121st ASEE Annual Conference & Exposition
Indianapolis, IN
June 15-18, 2014
Paper ID #8509

# Development of a cognitive tutor for learning truss analysis

**Dr. Paul S. Steif, Carnegie Mellon University**

Paul S. Steif is a Professor of Mechanical Engineering at Carnegie Mellon University. He received a Sc.B. in engineering from Brown University (1979) and M.S. (1980) and Ph.D. (1982) degrees from Harvard University in applied mechanics. He has been active as a teacher and researcher in the field of engineering education and mechanics. His research has focused on student learning of mechanics concepts and developing new course materials and classroom approaches. Drawing upon methods of cognitive and learning sciences, he has led the development and psychometric validation of the Statics Concept Inventory – a test of statics conceptual knowledge. He is the co-author of Open Learning Initiative (OLI) Engineering Statics, and he is the author of a new textbook Mechanics of Materials, published by Pearson.

**Levent Burak Kara, Carnegie Mellon University**
**Dr. Luoting Fu**

# Development of a cognitive tutor for learning truss analysis

## 1. Introduction

Statics is an important foundational subject in many engineering majors. It is often a pivotal first course, in which many students weigh whether to proceed with an engineering major. In addition, learning of statics is also a concern to instructors in advanced level courses, such as capstone design, who would like students to be able to utilize what they learned in statics[1]. Conceptually, statics breaks important ground for engineering students. Drawing free body diagrams (FBDs) and imposing equilibrium in statics is an early foray into a style of analysis that is a cornerstone of engineering problem solving: isolation of a sub-system upon which balance laws are applied.

Students who complete statics should be able to effectively apply its methods to analyze realistic engineering systems. Some exposure to realistic systems comes from tackling problems often identified as "structures", that is, trusses, frames and machines (mechanisms), which are addressed to some extent in most statics courses. Problems based on structures entail consideration of multiple interacting bodies simultaneously and are challenging because: (i) they utilize much of the core conceptual knowledge in statics, which is always more difficult to draw upon while solving problems, and (ii) they involve juggling several sub-analyses, and hence demand organization, strategy, and decision-making.

In typical classroom instruction, the basic concepts and approaches underlying the analysis of problems involving structures are presented, along with the solution of example problems. However, students then need to practice solving such problem themselves. It is widely recognized across a wide range of domains, that timely feedback during the practice of problem solving promotes learning[2-7]. While a human tutor can offer effective instruction during problem solving, offering such close monitoring via computer could benefit a greater number of students.

Many instructional interventions have been designed for statics, although computer-based interventions typically offer the most effective feedback when students exercise individual concepts or analysis skills. Recently, there have appeared computer systems that allow students to work on some simple statics problem more or less from start to finish, and provide feedback on individual steps[8,9]. There is also recent work on alternative user interaction modalities for problem solving in mechanics. A Truss Recognition tutoring system is being developed[10] in which students draw trusses with supports and loads using a pen. The strokes are recognized and interpreted by the system, which then analyzes the trusses to determine the loads in the bars. Sketch recognition is employed in another project[11] to develop a tutoring system for statics that allows students to solve problems by pen.

To give feedback to a user who can pursue various pathways in problem solving, a computer tutor must have a model of the thought processes needed to solve problems. Indeed, researchers have developed Intelligent Tutoring Systems[12,13], including even some relevant to the mechanics of structures[14-16]. Intelligent computer-based instruction can be effective in increasing student learning beyond that achieved in normal classroom instruction[17]. Cognitive tutors[18] merged the ideas of intelligent tutoring systems with computational models of cognitive theories of human learning, memory, and problem solving[19]. Cognitive tutors are based on a cognitive model of a student interacting with problems in a domain.

Among possible structures problems in statics, truss problems involve a reasonably complex, but still manageable, combination of graphics and text. Furthermore, there are multiple pathways to a correct solution. The underlying framework for solving statics problems generally, including the common errors of students, has been studied[20-23]. This research enables us to formulate a simple cognitive model for solving truss problems and thereby interpret student work and offer instructional support in response. In this paper, we describe the development of a cognitive-style tutor for truss problems, and we identify the trade-offs between fully free, paper-and-pencil solving and the modestly constrained solving that the tutor permits. The constraints imposed by the tutor enable interpretation of student work, while still allowing the errors commonly committed in paper-and-pencil solving. Results for an initial cohort of students are presented.

## 2. Designing tutor based on tasks required and on observations of student work

A tutor for problem solving in statics should enable users to carry out the general set of tasks expected of students in this subject. In particular, the tasks in applying statics to a real physical system include:

1. Survey the physical system to recognize the various parts, how they are connected or supported

2. Select a subsystem, that is, a some portion of the system, for analysis

3. Draw a free body diagram of the subsystem

4. Write down equations of equilibrium for the subsystem

5. Solve equations for unknowns, interpret them, and potentially use those results in the analysis of subsequent subsystems

Different types of statics problems exercise these facets of the problem solving process to varying degrees. Certainly, the initial stage of surveying a physical system, the true modeling stage, can be the most difficult; it is also the most challenging for instructors to assign so students' efforts can be observed. In confining ourselves in this tutor to truss problems as they appear in textbooks, we accept that the problems are pre-modeled and focus on the remaining steps in statics.

To design a tutor that partially constrains the solver (does not allow free form sketching), we should consider what errors students commit in their free form attempts to analyze trusses. The tutor should grant the user latitude to commit most of the same errors. As an example, Figure 1 depicts a student who has written equilibrium for a portion of the truss, but because the forces have not been drawn on the FBD, the assumed directions of the internal forces are uncertain. In fact, the resulting forces have sign errors. Further examples of errors[24] include failing to separate a truss into parts, even though internal forces appear in equations, being unclear as to which members or partial members are included in a subsystem, and drawing internal forces when entire bars are present.

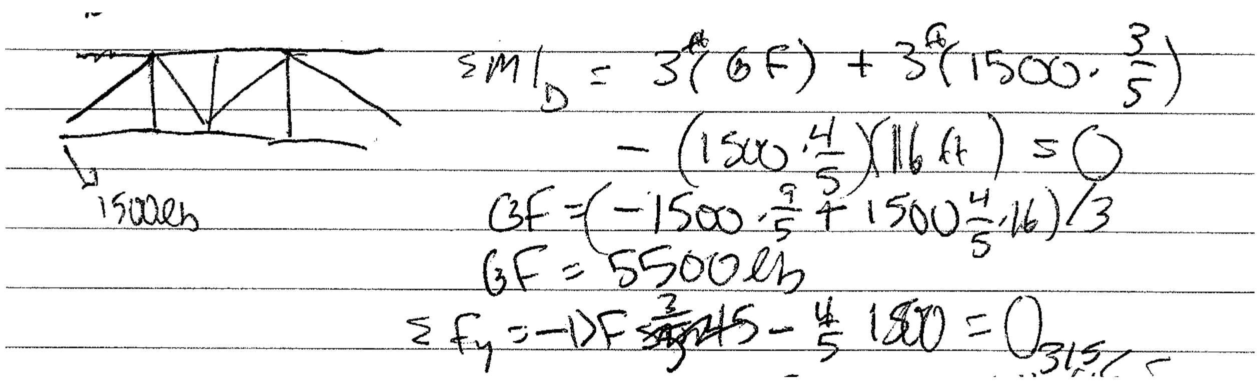


Figure 1. Example of typical student error: Internal forces (GF and DF) are not drawn on section, but appear in equilibrium equations; the solutions ultimately have sign errors.

While students commit many errors in truss problems, they do tend form subsystems by selecting pins, bars and partial bars, without other extraneous objects. In drawing free body diagrams, students do tend to draw forces on the pins or at the ends of partial members, but not in the middle of a member or at some random external point. Further, students tend to draw forces at those points, rather than concentrated moments or couples. (This is not to say that students fully understand why an idealized pin cannot exert a couple, or why the connections between truss members are idealized as pins.) The forces tend to be drawn parallel to x-y axes, or parallel or perpendicular to members present, but not in random directions. In writing down equations of equilibrium, students tend to write summations of forces in x and y directions, and they tend to write summations of moments about points coinciding with pins.

The above observations of student work inform how the tutor should constrain users in solving truss problems. We argue that a tutor that constrains user choices as follows will capture most student work (correct and incorrect) on truss problems:

- Each subsystem can be any collection of pins, members and partial members (there can be multiple such subsystems analyzed)

- In free body diagrams, only forces can be drawn, either at pins or at the free ends of partial members. Forces are confined to lie along x-y directions or parallel or perpendicular to bars, and may act in either sense.

- Equations of force equilibrium along x-y, and equations of moment equilibrium about any pin, can be written.

While students are free to carry out the actions just described, in devising the tutor we have made some non-obvious choices regarding: what constitutes a correct subsystem and what must be done to fully specify a free body diagram. These choices, which serve largely to make his or her thinking more visible to the student and to the tutor, are described in the next section.

In summary, we seek to develop a computer tutor with a simple, easily learned user interface that gives students reasonably wide latitude to solve truss problems using method of joints and method of sections with minimal distractions and unnecessary effort. Students using the tutor are expected to have learned about truss analysis through other means, such as lecture and textbook; the tutor focuses on helping students practice solving truss problems. The tutor should allow students to make the errors commonly committed by students when solving with pencil and paper, and provide guidance needed by students to correct such errors.

### 3. Description of tutor

A screen shot of the overall tutor, with a problem partially solved, is shown in Figure 2. The left half of the display contains a menu bar at the top and the problem diagram and statement. The problem diagram can be toggled to display the solution diagram, where support reactions and bar forces that have been determined are registered by the student, as described below. **The user chooses a subsystem for analysis by clicking on a set of pins, members, and partial members, and then clicking on the draw (pencil) icon from the menu bar.** The selected group of parts is added as another subsystem and would appear as one of the thumbnails to the right half of the display. Clicking on a thumbnail focuses on that subsystem, allowing the user to draw its FBD and write its associated equilibrium equations.

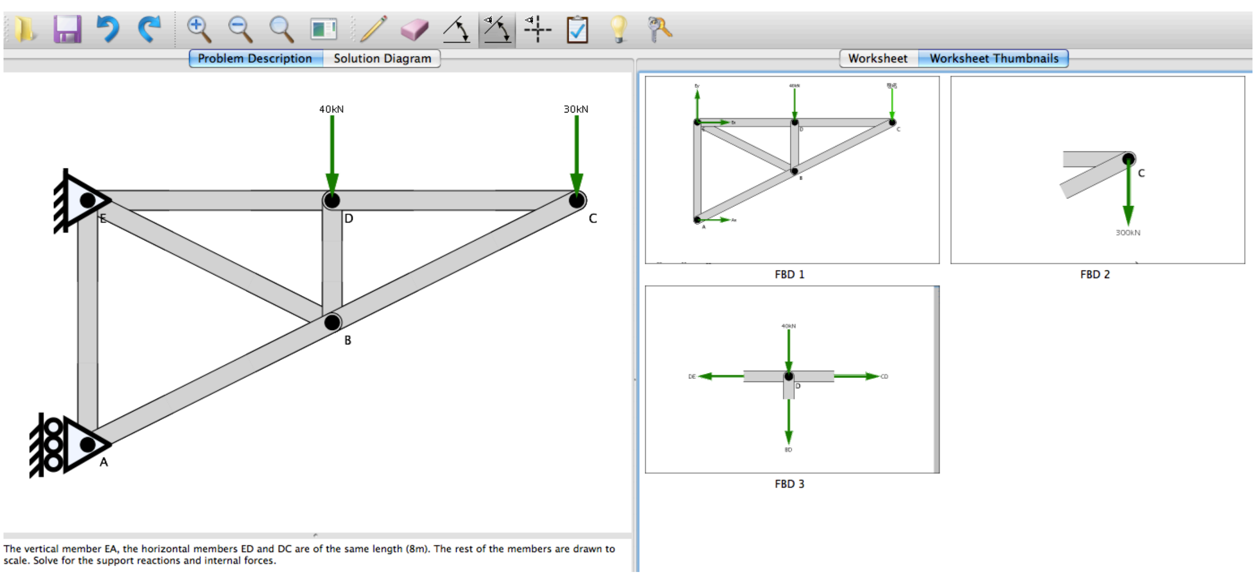

Figure 2. Screen shot of full display of tutor for trusses.

Note that the user can select any combinations of pins, bars, and half bars to form a system; most combinations would not be valid. To explain what the tutor treats as a valid subsystem, it is useful to note the practice in textbooks. With the method of sections, it is common to draw the partial members that have been cut, along with the whole members and pins to one side of the chosen section. With the method of joints, it is common to draw just the pin, perhaps as a dot, although some books draw the pin along with the partial members that connect to it. With the tutor we have chosen to insist on the latter method: a pin alone is not viewed as a valid subsystem, but must include the partial bars that connect to it (see Figure 3). There are two rationales for this. First, if the pin alone were drawn without the partial bars, the jumble of external and internal forces near just a pin would be difficult for the tutor to interpret. Second, showing partial bars, together with the method of specifying forces described below, is intended to help students develop a unified picture, relevant to both methods (joints and sections), as to the representation of the internal force in a member.

**The user creates the free body diagram by drawing forces on selected points, categorizing each type of force, and giving each a label.** In Figure 3, we show a subsystem with a pin and the two connected partial bars; a new internal force being added to a partial bar. In free form solving of trusses, students only draw arrows (for forces) and label those arrows with variables or numbers. With the tutor, we have also chosen to require the user to categorize each force being drawn; the choices, shown in the window labeled "Defining a force", include: known applied force, support reaction (unknown or determined), and internal force (unknown or determined). Depending on the force category chosen, a variable label or number is required. Even though a student in free form solving may not be thinking in terms of these categories, an expert, such as instructor, is surely clear when drawing a force which of these is being represented.

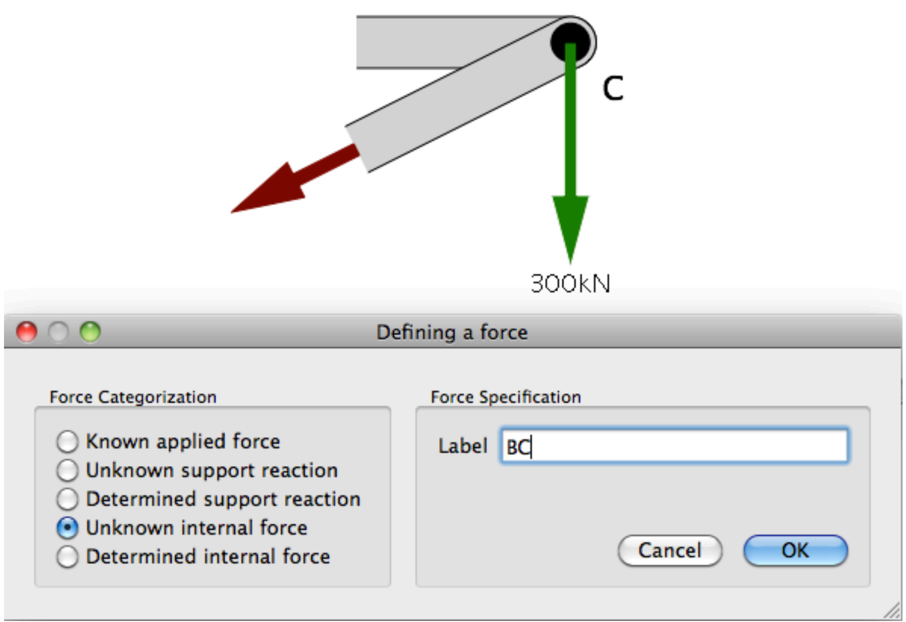

Figure 3. Screen shot of force being added to a free body diagram, showing force categorization.

Requiring force categorization, together with the insistence on including partial members and pins in a subsystem, provides two benefits: (i) it helps students organize their thinking about the various forces in a way that can carry over to paper-and-pencil problem solving after tutor use and (ii) it establishes some clear basis for the tutor to recognize errors in student work, namely that applied and support forces can only act at pins, and internal forces can only act at the ends of partial bars. This requirement of categorizing forces also addresses a general conceptual difficulty in statics: being clear which body exerts a given drawn force. It can also be seen that the requirement of selecting pins, as well as bars and partial bars, when forming a subsystem serves a similar purpose. The applied and support forces act on the pins, which have been explicitly selected to be part of the subsystem. Pins in the FBDs in the tutor also serve as convenient anchors for drawing these forces, again avoiding a jumble of forces that is difficult to interpret.

**Once a free body diagram is completed and is correct, the user types in equations of equilibrium.** Beneath the free body diagram the user can write equilibrium equations for the subsystem (Figure 4). Clicking on $\Sigma F_x = 0$, for example, initiates a place for such an equation; the user then enters the equation by typing it. The user can choose to write moments about any pin (as is typically done in truss analysis). Note that the interface naturally leads the user to associate any equations with a specific subsystem. Admittedly, students in statics do sometimes write down equations of equilibrium without specifying the subsystem or drawing its free body diagram. This design feature of the tutor reflects a trade-off between granting the user latitude to solve freely vs constraining the user. The task of interpreting a bundle of equations, each unclearly associated with a free body diagram, seemed likely to result in false errors. Note also that by clicking on $\Sigma F_x = 0$, for example, the user signals to the tutor that the equation should be judged by comparison with the correct summation of forces in the x-direction for that subsystem, in terms of the variables and constants as they appear in its free body diagram.

**The user can solve equations of equilibrium using a built-in calculating facility.** If the user has written down an equation with one variable (always a linear equation in truss analysis), upon request the tutor can solve the equation for that variable. This eliminates the need to use a calculator. The user can substitute such a solved variable into another equation that has more than one variable. But the tutor does not permit the simultaneous solution of multiple equations for multiple variables. This restriction on the solving capability promotes the practice of seeking to find an equation with a single variable, which can be determined and then used in subsequent equations. Such a practice of planning and organizing one's work is often wise when solving statics problems generally. Admittedly, some instructors might think that packages for solving simultaneous equations are common and even available on electronic calculators, and thus students should not be forced to choose equations in this way.

**Once a support reaction or internal force is determined, the user can declare it as solved and place it in a diagram where the solution is displayed.** Once a variable such as a support reaction or a bar internal force has been determined, the user needs to "register" that force in the solution diagram. Registration serves to declare that a force has been determined, and so it can be categorized as a determined force in a subsequent FBD. Registration is also an important opportunity for the student to signal the meaning of what has been solved. Unknown support forces can be drawn on FBD's in any direction; the associated variables may turn out to be positive or negative. But in the solution diagram the support force must be drawn in its actual sense and given a positive magnitude. Likewise, when the internal force of a bar is registered, the user gives it a magnitude and describes it as tensile or compressive.

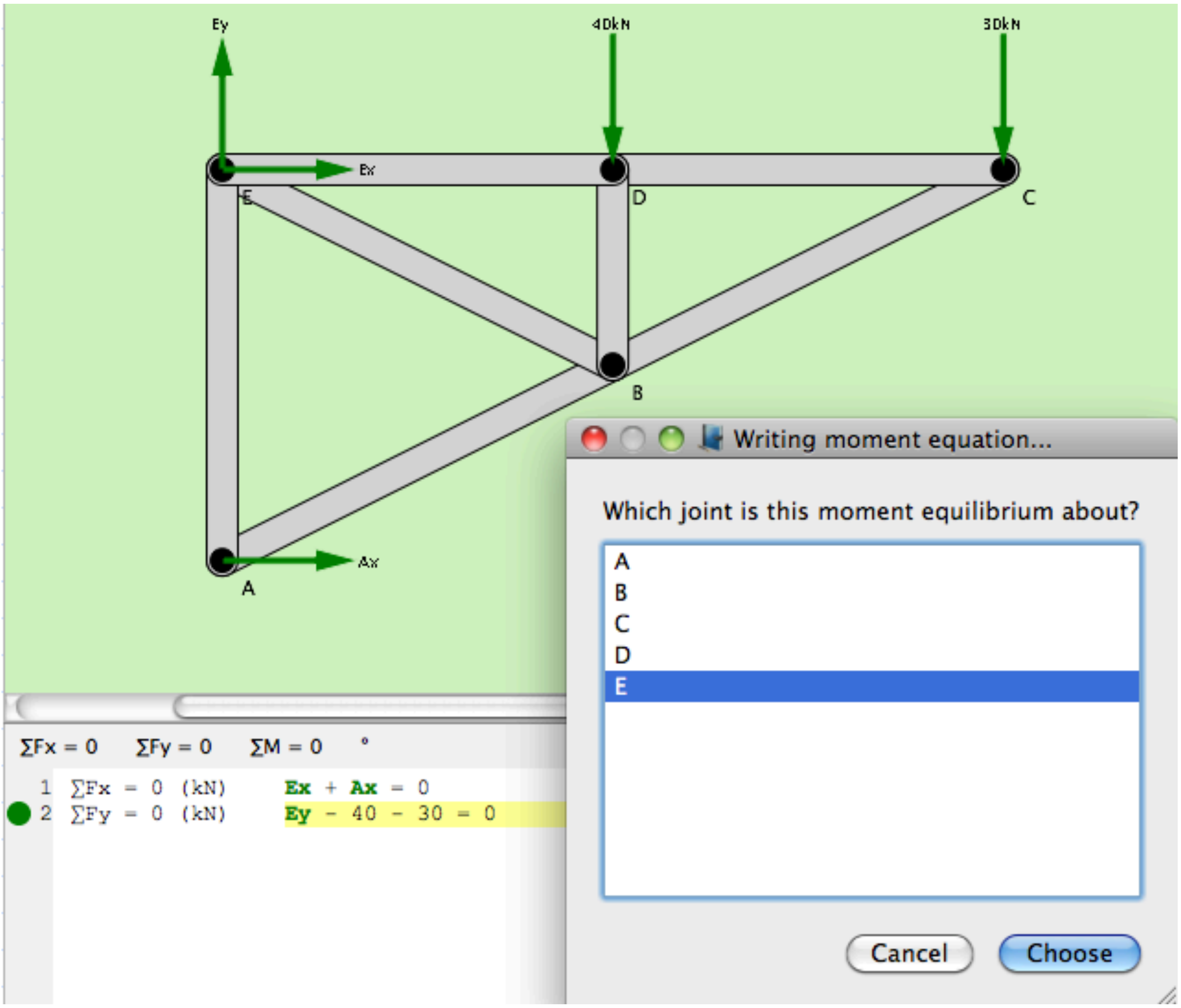


Figure 4. Screen shot of writing equations, and choosing moment center.

It is expected that a typical student would use the tutor for several hours over a period of a week or at most a few weeks, depending on the course. Therefore, the tutor must be easy to learn to use. Utilizing several rounds of user testing, we refined the user interface to be as an intuitive and simple as possible. However, some instruction in its use will inevitably be necessary. Therefore, when a student first starts the program, the tutor appears with an example problem loaded, and on top of the tutor window there appears a window with a voice-over instruction video that addresses how to solve the example problem. The instruction video contains four phases, which deal with successive features of using the tutor. The video pauses after each phase, and prompts the user to go to the tutor window and carry out the portion of the solution just described in the video.

There are major differences between Truss Tutor and existing statics problem solving systems[8,9]. One significant difference is that the Truss Tutor allows the user free reign to make mistakes, but at appropriate points signals the errors and gives feedback that enables the student to correct the errors. Thus, students have the experience of completing the problems correctly themselves. At the same time, Truss Tutor tracks whether a student makes error each time a step, such as drawing a force on a free body diagram or writing a term in a forces summation, is first undertaken. The collected data of successive attempts at different skills within solving truss problems enables another type of analysis[26] of student learning progress using the tutor.

**4. Judging student work and giving feedback**

A key component of the tutor is to judge and give feedback on what students have done thus far. When to offer feedback is a critical part of the tutor's design. On the one hand, we don't want to interrupt a student who is still formulating the current portion of the solution. On the other hand, we don't want to wait so long that the student builds new portions of the solution on others that are as yet unjudged and may be incorrect. In the latter, undesirable situation, the tutor would need to signal that the built-on portion is correct in and of itself, but that it must be redone to arrive at the correct answer because it was based on incorrect prior material.

We met this challenge in the tutor by judging student work just after the completion of each of the major phases of the solution; namely, after: selecting a subsystem, drawing a FBD of that subsystem, writing an equation of equilibrium for that FBD, and solving and registering results of an equation. Each of those tasks has a natural breakpoint at which it can be viewed as completed: upon selection of the parts for a subsystem, it is judged; upon choosing the first equation to be written (e.g., $\Sigma F_x$), the FBD of the subsystem is judged; upon typing return at the end of writing an equation, or choosing a next equation, the equation is judged; and, upon registering a result in the solution diagram, the registered result is judged. (Because the tutor solves one equation for one unknown upon request and the user must use this facility to obtain numerical solutions, all errors associated with algebraic manipulation and solving the equations are avoided.)

Provided the user does not make an error, the judging is invisible and the user can work without interruption. Upon making an error, the user receives an unmistakable error message. The message points out what is in error, with additional information to enable the user to fix the error and to learn why it is in error so it might not be repeated. The user can correct the indicated portion and proceed with the solution; the judging occurs at the same junctures so if the error is not fixed properly the error message will be sent again. **Thus, in the process of analyzing the truss, the user receives feedback on errors at selected instants; the user is then able to correct those errors and continue on with the solution of the problem.**

Note that students repeatedly execute a finite set of distinct judgeable actions, such as choosing a section, drawing an internal force on a partial bar, or registering a determined support reaction. More details on the breakdown of actions and errors are contained in a fuller description of the tutor[24]. Students will tend to err to varying degrees, depending on the type of action, but if the tutor is successful, the frequency of errors decreases with practice. The change in errors over the course of practice is discussed in the next section.

**5. Results for an initial cohort of students**

Here we report results from students taking a 3 credit-hour statics course at a community college, in a class comprising a total of 21 students. Students had received a lecture on trusses, covering the method of joints and method of sections, and the instructor worked through an example of each for the whole class. Thereafter, students practiced solving trusses exclusively using Truss tutor (no paper and pencil problems). Students were assigned five problems using the method of joints and five problems using the method of sections; nearly all students completed the method of joints first and then the method of sections problems. The following results are based on the 15 students who completed the full set of problems and consented to have their data studied.

Much can be gleaned from analyzing the log files and interpreting the progression of student errors. Here we show results for one type of analysis: the fraction of steps in error in each successive problem. Figure 5 shows boxplots that depict the distribution across students in the fraction of steps in error for successive method of joints problems. The box in a boxplot displays the middle two quartiles (from 25% to 75% of the students), with the median signaled by the intermediate line in the box. The whiskers at the top and bottom extend to the 10% and 90% percentiles.

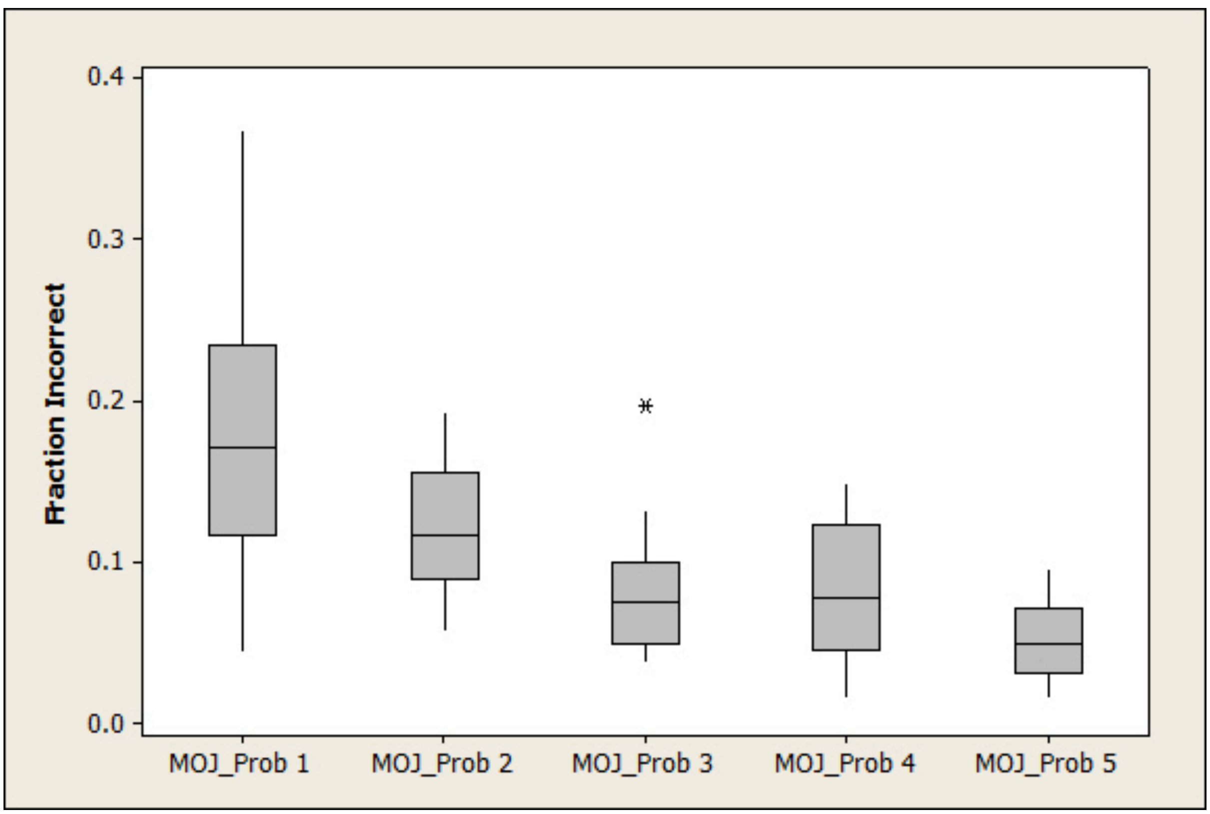


Figure 5. Boxplot depicting the distribution in fraction of steps incorrect across students for successive method of joints problems. The symbol * represents a student above or below the 90th or 10th percentile.

It can be seen that the median trends downward over successively problems (the mean, not displayed, exhibits a similar variation). Furthermore, using ANOVA it was determined that the means for the 5 problems were significantly different statistically ($F = 12.26$, $p < 0.001$). In particular, using a t-test to compare the means of two problems, we found Problem 2 had a significantly lower error rate than Problem 1 ($t = 2.84$, $p = 0.013$), and Problem 3 had a significantly lower error rate than Problem 2 ($t = 4.94$, $p < 0.001$). Hence, students using the tutor on average are learning: their error rate decreases from one problem to the next. No statistically significant changes were observed after the third method of joints problem.

Boxplots depicting the distribution of errors for the series of method of sections problems are shown in Figure 6. In this case there is too much variation to discern statistically significant reduction. Note that the median error rate is already relatively low, at approximately 0.1 or less. Students have already learned most of the skills needed to do method of sections problems by doing method of joints problems. The one skill that is peculiar to method of sections problems is drawing the section itself, that is, selecting an appropriate combination of pins, bars, and partial bars. A different analysis of errors, not for successive problems but successive attempts to use the same skill, has been carried out and is to be reported elsewhere. Indeed, the error rate for drawing the section itself decreases with practice. Since that is but one step of many in solving method of sections problems, depicting the average error rate for a whole problem masks this improvement. While a definitive study has not yet been undertaken, it appears that students who use the tutor are later able to solve truss problems with pencil and paper with the same facility as students who had done pencil and paper homework. This will be further explored.

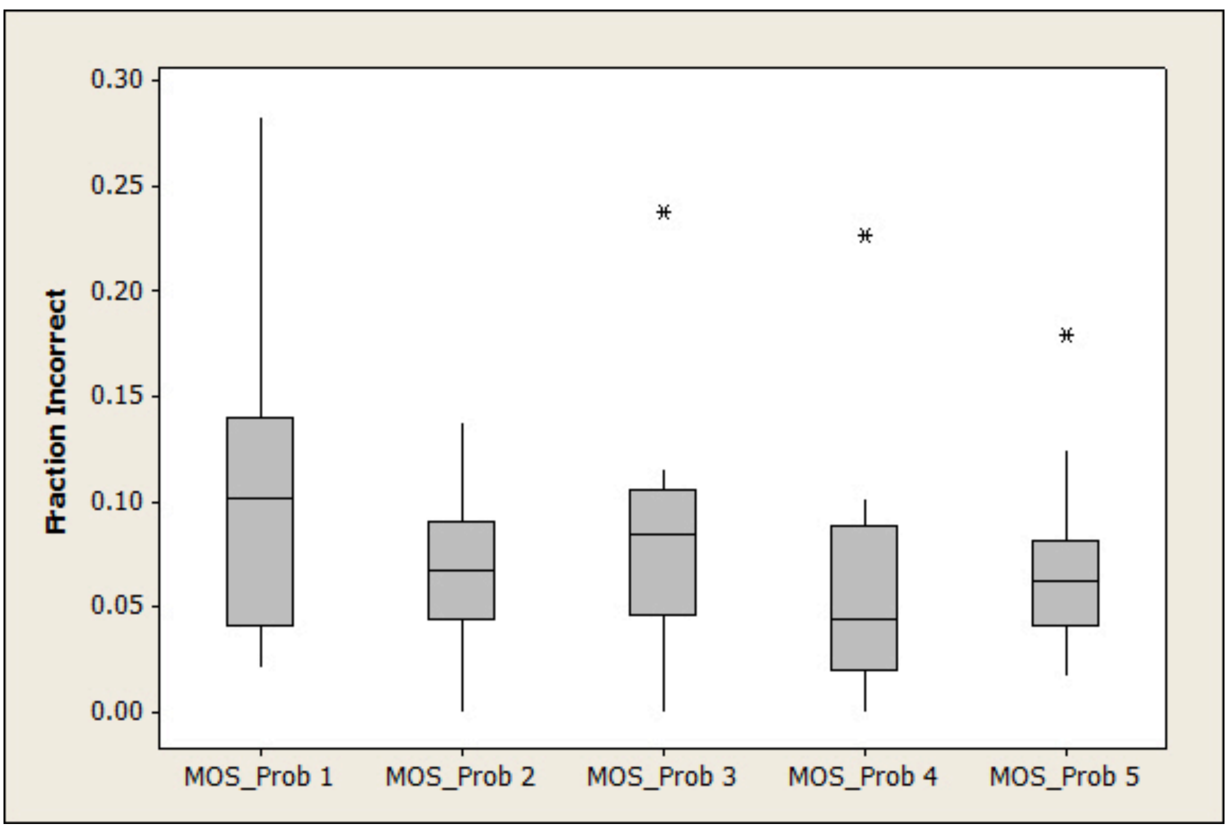


Figure 6. Boxplot depicting the distribution in fraction of steps incorrect across students for successive method of section problems. The symbol * is a data point above or below the $90^{th}$ or $10^{th}$ percentile.

## 6. Summary and conclusions

Statics is an important course for many engineering majors, and learning to solve problems is a key goal of statics. In more realistic topics in statics, such as structures, problems are often complex, involving multiple, but inter-related analyses; there are many pathways to correct results. Feedback is important to learning, particularly when students are engaged in problem solving. To the extent possible, one wishes to capitalize on the potential of the computer to provide such feedback to students when instructors or human tutors are not present. But, such feedback can be challenging for the computer to offer in the case of complex problems, when students are allowed to pursue different solution pathways. The computer needs a cognitive model that accounts for the different steps a user could take, which would enable it to follow the user's steps and compare them to correct steps pertinent to a chosen solution pathway. An additional challenge is for the computer to recognize the complex graphical and textual input in mechanics.

In response to this challenge, we have sought to develop a computer tutor that can offer students feedback as they practice analyzing trusses. The tutor should efficiently increase the student's ability to analyze trusses, with the improvements applicable to solving truss problems with pencil and paper. Such a tutor, if successful, may point the way to new computer tutors appropriate to student solving of complex multi-path problems in other subjects.

A first version of a computer tutor for trusses has been described. The design of the tutor has sought to strike a balance between (1) allowing students wide latitude to solve truss problems correctly and to commit errors typical of novices and (2) constraining student actions so as to be unambiguously interpretable by the tutor. In a few instances, users of the tutor must take actions or make choices that do not have precise counterparts to pencil and paper solving. These serve either to help students organize their thinking and/or to make interpretation by the tutor more straightforward. The tutor offers feedback to users at convenient points, seeking a balance between not interrupting work in progress, but not allowing incorrect work to accumulate and be the basis for subsequent work. The feedback is sufficient for nearly all students to complete work correctly on all problems.

The different steps needed to solve truss problems are tracked, and we can determine the rates at which errors are committed. The effectiveness of the tutor is gauged by whether the rate of errors decreases as students practice. In this paper we present data from students in a statics class who used the tutor in lieu of solving truss problems by paper and pencil. Students solved five problems using method of joints and five problems using method of sections. It was found that the fraction of steps with errors decreased from the first to the second to the third method of joints problem and that the error rate remained low thereafter. The error rates in method of sections problems, undertaken after the method of joints problems, were consistently rather low. Alternative methods of tracking the change in errors with practice have been undertaken. These methods, which also reveal a pattern of improvement with practice, is reported elsewhere[25]. Thus far, our approach has been applied only to truss problems. However, it can be adapted to solve other complex problems in statics, and potentially other subjects, and these opportunities are being explored.

In conclusion, tutoring courseware can be devised that enables students to solve relatively complex problems with some freedom, while at the same time understanding student efforts well enough to provide them with feedback. The effectiveness of the tutor in promoting student learning through this feedback can then be gauged, and ideally improved, by tracking the reduction in the rate at which errors are committed.

**Acknowledgements**

We thank Jackie Yang, Jeremy Jiang, and Rebecca Piston for their assistance in development and initial testing of Truss Tutor. Support by the National Science Foundation under grant DUE-1043241 and by the Department of Mechanical Engineering at Carnegie Mellon University is gratefully acknowledged.

**Bibliography**

1. Harris, T.A. and Jacobs, H.R., 1995, J. Eng. Educ., Vol. 84, p.343.
2. Bangert-Drowns, R. L., Kulik, C.-L., Kulik, J. A., & Morgan, M. (1991). The instructional effect of feedback in test-like events. Review of Educational Research, 61, 213-238.
3. Corbett, A.T. and Anderson, J.R. (2001). Locus of feedback control in computer-based tutoring: Impact on learning rate, achievement and attitudes. Proceedings of ACM CHI'2001 Conference on Human Factors in Computing Systems, 245-252.
4. Lewis, M. W., & Anderson, J. R. (1985). Discrimination of operator schemata in problem solving: Learning from examples. Cognitive Psychology, 17, 26-65.
5. Lee, A. Y. (1992). Using tutoring systems to study learning: An application of HyperCard. Behavior Research Methods, Instruments, & Computers, 24(2), 205-212.
6. Schmidt, R. A., & Bjork, R. A. (1992). New conceptualizations of practice: Common principles in three paradigms suggest new concepts for training. Psychological Science, 3(4), 207-217.
7. Schooler, L. J., & Anderson, J. R. (1990). The disruptive potential of immediate feedback. In M. Piatttelli-Palmarini (Ed.), Proceedings of the Annual Conference of the Cognitive Science Society (pp. 702-708). Hillsdale, NJ: Erlbaum.
8. Roselli, RJ, Howard, L and Brophy, S. (2006) A computer-based free body diagram assistant. Computer Applications in Engineering Education 14: 281-290.
9. Dannenhoffer, J. & Dannenhoffer, J. (2009). An online system to help students successfully solve statics problems. Proceedings of the 2009 American Society for Engineering Education Annual Conference & Exposition, Austin, Texas, June 2009.
10. Peschel, J., and Hammond, T. STRAT: A Sketched-Truss Recognition and Analysis Tool. International Workshop on Visual Languages and Computing (VLC 2008), Boston, Massachusetts, September 4-6, 2008, pp. 282--287.
11. Newton's Pen - A Pen-based Tutoring System for Statics WeeSan Lee, Ruwanee de Silva, Eric J. Peterson, Robert C. Calfee, and Thomas F. Stahovich, Computers & Graphics, 32(5):511-524, October 2008.
12. Sleeman, D. H., & Brown, J. S. (1982). Intelligent Tutoring Systems. New York, NY: Academic Press.
13. Wenger, E. (1987). Artificial Intelligence and Tutoring Systems: Computational and Cognitive Approaches to the Communication of Knowledge. Los Altos, CA: Morgan Kaufmann.

14. David J. Gunaratnam, A knowledge-based expert system for tutoring in structural engineering, Computers & Structures, Volume 30, Issue 3, 1988, Pages 767-773

15. Ana Lilia Laureano-Cruces and Fernando De Arriaga-Gomez, Multi-Agent Architecture for Intelligent Tutoring Systems, Interactive Learning Environments, Volume 6, Issue 3 December 1998 , 225 - 250

16. Ana Lilia Laureano Cruces and Fernando De Arriaga, REACTIVE AGENT DESIGN FOR INTELLIGENT TUTORING SYSTEMS, Cybernetics and Systems, Volume 31, Issue 1 January 2000, 1 – 47

17. Kulik, C. C. & Kulik, J. A. (1991). Effectiveness of Computer-Based Instruction: An Updated Analysis. Computers in Human Behavior, 7, 75-95.

18. Anderson, J. R., Boyle, C. F., & Reiser, B. J. (1985). Intelligent tutoring systems. Science, 228, 456-468.

19. Newell, A., & Simon, H. A. (1972). Human Problem Solving. Englewood Cliffs, NJ: Prentice-Hall.

20. P.S. Steif, 2004, An Articulation of the Concepts and Skills Which Underlie Engineering Statics, 34rd ASEE/IEEE Frontiers in Education Conference, Savannah, GA.

21. P.S. Steif and J. A. Dantzler, "A Statics Concept Inventory: Development And Psychometric Analysis", Journal of Engineering Education, Vol. 94, pp. 363-371 (2005).

22. Steif, P.S. and Hansen, M.A, "Comparisons Between Performances In A Statics Concept Inventory and Course Examinations," International Journal of Engineering Education, Vol. 22, pp.1070-1076 (2006)

23. P.S. Steif and M. A. Hansen, "New Practices for Administering and Analyzing the Results of Concept Inventories", Journal of Engineering Education, Vol. 96, pp. 205-212, (2007).

24. P.S. Steif, L. Fu, and L.B. Kara, "Technical Report: Development of a Cognitive Tutor for Learning Truss Analysis, Retrieved 02/10/14 from http://www.andrew.cmu.edu/user/steif/papers/Truss%20tutor%20Technical%20Paper%202013.pdf

25. P.S. Steif, L. Fu, and L.B. Kara, "The Potential for Computer Tutors to Assist Students Learning to Solve Complex Problems", Proceedings of the 2014 American Society for Engineering Education Annual Conference & Exposition, Indianapolis, IN, June 2014.