performed to test the agent's effectiveness.

Categories and Subject Descriptors

I.2.1 [Artificial Intelligence]: Applications and Expert Systems – Games;

I.2.3 [Artificial Intelligence]: Deduction and Theorem Proving – uncertainty and probabilistic reasoning;

K.3.1 [Computers and Education]: Computer Uses in Education – computer-managed instruction (CMI).

General Terms

Human Factors, Experimentation.

Keywords

Intelligent Agents, Educational Games, User Modeling, Dynamic Bayesian Networks.

1. INTRODUCTION

Because electronic games are highly engaging, researchers have started to investigate whether they could be used to assist learning, especially for those children who have lost interest in math or other science courses [7, 11]. However, there is little empirical evidence that electronic educational games can

promote learning [11], unless the interaction is led by teachers and integrated with other instructional activities [7]. One of the main reasons for this limitation of educational games is that learning how to play the game does not necessarily imply learning the target instructional domain. Learning happens only when students actively build the connections between game moves and underlying knowledge [2]. Whether students can build these connections usually depends upon individual differences in knowledge and in the meta-cognitive skills relevant to learn from autonomous exploration (e.g., self-explanation and self-monitoring) [13].

In light of these findings, we are working on making educational games more pedagogically effective by making them capable of providing interactions tailored to each student's needs and targeted at stimulating learning when necessary. Although there is well established research on building student-adaptive computer-based educational tools, to date very little of this research has focused on electronic educational games. There are two main challenges for our approach. The first is that in educational games it is especially difficult to assess students' knowledge and learning from the interaction with the game, because often game actions do not have a direct connection with a student's understanding of the underlying domain. The second challenge is how to provide individualized interventions that trigger learning, without interfering with the high level of engagement that educational games usually trigger precisely because they do not remind students of traditional educational activities.

In this paper, we describe our work on making Prime Climb, an educational game for number factorization, more effective through an intelligent pedagogical agent that can provide students with adaptive, individualized support.

Previous studies have shown that animated pedagogical agents tend to increase the learner's motivation and engagement [6, 10]. Furthermore, these agents have been shown to be pedagogically effective in several computer-based learning environments (e.g., [8, 9]). Thus, using animated pedagogical agents seems quite a natural choice for delivering instructional interventions when it is crucial to maintain a high level of learner engagement, as is the case for educational games.

Conati and Klawe [4] discuss how intelligent agents for educational games can improve learning and maintain engagement. Their approach takes into account student meta-cognitive skills and emotional reactions in addition to student cognitive skills, as they are assessed by a probabilistic student model. Together with [15], the research presented in this paper is

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

IUI-CADUI '04, January 13-16, 2004, Island of Madeira, Portugal
Copyright 2004 ACM 1-58113-815-6/04/0001...\$5.00.

a step toward the development of these emotionally intelligent agents. We have developed a simple pedagogical agent that provides individualized help during the interaction with the Prime Climb game, by relying on a probabilistic model of the student's factorization knowledge as it evolves during the interaction. Our goal is twofold; (i) to see how far this agent can go in improving student learning while maintaining engagement, by using only the assessment of student knowledge; (ii) to integrate the model of student learning with a model of student emotions [15], to allow for more informed agent interventions.

In the rest of the paper, we focus on the first of these two goals. We start with a description of Prime Climb. Then, we describe the design of the agent, and the probabilistic model that drives its behavior. Finally, we discuss the results of a user study to test the agent's effectiveness.

2. THE GAME: PRIME CLIMB

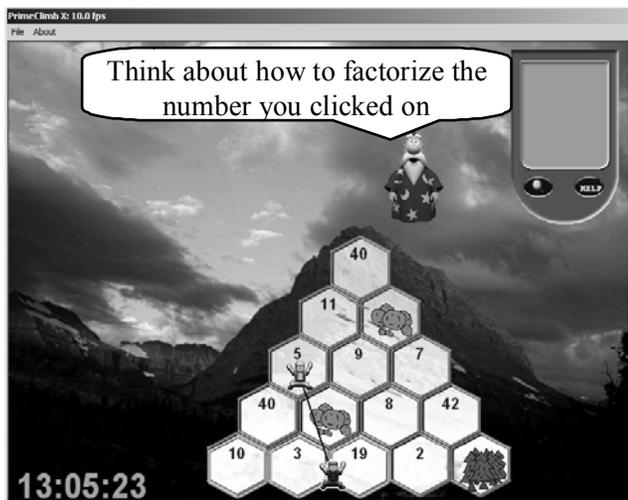


Figure 1: The Prime Climb Interface

Prime Climb is an educational game devised by the EGEMS (Electronic Games for Education In Math and Science) group at the University of British Columbia. The main goal of the game is to help grade 6 and 7 students learn number factorization in a highly motivating game environment. Prime Climb involves two players who must collaborate to climb a series of mountains. Each mountain is divided into hexes labeled with numbers (see Figure 1). The main rule of the game is that each player can only move to a number that does not share any common factor with the partner's number. If a wrong number is chosen, the climber falls and swings from the rope until she can grab onto a correct number. For instance, in Figure 1 the player at the bottom of the mountain fell because she tried to move to hex 10 while the other player was on hex 5.

To help students with the climbing task, Prime Climb includes a tool known as the *Magnifying glass*. This tool allows a student to view the factor tree for each number on a mountain in the PDA shown at the top right corner of the game (see Figure 1). The factor tree is a common representation used in math text books to visualize number factorization as the recursive decomposition of a



Figure 2: Sample Factor Tree

number into its non-prime factors (see Figure 2). Students can view the factor tree of a number incrementally, by clicking each of the visible nodes to see the first level of their decomposition. Although there can be different factor trees for a number, the Prime Climb PDA shows the most balanced, for easier display.

Several user studies performed by the EGEMS group have shown that Prime Climb, like most educational games that have been actually tested, is highly entertaining but fails to show reliable learning effects. Thus, we

have modified the game to include an animated pedagogical agent designed to trigger learning as it provides support during the climbing task, as described in the next section.

3. THE PRIME CLIMB PEDAGOGICAL AGENT

The Prime Climb agent is implemented with the Microsoft Agent Package. There is a pedagogical agent for each of the two players involved in the game. The agent (see Figure 1) gives hints either on demand, when the student asks for them, or unsolicited, when it decides that the student needs help in learning from the game.

A student can ask its agent for hints by clicking on the PDA "help" button (see Figure 2). This activates a dialogue box with a selection of questions reflecting the most common problems that students have when playing Prime Climb. The agent answers questions at an incremental level of detail as the student clicks on the "further help" button available in the dialogue box.

In addition to answering the students' requests for help, the agent provides unsolicited hints to overcome the students' tendency to avoid seeking help even when they need it [1, 4]. To decide when to intervene and what hints to provide, the agent relies on a probabilistic model of the student's factorization knowledge. The model is a Dynamic Bayesian network [5] that tracks student actions and agent interventions during game playing and uses this information as evidence to assess student knowledge as the interaction proceeds. We describe this model in more detail in the next section.

In general, the pedagogical agent provides unsolicited hints when the probabilities in the student model indicate that the student is missing key pieces of knowledge to learn from her current move. A sample of the agent's unsolicited hints is shown in Table 1. In the following, we illustrate when these hints are provided.

- Hint 1_1a is given when the student performs a wrong move and the agent thinks that the move is due to the student not knowing the factorization of the number she tried to move to. This happens when the student model reports a low probability that the student knows the factorization of the number in question. Similarly, Hint 1_1b is given when the agent thinks that a wrong move is due to the student not knowing the factorization of her partner's number. When both numbers involved in a wrong move have low probability, the agent addresses first the number the student moved to. Hints

1_2 and 1_3 are provided when the student continues to move incorrectly after receiving Hint 1_1a or Hint 1_1b.

- Hints 2_1 to 2_3 are given when the student performs a wrong move and the agent thinks that the student’s error is due to lack of understanding of either the game’s general climbing rule (Hint 2_1) or the concept of common factors. Because the agent’s student model currently does not explicitly represent the probability that the student knows these pieces of knowledge, hints of this type are given only when a student makes an error but the student model indicates that the student knows how to factorize the numbers involved in the wrong move.
- Hint 3_1 aims to counteract the problem of a student performing correct moves despite not understanding the underlying factorization (due to guessing, to remembering previous patterns, or to the agent more specific hints). The agent detects this problem when the student model gives a low probability that the student knows the factorization of the numbers involved in a correct move.

Table 1: Sample Agent’s Unsolicited Hints

<i>Hint1_1a</i>	“Think about how to factorize the number you clicked on”
<i>Hint1_1b</i>	“Think about how to factorize your partner’s number”
<i>Hint1_2</i>	“Do you need help? Use the Magnifying glass”
<i>Hint1_3</i>	“To factorize X, do this.....”
<i>Hint2_1</i>	“You can not click on a number which shares common factors with your partner’s number”
<i>Hint2_2</i>	“Are there common factors between your number and your partner’s? Use the Magnifying Glass to find out!”
<i>Hint2_3</i>	“Do you know that x and y share z as a common factor?”
<i>Hint3_1</i>	“Great, you are right this time. Do you know why?”

To avoid interfering too much with the pace of the interaction, we don’t require the student to explicitly acknowledge the agent’s hints. However, to make sure that students do notice the hints, the agent audibly verbalizes them, in addition to showing the text in a bubble (see Figure 1). We now proceed to illustrate the probabilistic student model that drives the agent’s interventions.

4. THE AGENT’S STUDENT MODEL

Modeling students’ knowledge in educational games involves a high level of uncertainty. The student model can only access student actions such as moves and tool usage, not the intermediate mental states that cause these actions. Furthermore, several studies have shown that students can learn to play an educational game without learning the underlying domain. Thus, analyzing student actions in Prime Climb often does not give a clear insight onto a student’s knowledge. A solution to this problem could be to insert into the game more explicit tests of factorization knowledge. However, this would endanger the high level of motivation that an educational game usually triggers exactly because it does not remind students of traditional pedagogical

activities. Thus, both Prime Climb and our agent are designed to interrupt game playing as little as possible, making the interpretation of student actions highly ambiguous. As we mentioned in the previous section, we use Dynamic Bayesian Networks (DBNs) to handle the uncertainty in the student model assessment.

DBNs are a framework for reasoning under uncertainty designed to model situations that evolve over time. They are, therefore, well suited to model the unfolding of a student’s interaction with the game, and the corresponding evolution of the student’s factorization knowledge.

A DBN keeps track of variables whose values change overtime by representing multiple copies of these variables, one for each snapshot of the temporal process (or time slice). Links between variables across time slices represent their temporal dependencies. However, it quickly becomes impractical to keep in a DBN all the relevant time slices. The rollup mechanism keeps at most two time slices¹ [47]: the network at slice t-1 is removed after the evidence in this slice has been processed and the network for slice t is established. The prior probabilities of root nodes in t are set to the posteriors of the same nodes in slice t-1.

The DBNs for the Prime Climb student model actually work over two levels of temporal evolution, to deal with the computational complexity of modeling the fast-paced interaction that the game generates. One level, which represents the *short-term student model* (described in more detail in the next section), uses a DBN to capture the evolution of student knowledge from one interface action to the next, while climbing a specific mountain. The second level models the evolution of student knowledge when moving between mountains, either within a game session or across sessions. At this level, a time slice encompasses the climbing of a specific mountain and thus corresponds to the short-term model for that climbing task. Following the approach suggested in [3], at this level a form of *roll-up* is performed every time a student finishes climbing a mountain. Before the short-term model for the current mountain is discarded, the probabilities of the relevant student knowledge are saved in the *long-term student model*. The probabilities in the long-term model are then used as priors in the new short-term model when the student accesses a new mountain. Thus, at any given time the long-term student model encodes the assessment of a student’s factorization knowledge given all the mountains completed so far. In the rest of this section, we provide more details on the short-term model. More information on the long-term model can be found in [14].

4.1 The Short-Term Student Model: Nodes and Assumptions

As we mentioned earlier, each time slice in the Prime Climb short-term model represents the student’s knowledge state after a relevant interface event.

The short-term model uses several random variables to represent the relevant student knowledge and interface events. The core of these variables includes:

- **Factorization Nodes F_X** : for each number X on a mountain, the corresponding short-term model includes a binary node F_X ,

¹ Or as many slices as are necessary to model the direct dependencies of the current state from past states.

representing whether the student has mastered the factorization of X down to its prime factors. The model also includes a factorization node for every number that is part of the factorization of X .

- **Node KFT (Knowledge of Factor Tree):** this binary node models whether or not a student knows the factor tree representation, and thus can learn the factorization of a number by seeing the factor tree of that number.
- **Nodes Click_X:** each binary node $Click_X$ models whether or not a student's action of clicking number X is correct, i.e. whether or not X shares any common factor with the partner's number.
- **Nodes Mag_X:** each binary node Mag_X denotes a student action of using the magnifying glass to see the factorization of number X .

$Click_X$ and Mag_X nodes are evidence nodes that are introduced in the model when the corresponding actions occur, and are immediately set to either one of their values. As the list above indicates, the student model currently does not include nodes that explicitly represent knowledge of the game rules or knowledge related to the concept of common factors. As we discussed in the previous section, the agent infers lack of this knowledge when it can rule out lack of factorization knowledge on specific numbers as a cause of incorrect actions.

Before going into detail as to how the nodes described above are structured into the short-term student models, we list the set of assumptions that we use to define the structure.

Assumption 1: Knowing the prime factorization of a number (i.e., the factorization of a number down to its prime factors), influences the probability of knowing the factorization of its non-prime factors. In particular, if a student knows the prime factorization of Z , where $Z=X_1*X_2*Y_1*Y_2$, then she probably knows the factorization of X and Y , where $X = X_1*X_2$ and $Y=Y_1*Y_2$. This assumption was made after consulting the math teachers involved in developing Prime Climb. The teachers also pointed out that the opposite is usually not true: it is hard to predict if a student knows a number's factorization given that the student knows how to factorize its factors.

Assumption 2: Clicking on a number that does not share common factors with the partner's number increases the probability that the student knows the factorization of the two numbers, although this action could also be due to a lucky guess or remembering previous moving patterns. A wrong click decreases the probability that the student knows the factorization of the two numbers, although it could also be due to a slip.

Assumption 3: When a student uses the magnifying glass on number X , the probability that the student knows how to factorize X increases if the student knows the factor tree representation. Otherwise the probability stays the same.

Assumption 4: When a student uses the magnifying glass on number X at time $t-1$, and then correctly (incorrectly) moves to X at time t , the move provides evidence that the student learned (did not learn) the correct factorization of X by using the magnifying glass at time $t-1$. Thus, this action provides evidence that increases the probability of the student knowing (not knowing) the factor tree representation.

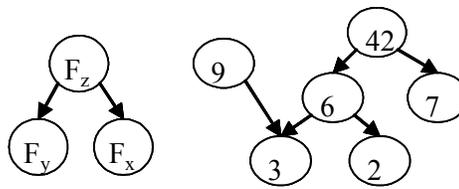


Figure 3: Dependencies between factorization nodes. On the left, $Z=X*Y$.

4.2 Structure of the Short-Term Model

The process of building the short-term model for a given mountain starts by generating the part of the model that includes all the nodes representing the knowledge relevant for climbing that mountain. These consist of the factorization nodes for the numbers on the mountain, as well as the *KFT* node. The model represents the factorization of each number on the mountain following the factor tree representation (see Figure 2), and encodes only the factorization shown in the PDA. The dependencies among each number and its factors are defined by assumption 1 above, and are represented in the model as shown in Figure 3. The right side of the figure shows some of the dependencies for the numbers of the mountain in Figure 1. Figure 7, in a later section, shows the complete model for that mountain. The conditional probability table (CPT) for each non-root factorization node is defined so that the probability of the node being known is high when all the parent factorization nodes are true, and decreases proportionally with the number of unknown parents. At the beginning of the interaction, the *KFT* node is not connected to any other node in the model.

The priors of root factorization nodes and of the *KFT* node in the short-term model are initialized using the long-term student model. As we mentioned earlier, the long-term model encodes the current assessment of the student's knowledge, based either on evidence accumulated through the student's previous climbing tasks or on alternative forms of assessment if the student is a first time player (e.g. a pre-test on the relevant factorization knowledge or teachers' estimates).

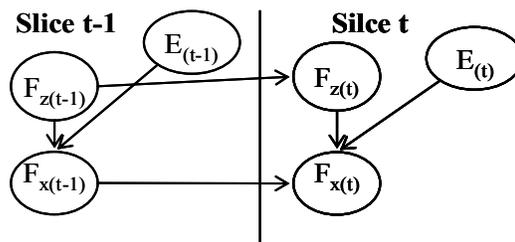


Figure 4: General relation between knowledge and evidence nodes in the short-term model

The rest of the short-term model is built dynamically as the student interacts with the game, following assumptions 2 to 4. Figure 4 shows the basic structure that, in our DBN, represents the relations between factorization nodes and evidence E coming from interface actions. As the figure shows, these relations are represented in the diagnostic rather than causal direction. The reason for this choice is that a causal representation would generate unwarranted dependencies between factorization nodes.

For instance, if node $E_{x(t-1)}$ in Figure 4 was inserted as a child of node $F_{x(t-1)}$, the evidence would propagate upward to both $F_{x(t-1)}$ and $F_{z(t-1)}$, contradicting the teachers’ suggestion that factorization knowledge of a number’s factors does not provide a direct indication that a student can factorize that number. This, along with the fact that teachers did not find the causal representation more intuitive than the diagnostic one, made the latter more appropriate for this particular model.

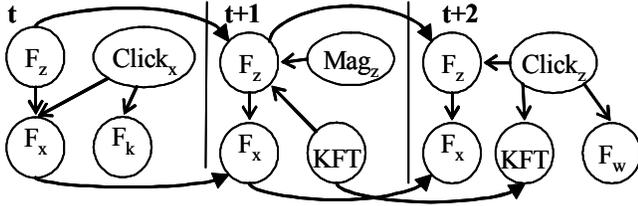


Figure 5: Representation of a student’s actions in the short-term model

Figure 5 shows a schematic representation of how some of the most common game actions and related assumptions are represented in the model. The action of clicking on number X when the partner is on number K is represented by (1) adding a node $Click_x$ to the model with links to both nodes F_x and F_k (see Figure 5, slice t); (2) augmenting (decreasing) the conditional probability of F_x and F_k if the action is correct (incorrect)², as suggested by assumption 2.

The action of activating the magnifying glass on number Z is modeled by adding a node Mag_z , as well as links from both Mag_z and KFT to F_z (see Figure 5, slice $t+1$). The corresponding CPT represents the assumption that the probability of gaining knowledge from seeing the factor tree of a number depends on the student’s knowledge of the factor tree representation (assumption 3).

Finally, when a click action on a number follows the activation of the magnifying glass on that number (say number Z in Figure 5), an additional link is added from the new Click node ($Click_z$ in Figure 5, slice $t+2$) to the KFT node, to represent the influence that a move following usage of the magnifying glass has on the assessment of the KFT node (assumption 4).

The short-term model includes similar mechanisms to model the incremental viewing of the factor tree and the influence of the agent’s hints on student knowledge.

Although Figure 5 shows a sequence of three time slices, we maintain at most two time slices at any given time, to reduce the computational complexity of evaluating the model. However, because of our networks’ structure we cannot apply the standard

² In the case of an incorrect click, the two factorization nodes should be conditionally dependent given the action, so that the node with the lower probability of being known can be “blamed” more for the incorrect move. Because of the non-causal structure of the model, the only way to encode this dependency is to add a direct arc between the two factorization nodes, thus increasing the model complexity. We decided to first try the simplified structure in Figure 5, slice t , and to change it if empirical testing shows that it significantly affects the agent’s performance.

roll-up technique to adopt this approach. In the next section, we illustrate why and we describe the alternative form of roll up that we use.

4.3 Modified Roll-Up for the Short-Term Model

The standard roll-up technique in DBNs consists of evaluating the effect of new evidence (e.g. $E_{(t-1)}$ in Figure 4) at time $t-1$, creating a new time slice at time t , assigning the posteriors of root nodes at time $t-1$ as the priors of the same nodes at time t , and finally removing slice $t-1$.

This technique is only accurate when the only nodes that evolve over time (*dynamic nodes* from now on) are root nodes, as is the case in most applications of DBNs. When this is true, “rolling up” the posteriors of dynamic nodes in one time slice as priors of the same nodes in the next slice carries over all the information accumulated on dynamic nodes through past evidence. On the other hand, when some of the dynamic nodes are non-root (as is the case in our model, see for instance, $F_{x(t-1)}$ and $F_{x(t)}$ in Figure 4), basic roll-up loses most of the information accumulated through direct evidence for these nodes, once time slices are removed.

A possible solution, discussed in [12], is to maintain several time slices and “roll-up” a slice (and the corresponding evidence) only when its influence on the current slice becomes negligible. However, this approach can be computationally expensive, especially when the non-root nodes can have several parents and thus quite large CPTs. In our model, for instance, it causes delays in the order of seconds, which highly disrupt the pace of the game.

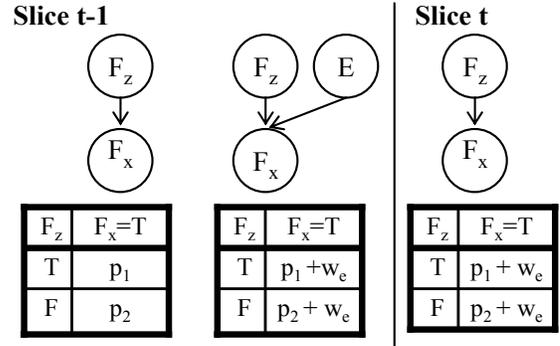


Figure 6: Alternative roll-up in the short-term model

We use an alternative approach to keep only two time slices in the short-term student model. In this approach, changes in the probability of a non-root factorization node due to a new interface event are transferred to the next time slice by changing the CPT of that node given its parent factorization nodes. Figure 6 illustrates this process when the new action (node E) brings positive evidence on student knowledge. In Figure 6, w_e in the CPTs for node F_x is the weight that action E has on the assessment of that knowledge node. We further reduce the complexity of the model by assuming that a learned node stays learned (i.e., there is no forgetting), so that we can avoid the specification of all the CPT entries corresponding to states in which the non-root node was known in the previous time slice.

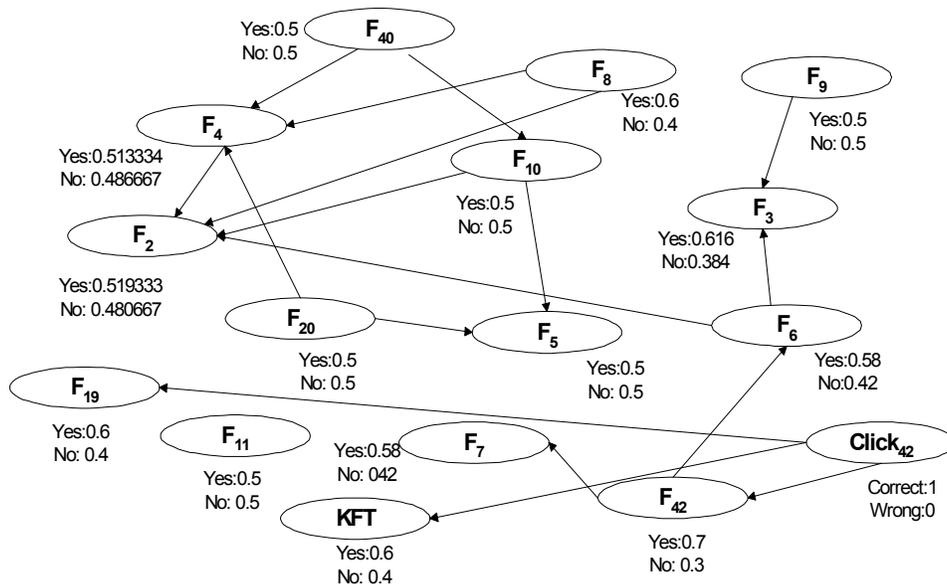


Figure 7: State of the short-term model after a sequence of student actions

As Figure 6 shows, what we have done in effect is to represent the influence of new evidence as a change in the belief about how the student factorization knowledge of a number predicts the student factorization knowledge of its non-prime factors. A possibly awkward implication of this approach is that, in the short-term model, the dependencies among knowledge nodes will be different for different numbers and time slices. This, however, is not a real inconsistency, because the relation between factor nodes summarized by Assumption 1 does not represent a causal mechanism with well defined, general properties, but rather a part-of relation which does in fact depend on each number and each student.

Figure 7 shows a concrete example of a short-term model slice. This is the final time slice after the student performs the following sequence of actions, starting from the game state in Figure 1: moving to 8 when the partner is on 3 (correct click action), activating the magnifying glass on 42 after the partner moved to 19, then moving to 42.

5. THE STUDY

To evaluate the effectiveness of the Prime Climb pedagogical agent and of the underlying student model, we ran a study with 20 grade 7 students.

The study participants were divided into an experimental and a control group. In the experimental group, ten students played with the version of Prime Climb including the pedagogical agent. In the control group, ten students played with the original version of the game.

The study took place in the school and consisted of ten sessions, each running two subjects in parallel. Each subject played with an experimenter, rather than with a peer. We chose this setting to avoid the possible confounding effect due to the students' different playing patterns and prior knowledge.

Before the study, students were given a pre-test involving seven multiple-choice questions on finding common factors between two numbers. Five of the questions were worth a maximum of 2 points, and two a maximum of 4 points, depending upon the number of common factors involved. Thus, the maximum total score for the pre-test was 18. Unfortunately, we did not have time to use the pre-test results to initialize each student's long term model for this study, so we set the probability of all the nodes to 0.5 for every student. After receiving a brief introduction to the game, students played Prime Climb for 20 minutes. Then, they took a post-test. Students in the experimental group also filled out a short questionnaire to gauge their opinion of the pedagogical agent. Log files were collected, recording the relevant student interface actions, as well as all the agent's hints.

5.1 Results

We started our analysis by comparing learning gains in the two groups, where "gain" is defined as the difference between a student's post-test and pre-test scores.

As Table 2 shows, the experimental group gained more than the control, and the difference is marginally significant ($p = 0.068$)³, with an effect size of 0.7. This is quite a large effect, especially considering that students interacted with the system only for 20 minutes.

Table 2 also shows that students in the control group hardly improved from pre-test to post-test, confirming that students can often learn very little from a game environment without any external guidance. In addition, the fact that during the game students in the control group climbed slightly more mountains (on

³ This is a one-tailed t-test, due to our initial assumption that students from the experimental group should gain more because of the agent's interventions

average) than students in the experimental group (7.1 vs. 6.6) confirms that students can become quite proficient in playing an educational game without necessarily learning the underlying knowledge. The higher number of mountains climbed by the control group also rules out the possibility that students in the experimental group learned more because they had more chances to practice the relevant factorization skills.

Table 2: Statistics on learning gains (post-test - pre-test)

<i>Group</i>	<i>Mean</i>	<i>Std. Dev.</i>	<i>t</i>	<i>p(1-tailed)</i>
<i>Experimental (10)⁴</i>	2.40	3.406	1.56	0.068
<i>Control (10)⁴</i>	0.30	2.541		

A t-test on the pre-test scores of the two groups ruled out pre-existing knowledge as a confounding variable, since the difference in the score of the two groups (11.6 for control and 10.8 for experimental) is not significant ($p > 0.7$).

Table 3: Correlation between agent's hints and learning

<i>Hint Type</i>	1_1a	1_1b	1_3	2_1	3_1
<i>Pearson r</i>	0.606	0.216	0.7	0.038	.460
<i>p (1-tailed)</i>	0.03	0.275	0.013	.46	0.095
<i>% of total hints</i>	21%	9.2%	22.4%	7.9%	11.8%
<i>% followed by correct action</i>	87%	83%	65%	100%	N.A.

To further verify that the better performance of the experimental group is actually due to the agent's interventions during the game, we checked the correlation between learning gains and the various kinds of hints that the agent provided. We only considered unsolicited hints because during the study no student asked the agent for help. Furthermore, we excluded hint 1_2 (one of the two hints to use the magnify glass) because students never followed it, as well as hints 2_2 and 2_3 (which are a follow-up to hint 2_1), because none was generated during the study.

As table 3 shows, hints 1_1a (which, after a fall, suggests thinking about the factorization of the number the student tried to move to) and 1_3 (which gives the factorization of a number after a more generic hint of type 1_1a or 1_1b) have a high and significant correlation with learning gains. This indicates that they are a possible cause of student learning in the experimental group. Although the correlation could be due to additional factors, such as student general ability and conscientiousness, the fact that 87% of hints 1_1a and 65% of hints 1_3 were followed by a correct move (see Table 3) supports the hypothesis that they played a role in triggering student **learning**. These percentages also indicate that the student model is quite effective in detecting holes in the student's knowledge.

⁴ These numbers are reported incorrectly in the version printed in the conference proceedings

We cannot draw any principled conclusion from the low correlation of hints 1_1b (which suggests thinking about the factorization of the partner number after a fall) because the correlation is not significant, probably due to the fact that the agent gave few of these hints. However, the fact that 83% of these hints were followed by a correct move provide further evidence that the student model is quite good at detecting missing student knowledge.

Hints 2.1 (which remind students that they cannot move to numbers that share factors with their partner's) basically have no correlation with learning. However, this is consistent with their purpose. These hints are given when the agent thinks that the student has not understood the game's general rule, although she has all the necessary math knowledge to follow it. Thus, they have no ambition to increase learning.

It is difficult to make any conclusive statement on hints 3_1, because they have a medium correlation at the 0.1 level of significance and we have no other way to tell whether they had any effect on student behavior (students were not required to perform any action in response to these hints). These hints are designed to overcome the common problem that students often learn to play an educational game effectively without learning the underlying knowledge. The fact that they were given quite infrequently (see table 3) may be an indication that our probabilistic model overestimates student knowledge, possibly giving too much credit to correct actions, too little blame to incorrect ones or using a too low threshold probability for judging a factorization node to be unknown. We are planning to experiment with different weights and thresholds, to see if there are parameter combinations that can improve the agent's pedagogical effectiveness.

6. DISCUSSION AND FUTURE WORK

The results discussed in the previous section offer encouraging evidence that providing individualized instruction by using an animated pedagogical agent can improve student learning with educational games. However, our goal is to have pedagogical agents that can both stimulate learning and maintain the high level of engagement that educational games usually generate. Currently, the agent tries to avoid interfering too much with student game playing by keeping a low threshold for unknown nodes and by allowing the student to ignore the agent hints, but it still gives priority to stimulate learning when the student model indicates that the student needs it. For instance, the agent always provides hints if the model says that the student is probably lacking the knowledge underlying a move, regardless of its correctness and without considering how these hints may impact engagement.

To gauge how the agent described in this paper influences student engagement in Prime Climb, we analyzed the students' answers to the post-study questionnaire. One of the items in the questionnaire required to score the statement "I like the agent Merlin" on a scale from 1 to 5. The average score for this item was 3.7. A second item asked whether a student would rather play Prime Climb again with or without the agent. Four students said "with", two students said "without", and four gave ambiguous answers ("sometime", "I am not sure", "both" and no answer). Thus, although these reactions are quite positive considering that the agent didn't really take any proactive action

to increase student engagement, we can argue that there is still quite a lot of room for improvement in making our pedagogical agent a fun addition to the Prime Climb game. Some of these improvements should definitely go in the direction of exploiting the non-verbal communication channels available to the agent (such as gesture, gaze and locomotion), which currently the Prime Climb agent does not utilize during the interaction. However, we also believe that the pedagogical agent could strike a better balance between learning and engagement if it could have access to the student's affective reactions to the game in addition to the student's knowledge. To this end, we are currently working on a probabilistic model of student affect also based on DBNs [15]. The goal is to integrate this model with the model of student learning, so that the agent's actions can be informed by both the student cognitive and affective state.

7. ACKNOWLEDGMENTS

We thank B.C. ASI (Advanced Systems Institute) for providing financial support to Xiaohong Zhao during this research. We also thank the EGEMS group for developing the Prime Climb game, Kasia Muldner and Andrea Bunt for their comments on an earlier version of this paper, and Xiaoming Zhou for his help with the user study.

8. REFERENCES

- [1] Aleven, V. and K.R. Koedinger. *Limitation of students' control: do students know when they need help?* Proc. of ITS 2000, Springer-Verlag. S. Antonio, TX, U.S.A.
- [2] Conati, C. and J. Fain Lehman. *Toward a Model of Student Education in Microworlds.* Proc. of the 15th Annual Conference of the Cognitive Science Society, 1993, Boulder, CO, U.S.A.
- [3] Conati, C., A. Gertner, and K. VanLehn, *Using Bayesian Networks to Manage Uncertainty in Student Modeling.* Journal of User Modeling and User-Adapted Interaction, 2002, 12(4).
- [4] Conati, C. and M. Klawe, *Socially Intelligent Agents in Educational Games*, in In *Socially Intelligent Agents - Creating Relationships with Computers and Robots.*, K. Dautenhahn, A. Bond, D. Canamero D, and B. Edmonds, Editors, 2002, Kluwer Academic Publishers.
- [5] Dean, T. and K. Kanazawa, *A Model for Reasoning about Persistence and Causation.* Computational Intelligence, 1989, 5(3).
- [6] Johnson, W.L. and J. Rickel. *Research in Animated Pedagogical Agents: Progress and Prospects for Training.* Proc. of IUI 2001, International Conference on Intelligent User Interfaces, 2001, Santa Fe, NM, U.S.A.
- [7] Klawe, M. *When Does The Use Of Computer Games And Other Interactive Multimedia Software Help Students Learn Mathematics?* NCTM Standards 2000 Technology Conference, 1998, Arlington, VA, U.S.A.
- [8] Lester, J., S. Converse, B. Stone, S. Kahler, and T. Barlow . *Animated Pedagogical Agents and Problem-Solving Effectiveness: A Large-Scale Empirical Evaluation.* Proc. of the 8th World Conference on Artificial Intelligence in Education, 1997, Kobe, Japan.
- [9] Lester, J.C., B.A. Stone, and G.D. Stelling, *Lifelike Pedagogical agents for mixed initiative problem solving in constructivist learning environments.* Journal of User Modeling and User-Adapted Interaction, 1999, 9(1-2).
- [10] Mitrovic, A. and P. Suraweera. *Evaluating an animated pedagogical agent.* Proc. of ITS 2000, Springer-Verlag. S. Antonio, TX, U.S.A.
- [11] Randel, J.M., B.A. Morris, C.D. Wetzel, and B.V. Whitehill, *The effectiveness of games for educational purposes: A review of recent research.* Simulation & Gaming, 1992, 23(3).
- [12] Schafer, R. and T. Weyrath. *Assessing Temporally Variable User Properties With Dynamic Bayesian Networks.* Proc. of the 6th Int. Conference on User Modeling, 1997, Chia Laguna, Italy.
- [13] Shute, V.J., *A comparison of learning environments: All that glitters....*, in *Computers as Cognitive Tools*, S. Lajoie, P. and S. Derry, Editors, 1993, Lawrence Erlbaum Associates: Hillsdale, NJ.
- [14] Zhao, X., *Adaptive Support for Student Learning in Educational Games*, M.Sc. thesis. Department of Computer Science, University of British Columbia, 2002, Vancouver, Canada.
- [15] Zhou, X. and C. Conati. *Inferring User Goals from Personality and Behavior in a Causal Model of User Affect.* Proc. of IUI 2003, International Conference on Intelligent User Interfaces, 2003, Maimi, FL, U.S.A.