Journal of Visual Languages and Computing 25 (2014) 1021–1029

Contents lists available at ScienceDirect

Journal of Visual Languages and Computing

journal homepage: www.elsevier.com/locate/jvlc

Computer tutors can reduce student errors and promote solution efficiency for complex engineering problems ☆

Paul S. Steif, Matthew Eicholtz, Levent Burak Kara *

Department of Mechanical Engineering, Carnegie Mellon University, Pittsburgh, PA, USA

ARTICLE INFO

Article history:
Received 30 September 2014
Accepted 5 October 2014
Available online 14 October 2014

Keywords:
Concept inventory
Interactive learning
Measures of knowledge
Pre-post tests
Statics
Web-based courseware

ABSTRACT

An ability to solve complex problems, for which a variety of solution paths are possible, is an important goal in engineering education. While feedback is critical to learning, hand grading of homework rarely provides effective, timely feedback on attempts to solve complex problems. Such feedback is also unfeasible in distance education contexts. A technology, based on the approach of cognitive tutors, is presented as a generally applicable method of providing automated feedback on complex problem solving, with truss problems studied in engineering as an example. The tutor maintains a cognitive model of problem solving for this class of problems, and associates various solution steps with distinct skills or knowledge components. One can determine whether students learn individual skills by measuring the error rate as a function of practice. Prior work has shown that for many skills the error rate indeed decreases with practice. New insight into the tutor's effectiveness, pertaining to the efficiency of student solution paths, is presented in this paper. While no explicit feedback is given regarding solution efficiency, it is found that students using the tutor become more efficient with practice. Furthermore, more efficient paths are found to be associated with making fewer errors.

1. Introduction

The development of problem-solving skills is a cornerstone of engineering education. While some problems that students learn to solve are simple, utilizing a single concept or principle, more complex problems are undertaken even in lower division courses. Students may need to coordinate and organize several concepts and steps, and many pathways to correct answers may be possible.

It is recognized in general that learning of any new skill is promoted by timely and effective feedback [1–4]. The opportunity for feedback on complex problem solving traditionally occurs through grading of handwritten homework. With weeklong turnaround such feedback is virtually never timely, nor is it readily made *effective*. Solutions can vary from one student to another, and with an incorrect answer it is laborious for graders to identify and communicate to the student how the solution deviated from a correct path. Further, in a distance-education setting, hand grading would be largely unfeasible.

This paper describes a technology that can provide students learning to solve complex engineering problems feedback on their efforts. The technology must be able to follow and assess student solutions for a variety of pathways pursued. To that end we adapt the approach of cognitive tutors, which have been developed for computer programming [5], math [6,7], and other fields. Such tutors are based on a cognitive model for a learner encountering the chosen tasks, and so can potentially provide feedback for a range of solution pathways. There do not appear to be previous efforts to devise cognitive tutors to assist students with complex engineering problems. The feasibility

☆ This paper has been recommended for acceptance by Shi Kho Chang.

* Corresponding author. Tel.: +1 412 268 2509.
E-mail address: lkara@cmu.edu (L.B. Kara).

http://dx.doi.org/10.1016/j.jvlc.2014.10.012
1045-926X/

of a cognitive tutor style approach to providing feedback on complex engineering problems has been demonstrated [8] through a tutor focusing on truss problems, which are commonly studied in mechanical and civil engineering. This new application of cognitive tutors is distinct from previous tutors in that student work involves the coordination of multiple open-ended, student-initiated vector diagrams and equations, all of which are interpreted on the fly in terms of a set of skills needed to properly solve the class of problems.

In the present paper, we consider in more depth the solution path taken by students solving problems with the tutor. When the tutor gives feedback on individual errors it does not prompt students, except for rare circumstances, to think about whether their overall solution strategy is efficient. However, one can speculate that more efficient strategies may lead to fewer errors and that, in the course of solving problems, students may discover such solution strategies on their own. Here efficiency relates to the maximum number of unsolved, yet defined, variables at any given time in the solution path. Using this definition, we investigate whether solution efficiency changes with practice and whether higher efficiency is associated with lower propensity for errors.

2. II. Description of tutor for trusses

Since use of the tutor is intended to ultimately lead to success in solving problems with paper and pencil the user interactions with the tutor should be as unconstrained as possible, provided the tutor maintains the ability to judge user work. Although progress continues to be made in computerized interpretation of completely freeform work, for example via writing with a stylus on a tablet [9–13], such technologies may be limited for the foreseeable future; we have therefore defined unconstrained as still within the confines of a mouse and keyboard user interface.

Fig. 1 displays a typical truss problem as it would appear in a textbook. The problem consists of a set of pins (dark circles) and connected bars. There are specified forces (10 kN) and supports (idealized constraints that keep the pins in position). Fig. 2 shows a portion of a solution to the problem in Fig. 1; user input corresponding to such solution elements must be enabled by the technology. A portion of the truss (a subsystem) including

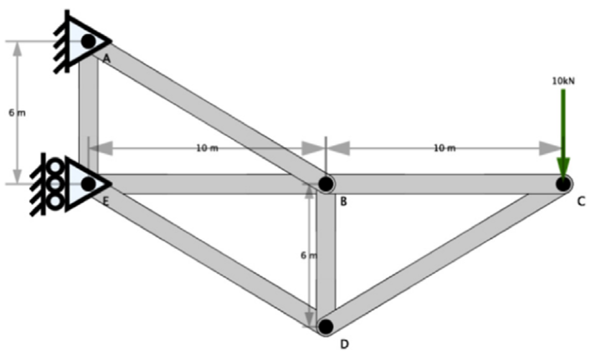

Fig. 1. Typical truss problem, in which forces within the bars (members) are to be determined.

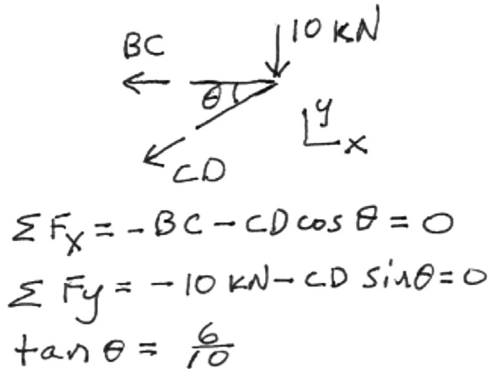

Fig. 2. Portion of handwritten solution to problem in Fig. 1, showing the free body diagram of the pin at C and its connected partial bars, and associated equilibrium equations.

point C has been singled out for attention, the unknown and known forces drawn on the diagram (a so-called free body diagram or FBD), and equations of equilibrium (imposing Newton's laws of motion) have been written. In solving truss problems, students select multiple portions of the truss and for each subsystem draw free body diagrams and write equilibrium equations. Students must also organize the solving of equations and interpret results physically in terms of the original truss. The solver can choose any portion of the truss, write equations in any order, then choose any other portion, and so forth. The technology must grant the user latitude to pursue this large space of solution paths and still be able to judge and give feedback regardless of the path chosen.

Even within the confines of a mouse and keyboard user interface, there are a few additional intentional constraints on how closely students' actions with the tutor mimic paper and pencil solving. First, to reduce the cognitive load [14,15] associated with exercising skills already mastered by the student, certain tasks have been offloaded to the tutor; for example, we removed the need to enter numbers into an electronic calculator to obtain numerical solutions. Second, motivated by the self-explanation effect [16] in educational psychology that students who explained problems to themselves learn more, the tutor introduces selective highly targeted opportunities to make the student's thinking visible, thinking which is rarely visible in pencil and paper solving. Specifically, the tutor requests the user to designate each defined force as falling into one of several categories.

We assume that students using the tutor have learned about truss analysis through other means, such as lecture and textbook. The tutor focuses exclusively on helping students solve problems, allowing a solution process such as depicted in Fig. 2 to be conducted on the computer with as little constraint as possible, while maintaining the ability to interpret student work. Observations of student work and typical errors [17] solving truss problems have guided tutor design. The goal is to allow a student using the tutor to commit most, if not all, errors that are observed in pencil and paper solutions.

Based on an analysis of the required tasks to solve truss problems, informed by prior work on the concepts and skills needed in the overall subject in which trusses are