COMPUTER-BASED LEARNING SOFTWARE FOR ENGINEERING STUDENTS

G. J. Poitras¹, E. G. Poitras²
¹Faculté d’ingénierie, Université de Moncton, N.B., Canada; ²Faculty of Education, McGill University, QC, Canada
¹gerard.poitras@umoncton.ca; ²eric.poitras@mail.mcgill.ca

Abstract —The rapidly evolving technological knowledge required by engineers imposes important challenges on how to best prepare the engineering workforce of the future. Novice engineers are faced with formulating creative solutions to problems they have never seen before; they did not really learn to solve “real world messy problems”. Problem-solving in engineering requires the use of disciplinary-based heuristics that are not immediately apparent to novices in the field. The aim of this study is to evaluate and expand engineering education using cognitive apprenticeship as an instructional framework. Cognitive apprenticeship outlines pedagogical strategies that makes visible experts’ problem solving heuristics to support novices in becoming more proficient. This work will provide an overview of a computer-based learning environment in a classroom with third- and fourth-year undergraduate students. First, we discuss the implementation of cognitive apprenticeship in the classroom and evaluate the learning outcomes of students with individual differences in learning styles. Second, we apply cognitive apprenticeship principles to guide the design of a Civil Engineering tutor, a computer-based learning environment that serves as a cognitive tool outside the classroom. The design of the software is driven by a cognitive task analysis of several knowledge areas that mediate competency in calculating the snow loads on buildings. This teaching tool includes the main assumptions of the cognitive apprenticeship approach and also defines instructional methods for enhancing learning. We analyze user satisfaction towards the different features of the software and derive some implications for future innovative educational tools.

Keywords: Cognitive tools, Cognitive apprenticeship; Cognitive tutor, Civil Engineering, Computer-based learning environment, Problem-based learning.

1. INTRODUCTION

One of the most important goals of engineering education is to support the acquisition of knowledge that is relevant to practice [1-2], ensuring that post-secondary training focuses on core professional competencies. The value of training effectiveness in this field is essential, given that members of the civil engineering and construction industry report that education is an important factor in preventing structural failures and promoting public safety [3]. As such, instructors have addressed this challenge by using pedagogical practices that aim to better articulate and explain the hidden thought processes that mediate problem-solving [4-6].

This paper will address an approach to guide the design of learning technologies based on cognitive apprenticeship teaching practices, a well-known instructional method used in teaching professional competencies [7]. This theory-driven approach was applied to design the Civil Engineering Tutor, a computer-based learning environment that trains novice engineers to solve structural design problems pertaining to snow loads. The Civil Engineering Tutor provides students with several roof configurations that are commonly encountered in practice. In doing so, students practice solving problems with the tutor and receive feedback.

In the following sections, the method that was used to identify the cognitive competencies that mediate proficiency in calculating snow loads on building structures is presented. The design guidelines of the Civil Engineering Tutor are reviewed, and user satisfaction, a measure of how system features meet the students’ expectations is evaluated. The challenges that were met in implementing cognitive apprenticeship to assess cognitive competencies and adjust instruction are also raised in the fourth section.
2. DESIGN GUIDELINES OF THE CIVIL ENGINEERING TUTOR

The Civil Engineering Tutor is a computer-based learning environment designed to support students in solving structural design and load problems. The tutoring system interface consists of a three-paned main window, as shown in Fig. 1. In the tutor panel, an artificial pedagogical agent provides students with tutoring during problem-solving. Example problems for calculating snow loads are presented with pre-determined roof snow load configurations displayed in the problem window. Students use the tools that comprise the solution workspace in order to access the resources that are needed to practice problem-solving and receive feedback. Each tool is featured in a separate panel and has a distinct function.

The first panel establishes the relevant problem-solving steps, using textboxes to define the relevant variables and known parameters that characterize the problem. Answers are color-coded in order to distinguish correct responses (i.e., green), incorrect responses (i.e., red), and known parameters (i.e., blue). Students submit their responses to receive corrective feedback from the system, while the pedagogical agent provides hints that are meant to support students in finding the correct answers.

The second panel provides students with access to instructional materials. These materials include the PowerPoint slides that were used in the classroom by the instructor. These slides are organized according to topics, which students can use to access the relevant material by clicking on a hyperlink. These topics cover a wide range of factual and conceptual information pertaining to the meaning and pre-determined values of formulas and their coefficients.

The instructional videos panel allows an instructor to demonstrate the procedure that needs to be followed in order to solve the problem. The videos include a voiced over narration with an animated step-by-step instruction in how to solve similar problems. However, the known parameters of each problem are slightly different from the one the student has to solve. Once the correct sequence of steps has been demonstrated, students apply the procedure to solve an analogous problem.

The instructional examples panel is delivered to students after submitting a solution. The example provides a worked out solution, where the calculations for each steps are shown to students. In doing so, students are able to compare their own solution process to the correct sequence of steps, and to identify discrepancies and areas of misunderstanding.
3. ASSESSING AND TRAINING COGNITIVE COMPETENCIES

The Civil Engineering Tutor represents cognitive competencies as conditional, procedural, and declarative knowledge [8]. Table 1 provides some examples of how knowledge is characterized by the system. This form of knowledge representation was adapted from research on skill modeling with intelligent tutoring systems [9-11]. The system is currently under development with the assistance of domain experts in order to be capable of updating these representations as students are interacting with the learning environment. In doing so, the system will be able to make inferences in relation to the acquisition of cognitive competencies on the basis of student performance.

*Conditional knowledge* defines when to use procedures to solve classes of problems, and under what conditions certain procedures are applicable [8]. *Procedural knowledge* consists of techniques and heuristics that are used to solve a particular class of problem [8]. As such, conditional knowledge allows students to determine when to use certain skills, such as when a structure exhibits a certain set of characteristics. Procedural knowledge enables students to engage in the series of actions that are needed to solve the problem, such as the correct sequence of steps involved in calculating snow loads for a particular structure. For example, Table 1 designates conditional and procedural knowledge as a sequence of condition-action rules. These rules are mapped onto user actions and answers that are detected by the system.

*Declarative knowledge* refers to factual and conceptual information pertaining to the domain [8]. Declarative knowledge includes facts such as the values of coefficients of local risk for snow load, slope, and roof shape. As an example, Table 1 shows declarative knowledge as designated by conditions that restrict both the outcomes of possible student actions (i.e., relevance conditions, as shown in C\textsubscript{R1}) and student answers (i.e., satisfaction conditions, as shown in C\textsubscript{S1}). The relevance conditions are matched with the conditional rules, while the satisfaction conditions are mapped onto the procedural rules. In doing so, the system is able to track the acquisition of specific classes of cognitive competencies and make necessary adjustments to instruction.

To teach cognitive competencies directly, instructors can apply cognitive apprenticeship teaching methods to facilitate learning inside the classroom, which can also be supported through the use of technology [7]. Instructional practices can be implemented to design classroom activities and software features with the aims of facilitating the acquisition of the core cognitive competencies.

<table>
<thead>
<tr>
<th>Table 1: Representation of cognitive competencies.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Correct Conditional Rule</strong></td>
</tr>
<tr>
<td>R\textsubscript{1,1} IF goal is to find the slope coefficient (C\textsubscript{s}) and snow can accumulate on roof, THEN calculate the slope of the roof (α)</td>
</tr>
<tr>
<td><strong>Correct Procedural Rule</strong></td>
</tr>
<tr>
<td>R\textsubscript{1,1,1} IF the slope value (α) is less than 30 degrees, THEN the slope coefficient (C\textsubscript{s}) is equal to 1.0</td>
</tr>
<tr>
<td><strong>Incorrect Procedural Rule</strong></td>
</tr>
<tr>
<td>R\textsubscript{1,1,2} IF the slope value (α) is between 30 and 70 degrees, THEN the slope coefficient (C\textsubscript{s}) is equal to (70-α)/40</td>
</tr>
<tr>
<td>R\textsubscript{1,1,3} IF the slope value (α) is more than 70 degrees, THEN the slope coefficient (C\textsubscript{s}) is equal to 0</td>
</tr>
<tr>
<td><strong>Incorrect Conditional Rule</strong></td>
</tr>
<tr>
<td>R\textsubscript{1,2} IF goal is to find the slope coefficient (C\textsubscript{s}) and snow cannot accumulate on roof, THEN calculate the slope of the roof α</td>
</tr>
<tr>
<td><strong>Declarative Conditions</strong></td>
</tr>
<tr>
<td>C\textsubscript{R1} The slope value (α) is known and snow can accumulate on roof, and the student has calculated the slope coefficient (C\textsubscript{s})</td>
</tr>
<tr>
<td>C\textsubscript{S1} The slope coefficient (C\textsubscript{s}) is equal to 1.0</td>
</tr>
</tbody>
</table>

Cognitive apprenticeship is an instructional approach where cognitive competencies are taught in the context of performing authentic tasks that are relevant to professional practice [12-13]. Classroom activities are designed to simulate real-life situations, describing the problem in terms of how it is likely to arise for a practitioner working in the local area. The goal of the instructional activities is to solve a structural design problem in collaboration with other class members.

In both the classroom activities and software features, the complexity of each instructional objective can be gradually increased as students gain knowledge. Problems vary considerably in the amount of cognitive competencies that are required to solve them. Furthermore, the characteristics within specific classes of problems can be varied to increase the diversity of situations that are addressed. This aspect of instruction ensures that cognitive competencies are taught to students with the aim of transferring them to other future problem-solving situations.

Instruction should promote a community of practice, wherein students are able to collaborate and exchange best practices in solving problems [12-13]. The classroom is thus organized in groups of three to four students. Students are first instructed to read the relevant material before class in order to allocate more time to problem solving inside the classroom. The instructor reviews the class material and introduces the problem and the expected outcomes. Group leaders are responsible for ensuring that task objectives are met and for asking...
questions to the instructor. Students reflect on the benefits of different solution approaches by discussing within each group. The solutions of each group as well as relevant issues that were raised during the class activity are shared across class members and differences are discussed.

Cognitive apprenticeship outlines several techniques to articulate, monitor, and adjust students’ thought processes during problem-solving [12–13]. The Civil Engineering Tutor provides modelling and coaching while solving problems. The software is designed to demonstrate the use of certain techniques (i.e., instructional materials and videos) and provide feedback when students make errors (i.e., instructional feedback). The Civil Engineering Tutor also allows students to explore more novel problems outside the classroom. This is meant to facilitate the role of the instructor, which is to articulate students’ knowledge in order to provide scaffolding. The instructor prompts groups of students in explaining the solution process, and then provides help when required.

4. EVALUATING USER SATISFACTION WITH THE CIVIL ENGINEERING TUTOR

A sample of 15 participants (Male = 11, Female = 4) with an average age of 20.4 (SD = 1.0) volunteered to complete the study. The recruitment was conducted by the instructor during class. Participants were in the third or fourth year of completing a bachelor’s degree in civil engineering (M = 3.6, SD = 0.9).

User satisfaction was assessed using self-report measures that were translated in French and adapted to the specific features embedded in the Civil Engineering Tutor. The questionnaire included 29 items on a 10-point rating scale from the Questionnaire for User Interface Satisfaction (QUIS) [14] as well as 50 items on a 5-point Likert scale from the Report on the Enjoyment, Value, and Usability of an Intelligent Tutor (REVU-IT) [15]. The QUIS items assess user satisfaction in terms of a global reaction (e.g., difficult vs. easy), the organization of information in the main user interface (e.g., complex vs. clear), the clarity of the terminology and instructions (e.g., complex vs. clear), the learning experience and task performance (e.g., never vs. always simple), as well as the system capacity (e.g., too slow vs. fast enough). The REVU-IT items assess the ease in which students interacted with the tutor (e.g., finding information with the tutor is simple and quick). The questionnaire also assessed the overall impression of the system (e.g., the learning experience with the tutor was satisfactory). The questionnaire also examined satisfaction in relation to specific features of the system, including the instructions (e.g., I would have appreciated more instructions on how to use the tutor), learning activities (e.g., it was difficult to learn in this environment), problems (e.g., the problems were well designed to illustrate the concepts), materials (e.g., it was easy to navigate through the professor’s notes), videos (e.g., the videos were useful and interesting), and menus (e.g., the menus were easy to use).

The following procedure was used to assess user satisfaction in relation to the design guidelines of the Civil Engineering Tutor. First, the participants attended a lecture where instructions were provided on how to use the software to calculate snow loads on different types of structures. The students had previously attended courses that addressed snow load evaluations the year before. After the class, participants were provided with a questionnaire and a USB disk that contained the software installation package. Participants were instructed to install the software in order to solve three different classes of problems pertaining to snow loads on structures. Participants then reported their satisfaction with the system by completing the questionnaire.

Overall, the results obtained from the self-report measures indicate that participants were satisfied with the system. The average degree of satisfaction reported on both the QUIS and REVU-IT were on the upper end of each measurement scale, (M = 8.01, SD = 1.29, Range = 1-9, and M = 4.12, SD = 0.97, Range = 1-5, respectively). This pattern remained relatively consistent across each sub-scales of both measures. After completing the task, the participants reported being both familiar with calculations of snow loads on structures (M = 4.3, SD = 0.8, Range = 1-5) as well as confident in their abilities to solve problems correctly (M = 3.8, SD = 0.8, Range = 1-5). As such, the Civil Engineering Tutor allowed students to practice problem-solving and become more confident.

The average values of each item in the REVU-IT and QUIS measures of user satisfaction were ranked in ascending order. This was carried out in order to identify the items where users responded as being the least and most satisfied. The values of the items with a negative wording were inverses for the purposes of the comparison. The fifth lowest and highest ranked items for each measure are shown in Fig. 2 and 3.

On the one hand, the results show that students were most satisfied with the ease in which calculations are accomplished in the system. The information required is readily available and the videos are reported as very useful. The system itself seems to perform well, reacting quickly to user interactions. The steps involved in the calculations seem to be clear to the users, as well as the terms and menus that are featured in the system. Overall, the students reported that they were satisfied with their experience and that they would use a similar system to practice other types of load evaluations in the future.
Fig. 2. REVU-IT scales for user satisfaction in relation to the system features of the Civil Engineering Tutor on a 5-point Likert scale (1 – Strongly Disagree, 5 – Strongly Agree) with negative item values inversed.

Fig. 3. QUIS scales for user satisfaction in relation to the system features of the Civil Engineering Tutor on a 9-point bipolar rating scale.
On the other hand, the students reported that there is a need to clarify the information and instructions that are provided to students, and how the tutor interacts with them. For example, the participants were moderately satisfied with the amount of instructions on how to use system features, the task demands, and the nature of the feedback. Moreover, the students also reported a lack of engagement in that the software seemed artificial and distinct from real-life situations, and that there was no opportunity to choose amongst a wide range of possible activities. This was evident from the range of mild negative reactions that were reported by the participants as well as their different preferences towards instruction.

5. CONCLUSION

This paper has outlined a method to identify the cognitive competencies that mediate proficiency in calculating snow loads on building structures. The cognitive competencies model and the cognitive apprenticeship teaching practices that guided the design of the Civil Engineering Tutor were reviewed. The user satisfaction evaluation was promising in that students reported an overall high level of satisfaction with the current system features as well as confidence in their abilities to solve problems. Several areas where highlighted where improvements can be made related to how instruction is provided to students, namely, the clarity of the information that is delivered to students as well as the level of engagement that is experienced with the system. As such, this study has broader implications for researchers that wish to improve the design of instructional systems in the later stages of development.

References


