ABSTRACT
The instruction of Computer Science, which many students have difficulty coping with, can be greatly aided by the use of educational software. To improve adaptation to its users (i.e., students), these systems are capable of maintaining a model of students’ knowledge and learning preferences. This paper presents POOLE III, an intelligent, interactive learning environment that creates and maintains individual and group student model which is in turn, used to customize its feedback and remediation to students. Results are promising. User evaluation indicates positive feedback from students on new functionalities supported by the use of the model.

Keywords
Student modeling, Adaptive Systems, Collaborative Learning Environment, Educational Software

1. INTRODUCTION

An intelligent tutoring system (ITS) is a computer program aimed at providing knowledgeable, individualized instruction in a one-on-one interaction with a learner (Holt, Dubs, Jones, & Greer, 1994). It acts as a tutor that diagnoses and remediates errors of the student. It knows the causes of student’s mistakes, such as misconceptions, and works to alleviate those causes. The ITS makes an approximation of student’s knowledge and mal-knowledge (Sison, 2001). An integrated learning environment (ILE) is a program that acts as a coach that gives guided instructions and keeps track of student’s records. The feedback it gives tells the student where he is wrong and where to proceed, but it is up to the student to determine if he wants to advance in its course (Sison, 2001). Various ILE, ITS and even integrations of both technologies have been made for the field of computer programming. C-Tutor is an ITS that teaches the C programming language (Hahn, Song, Tak, & Kim, 1995). An example of a combination of an ILE combined with an ITS is ELM-ART, a tutoring system for the LISP programming language (Weber & Brusilovsky, 2001).

CIMEL ITS (Collaborative, Constructive, Inquiry-based Multimedia E-Learning) is a system for object-oriented programming (OOP). Its proponents found it necessary to create an ITS because it was difficult to teach OOP in a classroom setting. The teacher has to know what the student is thinking to be able to pinpoint the problem. It was also reported that students with private tutors learn four times faster than students in a classroom setting. Hiring a tutor for every student who wants to learn OOP, however, is not cost-efficient. The next option would have to be an ITS (Wei, Moritz, Parvez, & Blank, 2005). CIMEL ITS, however, does not have a collaboration feature to allow interaction between students. On the other hand, POOLE II (Programmer’s Object-Oriented Learning Environment II) is a system with such a collaboration feature. It also has ILE components in a form of a learning companion which guides students through the lessons. However, this learning companion does model student’s knowledge. (Chan et al, 2005) recommended adding a student model to improve the interaction between the user and the learning companion.

A student model is an abstract representation of the learner in a system (Holt, Dubs, Jones, Greer, 1994). Student models in ITSs allow it to simulate the ability of a teacher to adapt to the specific needs of students during the teaching process. The student model is one of the key components of an ITS (Brusilovsky, 1992). It also facilitates one-on-one tutoring ability of the system by “keeping track of student’s proficiency in the domain” (Beck, Stern, & Woolf, 1997). By adding this component to
POOLE II, the learning companion could change styles to better interact with the student.

Because of the popularity and effectiveness of collaborative learning, a natural question in relation to student modeling will be how to model groups of students. According to Paiva, as quoted by Tongchai and Brna (2005), group modeling is “a way of capturing the aspects that identify a group as a whole”. It also includes the group’s beliefs, actions, goals, misconceptions, and conflict differences between an individual and the group. The group student model (GSM) used in (Kinshuk, Han, Hong, & Patel, 2001) contains information “extracted by a cumulative analysis of the individual student models”. The GSM bases its cumulative analysis on each student’s domain competence and behavior. A student can belong to more than one group. These GSMs are updated every time the program is run. According to the authors, “the mutuality of the group student model and the individual student model highly improves the effectiveness and accuracy of the system’s adaptivity”. Collaboration helps students learn faster because they learn from each other. “Learners can better improve their knowledge while learning with peers than learning individually (Tongchai & Brna, 2005).” Therefore, a GSM can be used to rate group performance or analyze events during collaboration, capturing the group’s essence as a whole and determining how much the group has learned.

Because work in POOLE II can be done in groups, not only should it model individual student’s knowledge, but group knowledge as well. A GSM should be implemented. Just like a teacher supervising group activities, the creation of such a model would allow interaction of the learning companion with a group of students, enabling it to outline the mistakes of a group and discuss it to that group. Therefore, an implementation of both an individual and a group student model would allow the representation of individual and group student knowledge for use in dynamic system adaptation.

The paper is organized as follows. Section 2 discusses the proposed approach while Section 3 presents how the model was used in the system. Section 4 discusses test results, and section 5 concludes the paper.

2. PROPOSED APPROACH

Several student modeling techniques had been considered, such as overlay, perturbation, differential, genetic graph (Holt et al, 1994), Bayesian network (Jensen, 2001; Charniak, 1991; and Henze et al, 1999) and fuzzy logic. Bayesian Networks have been used to provide a framework for uncertainty in artificial intelligence research (Ziv, 1997). It has been used by systems such as Andes (Conati et al, 2002), which deals with Newtonian Physics and BITS (Butz et al, 2006), which is a web-based tutor for C++. Since Bayesian Networks "give certainties for events that are not observable or only observable at an unacceptable cost (Jensen, 2001)", it was selected to represent the student in POOLE III for the following reasons.

First, the probabilities in the network’s nodes will be needed for providing individualized feedback. During class diagram activities, a wizard gives solicited hints based on the student model. Previous and future modules will be considered for the current exercise taken by the student. Second, the system requires all students’ grades to recommend group mates for a certain student. Recommendations are made using a modified Hoppe’s (1995) rule to complement the student’s performance. Lastly, the system needs to keep track of the student’s improvement upon moving on to another lesson. After an exercise is answered in one lesson, the previous lessons’ probabilities will be recomputed accordingly, and the predictions for the future lessons will be recomputed as well.

A sample, partial Bayesian network is shown in Figure 1. Each node shows a concept to be learned, and its relationships to other concepts as indicated by its edges. This network (i.e., its structure and initial probability values) was created based on interviews of faculty members teaching object-oriented programming at the College of Computer Studies, De La Salle University. The first node in the network is initialized based on the user’s test score in an exercise on procedural programming given by POOLE III to the student.

![Figure 1. Partial Bayesian Network](image_url)
3. USAGE OF THE MODEL

The student solves exercises for different lessons in POOLE III. Most exercises given are class diagram creation in UML. POOLE III supports the objects-first approach in learning object-oriented programming, i.e. the student needs to learn and master object-oriented analysis and design to help him learn object-oriented programming easier.

Figure 2 shows the interface used by the student to answer UML design exercises. POOLE III can evaluate the user’s solution, give him a grade and detect missing/incorrect components from his solution as compared to the correct solution. The result of his exercises is used to update his individual model.

POOLE III also allows the student to work with a group of co-learners. “Learners can better improve their knowledge while learning with peers than learning individually (Tongchai & Brna, 2005).” In this case, the student can choose from other students who are also logged-in to invite them to solve an exercise with him. POOLE III. The result of the group’s exercise is also used to update the group’s model. The reader is referred to (Aurellano, et. al., 2007) for a more detailed discussion of the individual and group models. Figure 3 shows the interface of POOLE III when the whiteboard is used by a group of students.

Once the model has been created, it is used to support a variety of system functions in POOLE III. For instance, it is used by the Wizard to provide appropriate feedback to the student. Previous POOLE systems (Chan, et. al., 2005) only give the student a grade based on the UML diagram that was automatically evaluated. POOLE III is able to provide feedback based on the student’s score in the exercise. The Bayesian Network allows the wizard to identify a possible weak concept, i.e. a cause of the student’s mistake, make suggestions on what next move to make (study a lesson, solve an exercise, among others), and make predictions about what future lessons may be affected because of his knowledge of the current topic.

Based on the individual student model, POOLE III can also make recommendations to the student to form an effective group to facilitate learning. POOLE III bases his recommendation on the models of the other students currently logged-in the system. However, the student has control over which becomes member of the group he wants to form. He can select from POOLE III’s recommendations and decide which among these co-learners will become part of his group.

POOLE III also asks the student to view his individual model. The student may then agree or disagree with POOLE III’s assessment of his model. This is called open student modeling. Open student models allow students to reflect on their performance and (hopefully) make action plans to rectify poor performance, and proceed with the current strategy in case he is performing well.

Figure 4 shows a sample student record book. This illustrates the student’s model. It shows the student’s grade for exercise he has already taken and for lessons he has already viewed. It shows the system’s confidence of his knowledge of a concept.
Figure 4. The Open Student Model

Figure 5 shows a sample screen shot of the open student model. Here an achievement map is shown. The achievement map shows the student’s performance in relation to his other co-learners.

4. TEST RESULTS

The POOLE III system was deployed in a computer laboratory at De La Salle University-Manila, for user acceptance testing by students enrolled in an introductory course for object-oriented design and programming in Java for the 1st Trimester of SY 2007-2008. A total of thirty-four (34) students tested the software.

Testing was conducted during two class sessions of the two OBJECTCP classes. The students went to the testing room during their class time and answered individual and group exercises. They also viewed their grades and reflected on the feedback given by the system. Lastly, they answered the user acceptance test.

The questionnaire included question items on the wizard’s feedback, group whiteboard, and the open model. A set of statements with the student answering if he strongly agrees, agrees, neutral, disagrees and strongly disagrees is the basic format of the questionnaire.

Majority of the students who used the system agreed that the Wizard was useful in helping them learn the lesson. They specifically valued the accurate feedback provided by the wizard after it evaluated their UML solutions. Because of the accurate feedback, the students reflected on their performance during the exercise. The lowest score the wizard got was on question 8, that it was unable to answer the student’s questions sufficiently. A majority of the students failed to use the hinting facility so that some of them answered neutral, disagree or strongly disagree for this item.

Table 1. Test Results of POOLE III

<table>
<thead>
<tr>
<th>Wizard Feedback</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. The wizard gave me a good idea of how I am progressing in learning the lesson.</td>
<td>52.94%</td>
</tr>
<tr>
<td>2. The wizard’s hints helped me in answering the exercises.</td>
<td>50.00%</td>
</tr>
<tr>
<td>3. The wizard provided accurate feedback on the results of my exercises.</td>
<td>61.77%</td>
</tr>
<tr>
<td>4. The wizard’s feedback helped me reflect on my performance in learning the lesson.</td>
<td>67.65%</td>
</tr>
<tr>
<td>5. The wizard’s recommended individual exercises were appropriate for my level.</td>
<td>64.71</td>
</tr>
<tr>
<td>6. The wizard’s assessment of my knowledge of the topic is correct.</td>
<td>55.88%</td>
</tr>
<tr>
<td>7. The wizard’s feedback was clear and understandable.</td>
<td>64.70%</td>
</tr>
<tr>
<td>8. The wizard can answer my questions sufficiently.</td>
<td>38.23%</td>
</tr>
<tr>
<td>9. The wizard is a very helpful guide in the system.</td>
<td>52.94%</td>
</tr>
<tr>
<td>10. It was very easy to communicate with the wizard.</td>
<td>47.06%</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Collaboration Tool</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. The group mate recommendation function was helpful in knowing who I can best work with to improve my learning.</td>
<td>85.72%</td>
</tr>
<tr>
<td>2. The recommended group mates complemented my knowledge.</td>
<td>71.43%</td>
</tr>
<tr>
<td>3. Recommending group mates to complement my weaker points helps me to learn more.</td>
<td>100%</td>
</tr>
<tr>
<td>4. My group was able to work well because of our combined skills.</td>
<td>57.14%</td>
</tr>
<tr>
<td>5. The collaboration tool helped me study with my friends.</td>
<td>100%</td>
</tr>
<tr>
<td>6. The collaboration tool encouraged exchange of ideas to solve the exercise.</td>
<td>85.72%</td>
</tr>
<tr>
<td>7. Answering exercises as a group is better than answering by myself.</td>
<td>100%</td>
</tr>
<tr>
<td>8. I feel that I have improved my skills after working with a group.</td>
<td>85.71%</td>
</tr>
</tbody>
</table>
POOLE III’s collaboration tool includes the group recommendation system and group modeling. Based on the test results, the students agreed that the collaboration tool was very helpful in learning the lesson. They all agreed that working with the right group of co-learners via the group whiteboard facility allowed them to answer the exercise and learn from their peers. There lowest score the collaboration tool got in the questionnaire was 57.14%. Students were asked if the group was able to work well because of the group mate’s combined skills. This might be caused by the fact that there were occasions when unexpected bugs occurred and they had to use the same computer for answering a group exercise.

Students had mixed reactions seeing POOLE III’s representation of their knowledge. A majority of them viewed the assessment of POOLE III as accurate of their knowledge. They were also motivated to work harder seeing how they fare against other students. The graphical representation of the open model was also easy to understand. A majority of them agreed that knowing their performance (thru the model) helped them plan for future activities to better learn the lesson.

5. CONCLUSION

The objective of this research is create and maintain an individual and group student model in a web-based collaborative learning environment for learning object-oriented design and programming in Java, POOLE III. POOLE III makes use of a Bayesian Network to represent student’s knowledge of learning object-oriented (OO) design and programming in Java.

A Bayesian Network was manually created based on repeated interviews with experts teaching a course on introductory OO design and programming at De La Salle University-Manila. Initial values used come from a pre-test on procedural programming.

Results from student evaluation show the value of the student model. The wizard was able to provide more informed and adapted feedback since it consults the student model to decide how the student has performed, what activities to suggest and what hints to provide (in case the student gets stuck solving an exercise). The collaboration activities improved because the student can choose from a set of students recommended by the system. POOLE III uses each student’s model to decide on which students to recommend as groupmates of the student. Ideally, the capability of the group to answer an exercise is based on balancing their knowledge set, i.e. one student’s weakness should be compensated by a possible groupmate’s strength. The open model, on the other hand, encouraged the student to reflect on his performance and strategize to improve his performance for future activities.

6. REFERENCES


