121st ASEE Annual Conference & Exposition
Indianapolis, IN
June 15-18, 2014

Paper ID #8510

The Potential for Computer Tutors to Assist Students Learning to Solve Complex Problems

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.

Dr. Luoting Fu

Luoting Fu has a BS in aerospace engineering from Shanghai Jiao Tong University, and an MS and PhD in mechanical engineering from Carnegie Mellon University. This work was performed when he was with the Visual Design and Engineering Lab in the Department of Mechanical Engineering at Carnegie Mellon University.

Levent Burak Kara, Carnegie Mellon University

The Potential for Computer Tutors to Assist Students Learning to Solve Complex Problems

1. Introduction

Engineering education includes significant attention to problem solving skills, with students gradually confronting problems of increasing complexity. Even within any single fundamental engineering science course, which addresses a limited set of concepts, students can still face problems that require coordinating and organizing multiple parts. To solve, the student may need to decompose the problem into inter-related sub-problems, define variables of different types, carry out analyses of sub-problems, and finally combine and interpret the results. Such problems may have multiple pathways to the correct answers.

There is wide recognition that learning to solve problems, in general, is promoted by timely, formative feedback[1-6]. This paper addresses the issue of providing formative feedback for students confronting complex problems that involve significant latitude in decomposition and construction of solutions. Traditionally, students solve complex problems as part of written homework assignments that are hand graded. In such circumstances, offering effective formative assessment is exceptionally challenging, requiring careful attention to solution details and rapid, rather than weeklong, turnaround. Furthermore, within a single problem, later work can build upon earlier work; hence, grading of an already completed solution often involves judging off-path steps that may be irrelevant to the intended learning or steps that build upon prior incorrect work. Given the very limited effectiveness of human grading to provide feedback to students on complex homework problems, it is natural to inquire whether the computer can do better.

The research questions this paper seeks to answer are: (1) Is it possible to provide automated, formative assessment of efforts to solve complex engineering problems, (2) What metrics allow one to judge whether the feedback indeed promotes learning, and (3) On what basis can one seek improvements to the formative assessment offered?

We address these questions in the context of a test case: a tutor for students learning to solve truss problems, which are commonly studied in statics. Trusses exemplify complex problems: students select multiple portions of the truss, draw free body diagrams, write down appropriate equilibrium equations for each diagram, organize the solving of equations, and interpret results physically in terms of the original truss. Mastery requires clarity, systematic organization, as well as conceptual and mathematical competence. Recently, computer systems have been developed that allow students to work on some simple statics problem more or less from start to finish, and provide feedback on individual steps[7,8]. But, such systems do not involve problems with many solution paths, nor do they offer data upon which to judge how much students are learning.

To give feedback to a user who can pursue various pathways in problem solving, a computer tutor must have a model of the problem solver's thought process. Indeed, researchers have developed Intelligent Tutoring Systems[9,10], including even some relevant to the mechanics of structures[11-13]. Cognitive tutors[14] in particular merged the ideas of intelligent tutoring systems with computational models of cognitive theories of human learning, memory, and problem solving[15]. Cognitive tutors are based on a cognitive model of a student interacting with problems in a domain.

One approach to a cognitive tutoring system for solving problems such as trusses would (i) allow students to take correct as well as commonly incorrect pathways in the solution and (ii) recognize when a student has gone off a correct path and give guidance on how to correct errors. In this paper, we determine whether automated formative assessment is possible for truss problems following just such an approach, and we establish metrics for judging the effectiveness of the feedback offered. Building on observations of typical errors[16-19], a computer interface was created where correct steps and typical errors in solving truss problems can be executed with wide latitude to pursue solution paths. Algorithms for correct forward steps were developed that are applicable to any correct solution state. To grant latitude to the solver, while retaining ability to interpret work, the tutor must intervene in a timely way after errors are committed. The student can solve unimpeded until errors are made that can interfere with future solving steps; feedback is then offered which enables students to correct their errors. To judge whether the feedback is effective, steps hypothesized to involve the same components of knowledge have been grouped, and data is collected on the fly of successive attempts to apply the different knowledge components. Statistical models are used to determine whether errors in using different knowledge components decrease in frequency with practice. The determined learning rates give insights into whether feedback is effective and can inform future improvements in the tutor. The approach is tested out on data from students from two institutions.

2. Design of tutor

Here we offer a brief summary of the design of the tutor used for the present study; more details and rationalization for design decisions are presented elsewhere[20]. A tutor for problem solving of trusses was designed based on the tasks required and on observations of typical student errors. Solving truss problems involves several groups of tasks: selecting a subsystem, that is, a some portion of the truss, for analysis; drawing free body diagrams of selected subsystems; writing down equations of equilibrium for the free body diagrams; and, solving equations, interpreting results, and potentially using them to analyze subsequent subsystems. These groups of tasks share features with other problems in statics.

Students commit a variety of errors in solving truss problems. One typical error is illustrated in Figure 1: a student has written equilibrium equations 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. A second error is shown in Figure 2: internal forces drawn on entire bars included in the subsystem; they should instead be drawn only on partial bars. Further examples of errors in truss problem are presented elsewhere[20].

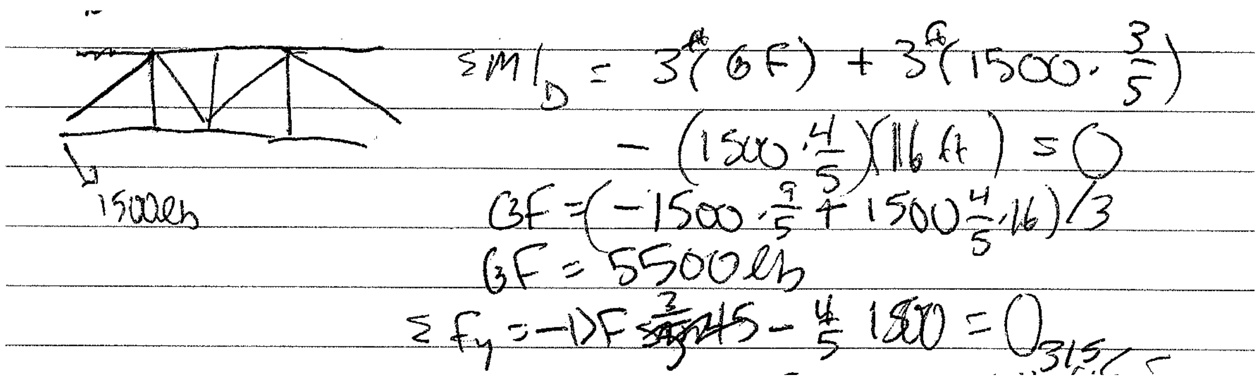

Figure 1. Internal forces (GF and DF) are not drawn on section, but appear in equilibrium equations; the solutions ultimately have sign errors.

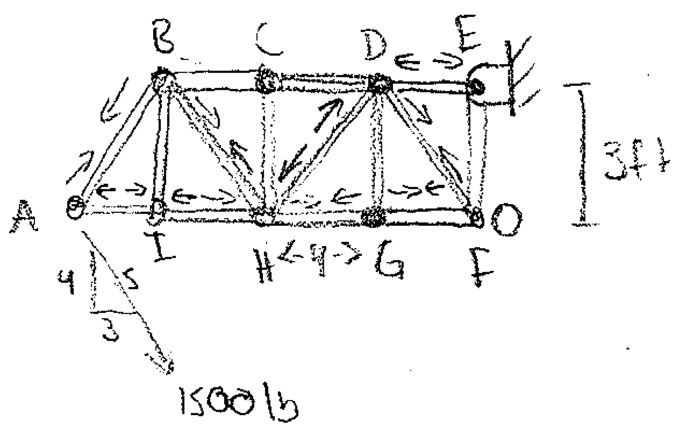

Figure 2. Internal forces drawn on bars that are fully included in the subsystem.

The tasks needed to be performed and the observations of student errors suggested that a tutor constraining user choice as follows would 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, 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.
- Equations of force equilibrium along x-y, and equations of moment equilibrium about any joint, can be written.

Finally, we have assumed that students using the tutor have learned about truss analysis through other means, such as lecture and textbook; thus, the tutor focuses exclusively on helping students solve problems. The computer tutor should have a simple, easily learnable user interface that gives students reasonably wide latitude to solve truss problems with minimal distractions and unnecessary effort, but still make errors commonly found.

Figure 3 contains a screen shot of the tutor, with a problem partially solved. 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 expands that subsystem, allowing the user to draw its FBD and write its associated equilibrium equations.

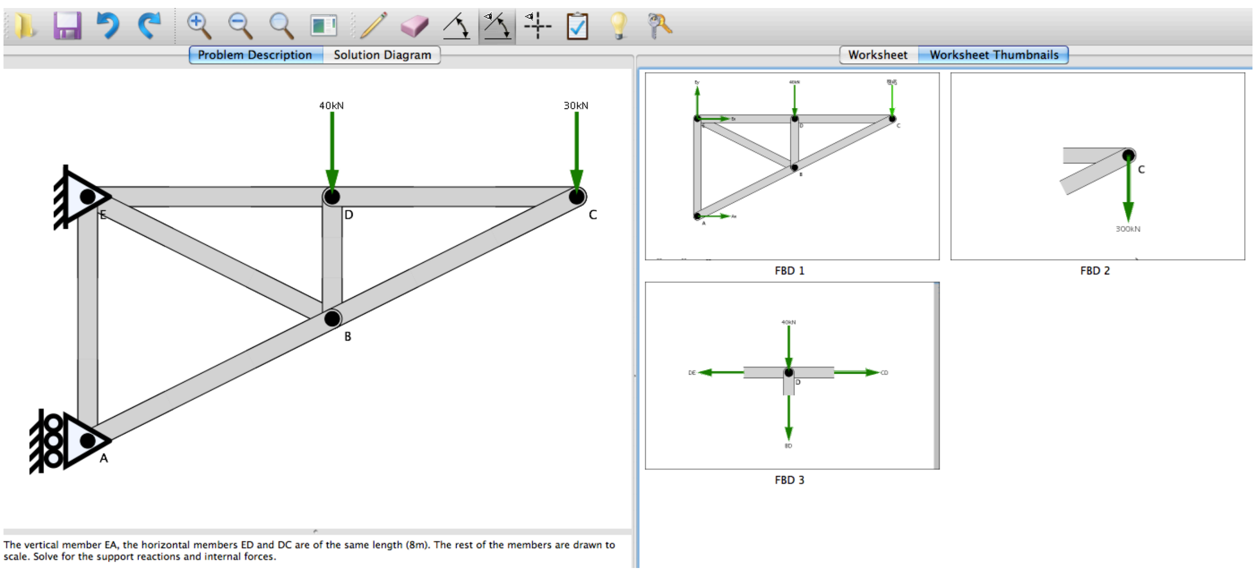

Figure 3. Screen shot of full display of truss tutor.

In Figure 4, we show a subsystem with a pin and the two connected partial members; a new force being added to a partial member. As seen in the window labeled “Defining a force”, the user categorizes each force being drawn.

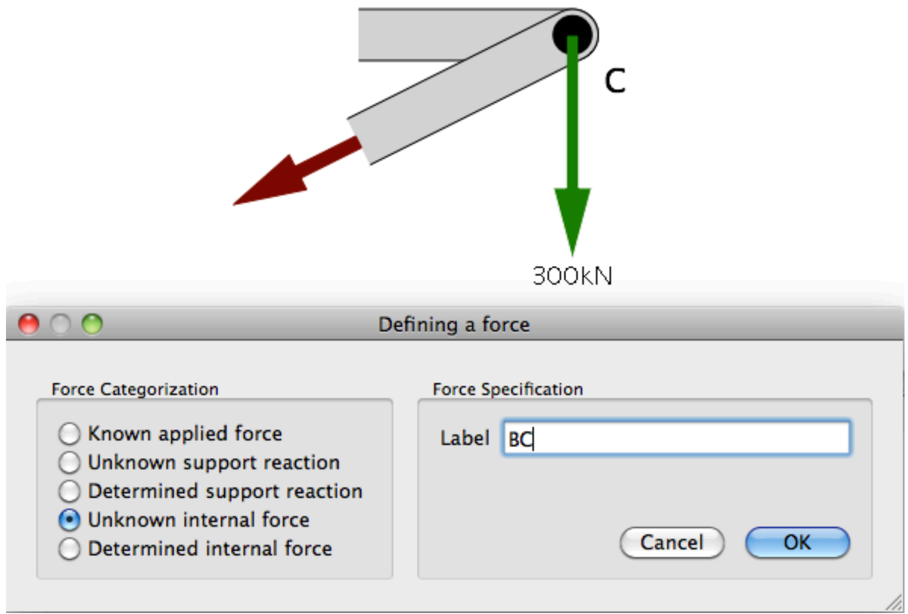

Figure 4. Screen shot of force being added to free body diagram, showing force categorization.

Beneath the free body diagram the user can write equilibrium equations for the subsystem (Figure 5). When 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 and also eliminates errors due to mistyping into a calculator. Once a variable such as a support reaction or an 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, 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 in tension or compression.

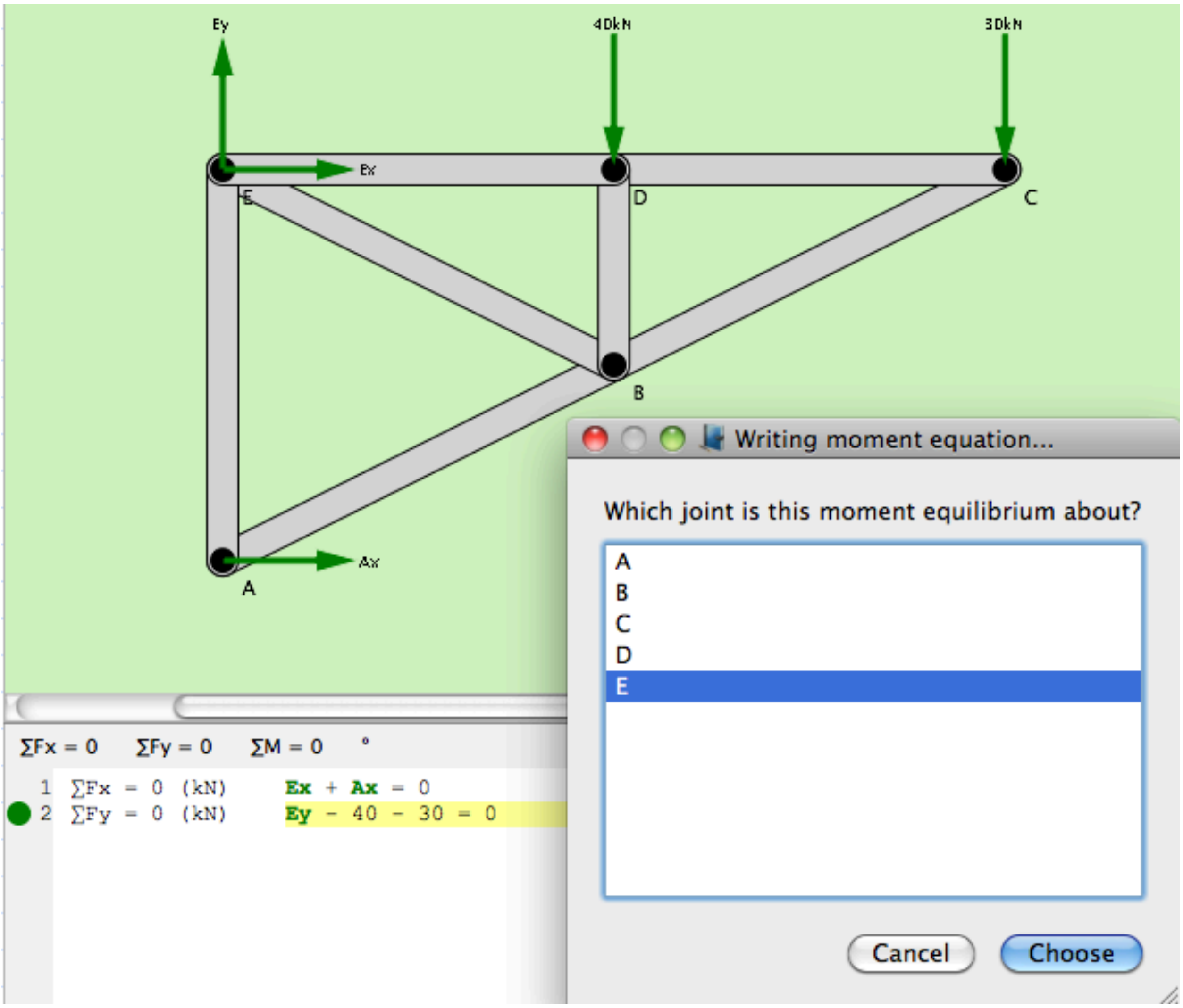

Figure 5. Screen shot of writing equations, and choosing moment center.

3. Judging student work and giving feedback

A key capability of the tutor is to judge and give feedback on work. The tutor can do this by having a cognitive model for solving truss problems. The cognition in the tutor consists of several elements:

- SUBSYSTEM: An algorithm for what configurations of pins, members, and partial members is a legitimate subsystem.
- FBD: Given a legitimate subsystem, and any forces defined or determined up to that point, an algorithm for the allowable forces that can be drawn on the pins and partial bars of the subsystem. The FBD of a given subsystem is not unique. For example, if an internal force has been determined, then one can represent that force in the FBD either as a determined force using the correct value, or as an unknown internal force using symbols consistent with the first definition.

- EQUILIBRIUM EQUATIONS: Given a legitimate FBD, an algorithm for the correct summation of forces along x and y and the correct moment about any pin in the truss. These summations include variables and constants, consistent with how the forces appear in the FBD.
- SOLUTION REGISTRATION: Given a correctly determined support or internal force (from the equilibrium equations), an algorithm for the correct registration of that force in the solution diagram.

While the student can pursue many different solution paths that can be followed by the tutor, it does not have algorithms for solution paths that build on earlier committed errors. Thus, when to offer feedback on an error 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 upon work that is as yet unjudged and may be incorrect. In the latter, undesirable situation, the tutor might need to indicate that the built-on portion is correct in and of itself, but that it is based on incorrect solution steps.

We met this challenge in the tutor by judging student work just after the completion of each of the major phases of the solution for each subsystem. Each task has a natural breakpoint at which it can be viewed as completed and thus ready to be judged: upon selection of the parts for a subsystem, it is judged; upon choosing the first equation to be written (e.g., ΣF_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.

Provided the user does not make an error, the judging is invisible (but recorded) 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, lessening the likelihood of repetition. The user can alter the indicated part of the solution and proceed; the judging occurs at the same junctures so if the error is not fixed a new error message will be sent.

4. Analysis of student work on tutor to track learning

Need for decomposition into skills or Knowledge Components

To investigate whether students learn while using the tutor to solve problems, we seek to determine if they are making fewer errors as they progressively solve more problems. Being able to solve something as complex as a truss problem involves a variety of subtasks. Further, some subtasks are more prone to error than others, and students may improve more quickly on some subtasks than others, potentially depending on the feedback. How we choose to view the problem as composed of subtasks is central to developing evidence as to whether learning is occurring. These choices constitute our model of learning to solve problems in the chosen subject or topic: they are the distinct chunks or Knowledge Components that the student needs to learn.

To see that recognizing distinct subtasks is necessary, consider the grading of homework problems that involve many steps or facets. It would clearly be ineffective to grade an entire problem as simply correct or incorrect. First, this would not capture the great variation within "incorrect" problems. Second, if we imagine that students received feedback on each problem and so had the opportunity to learn from each problem, we would likely not observe a steady improvement in their ability if the entire problem were merely deemed correct or correct. Instead, we would like to signal which subtasks are correct or not. Then, if we had the resources to track how student performed on different types of subtasks, and students received feedback on separate subtasks, then we might indeed observe improvement in their ability to execute the different subtasks.

Thus, our goal is to sensibly designate the different subtasks or distinct skills that must be mastered to ultimately solve such problems. Of course, we also want the clarity of being able to characterize each attempt to use a skill as unambiguously correct or correct. We hope the frequency at which those attempts are correct increases with practice. How then should we divide up the overall task? Critical to that division is its granularity – how small are the actions that are deemed to reflect individual skills and how many different skills are recognized among the different actions of the same general type.

To illustrate issues of granularity and variation, consider the task of drawing a free body diagram: in trusses we typically draw an FBD of the whole truss, of a joint, or of a section. One might deem the drawing of an FBD of an already chosen portion of the truss as a single skill. However, an FBD is typically composed of many forces. Say we treated an FBD as wrong if any force in it is wrong: since drawing different types of forces may incur more or fewer errors, treating any incorrect force as the same error may fail to capture inherent differences in the ease of learning to represent different types of forces. An alternative approach would be to view the drawing and labeling of different types of forces – applied forces, support reaction forces, and internal forces – as distinct skills. By contrast, all reaction forces associated with pin supports will be treated as reflecting the same skill in our current model for learning. There is, of course, no single correct model, and empirical evidence as described below can be gathered to compare different models.

Initial KC Model for Truss Tutor

The collection of Knowledge Components and their association with specific student actions is referred to as the KC Model. Many different KC models are possible; in this section we describe the initial KC model chosen for the tutor described here; more details have been presented elsewhere[20]. Each KC corresponds to an action that falls into one of the three phases of the solution process: (i) selecting a subsystem, (ii) drawing a free body diagram, and (iii) registering a result derived from an equation of equilibrium in the solution diagram.

Selecting a subsystem

Legitimate subsystems are either the entire truss, or a portion of the truss consisting of pins, connected members, and partial members. Distinct KCs correspond to defining these types of subsystems:

- whole truss as subsystem
- joint as subsystem
- section as subsystem

Drawing a FBD

The user can draw various types of forces on either the pins or on the ends of partial members. Distinct KCs correspond to the distinct situations in which a force may be defined:

- applied force
- free pin (should have no forces)

- new unknown pin support reactions (first definition at a given support)
- new unknown roller support reaction (first definition at a given support)
- already defined unknown support reaction (subsequent definition of support reaction must be consistent with first definition)
- determined support reaction (must be consistent with value and direction determined earlier)

- new unknown internal force (first definition for a given member)
- already defined unknown internal force (subsequent definition of internal force must be consistent with first definition)
- determined internal force (must be consistent with value and direction determined earlier)

Note that in defining the KCs as above, we are implicitly treating all instances of defining, say, a new pin support as the same regardless of, for example, where the pin is located, its label, and whether there are also applied forces acting on the pin. Likewise, all instances of designating a determined internal force are treated the same regardless of, for example, whether it was in tension or compression, where in the truss the member is located, and whether applied forces act on connected pins. One could have a KC model with more granularity by choosing to have distinct KCs corresponding to those different instances. It is empirical question as to whether such a finer grained model captures learning better. If, at a given instant in a student's learning process, the likelihood of making an error were to be significantly different for the different instances, then they ought to be tracked as distinct KCs.

Registering Results

Upon solving an equation of equilibrium, the user registers those results in the solution diagram. Distinct KCs correspond to:

- Registering a support reaction
- Registering an internal force

Students have experience in writing equations of equilibrium based on a completed FBD from earlier topics in statics; the tutor was not viewed as likely to produce meaningful improvement in writing equilibrium equations. Students do receive feedback on their equations of equilibrium, which enable them to correct those equations. However, other than pointing to the site of an equation error, the tutor does not provide explanatory feedback. Furthermore, it is likely that students are sloppy writing equations because they get feedback so quickly. Indeed, while different types of contributions to equations of equilibrium are tracked, errors of different types were not found to decrease systematically with practice. Thus, the KC Model presented here focuses on facets of solving truss problems other than writing equations of equilibrium.

Tracking Opportunities to Exercise Knowledge Components

The tutor records each new instance in which the user undertakes an action corresponding to one of the KCs. Each opportunity to exercise the KC is deemed either correct or incorrect. Any fixing of an incorrect step in response to feedback is not counted as a new opportunity to exercise the KC. Note that the tutor is quite distinct from most existing tutors in that there is no pre-defined set of questions or specific steps. The student charts his or her own solution pathway, which could involve analyzing, for example, an entire truss followed by joints in an order selected by the student. The tutor extracts on the fly the sequence of KCs attempted, which can be different for each student.

5. Analysis

To analyze the progression of learning quantitatively, we have adopted the terminology, methodologies, and tools from the PSLC (Pittsburgh Science of Learning Center) Datashop[21]. Data corresponding to the sequence of KC opportunities for each student are extracted from the files the student saves using the tutor; these data are imported into Datashop. Among the various outputs from Datashop pertinent to our study is the learning curve: the percentage of students that err in applying a particular KC is plotted as a function of the opportunity (first, second, third) to use that KC. From such data, which is typically noisy, one seeks to determine if there is some evidence of a pattern.

Hence, the Datashop tools also fit a statistical model to the sequence of opportunities to apply a KC. For our learning model in which each action is dependent on a single KC, the statistical model predicts error fraction as follows:

$$\ln[(1- e_{ij})/e_{ij}] = \theta_i + a_j + b_j\, T_j$$

In this equation, e_{ij} is the probability of an incorrect answer by the ith student on opportunity T_j for using the jth KC. Note that e_{ij} can range from 0 to 1, and T_j takes on values of 1, 2, 3, and so forth, for the first, second, and third opportunity. This logistic regression model[22] for measuring the progressive mastery of a skill with practice has been widely used and generalized.

Fitting this model to data for a group of students yields a student-specific, overall initial skill level θ_i, which is independent of the KC. The fit also produces coefficients a_j and b_j, both KC-dependent, but student-independent. The coefficient a_j, the intercept, corresponds to the initial probability of correctly applying the KC. The coefficient b_j, referred to as the slope, corresponds to the rate at which errors in using the jth KC decrease with successive opportunities to practice it. Thus, values for b_j are one measure of the tutor's effectiveness. In particular, more effective error messages or hints should lead to higher slopes.

6. Samples

The tutor described here is appropriate for students in virtually all statics courses. Because the tutor is intended to substitute for completing paper and pencil homework, use of the tutor fits into the rhythm of statics courses generally. Thus, the target population for a tutor such as this corresponds to most students who might take a statics course.

Because we wanted to capture how a tutor can give feedback on complex problem solving in the context of real engineering courses, the study was conducted within the scope of regularly scheduled statics courses. The tutor was used in lieu of solving paper and pencil homework problems in two distinct educational environments. Data was collected for all students and information on their completion of problems was returned to the instructor for the purposes of a assigning a grade on the homework assignment. When students first registered to receive the software, they were asked if they consented to have their data used for research; only data from those who consented were included in the analysis.

Sample 1 was from a statics course at a community college, in a class comprising a total of 21 students. Of those students, 18 consented to have their data studied. Sample 2 was from a statics course at a military academy, in a class comprising a total of 109 students. Of those students, 99 consented to have their data studied. Students had received lecture on trusses, covering the method of joints and method of sections, and were shown the solution of example problems. Thereafter, students practiced solving trusses exclusively using Truss tutor (no paper and pencil problems). Students in sample 1 were assigned five problems using the method of joints and five problems using the method of sections; sample 2 students were assigned three problems using the method of joints and five problems using the method of sections. There is no claim that these two samples are broadly representative of students; nor does one expect them to be exceptional.

7. Results

Typical learning curves are shown in Figures 6 - 8. The data points and solid lines connecting them (in red) are the actual error fractions. The dotted (blue) curve is the prediction based on the fit of the statistical model. These learning curves are from Sample 2; the early opportunities correspond to all 99 students, while a diminishing number of students might contribute to the error rate with successive opportunities (because different students have different solution paths and even solve fewer problems). These three learning curves represent three typical outcomes. For the KC depicted in Figure 6, registering an internal force, the error starts low and remains low. There is little need for tutoring on this skill. For the KC depicted in Figure 7, utilizing a determined support reaction in a subsequent FBD, the error is reasonably high initially and becomes progressively lower with practice. This suggests that practice is having a desired effect – getting feedback on the errors enables students to gradually make fewer errors. Finally, for the KC depicted in Figure 8, which pertains to one facet of writing equations of equilibrium, the error rate is initially high and never improves. (The wild error rate at the end corresponds to a very few students making many errors.) Practice is having no observable benefit. Results for knowledge components associated with writing equations of equilibrium are not presented here.

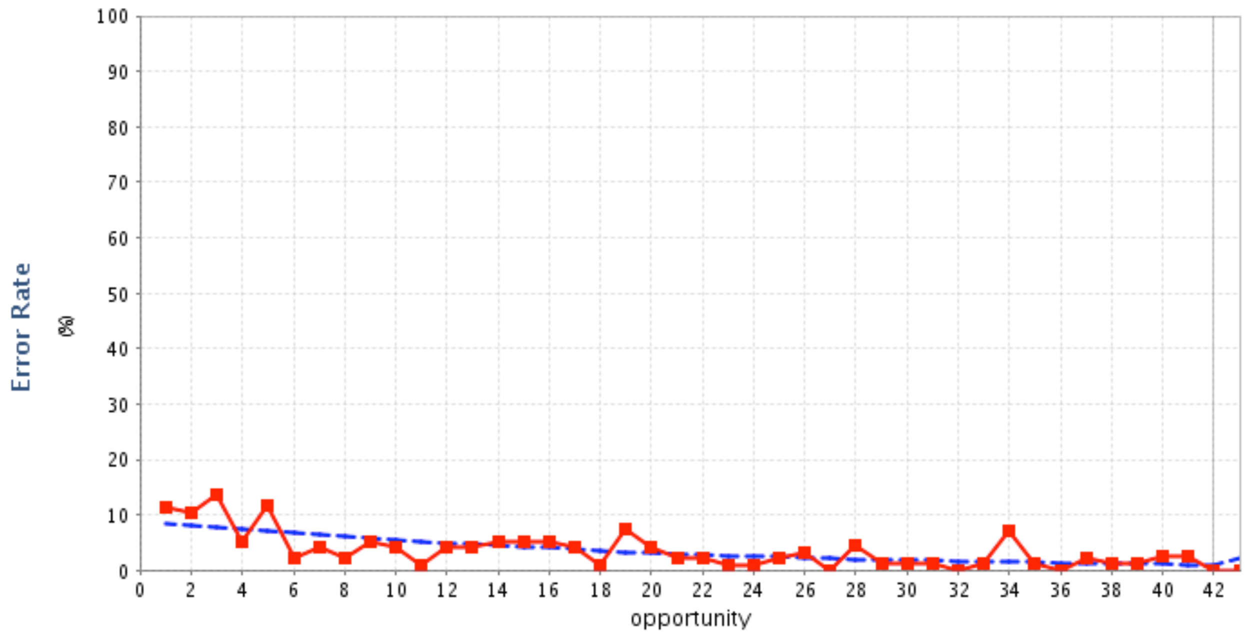

Figure 6. Percentage of students in error plotted as a function of opportunity (Learning curve) for a KC (registering an internal force) for which the error rate starts low and remains low.

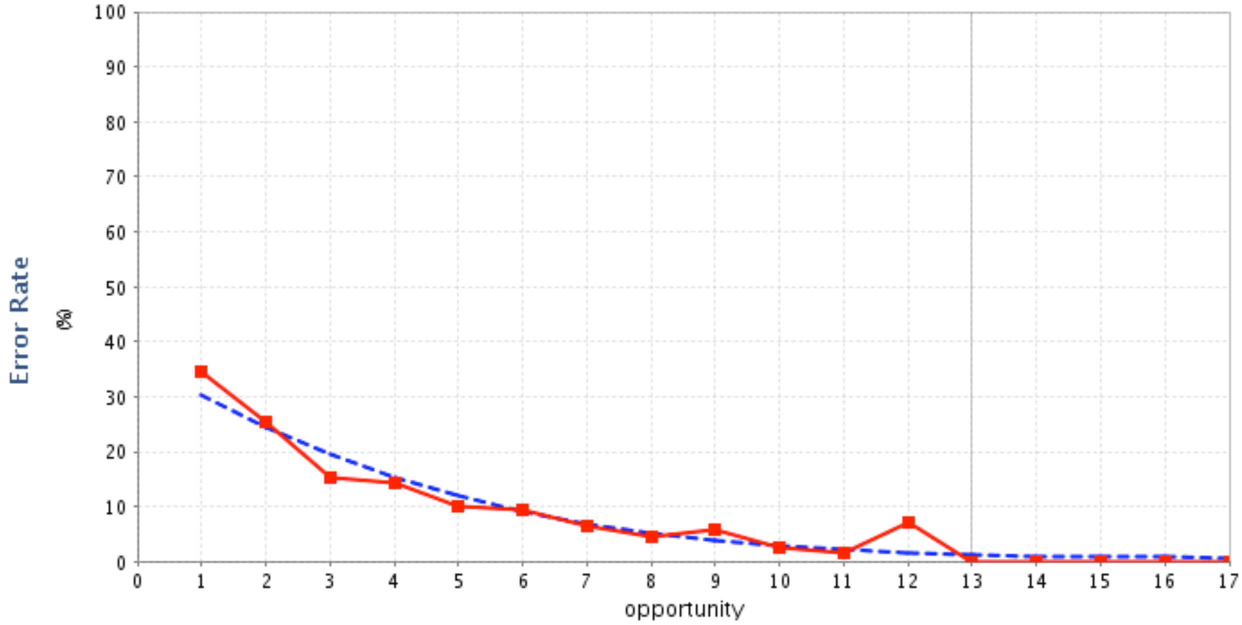

Figure 7. Percentage of students in error plotted as a function of opportunity (Learning curve) for a KC (representing a determined support reaction) for which error rate is initially high, but decreases with practice.

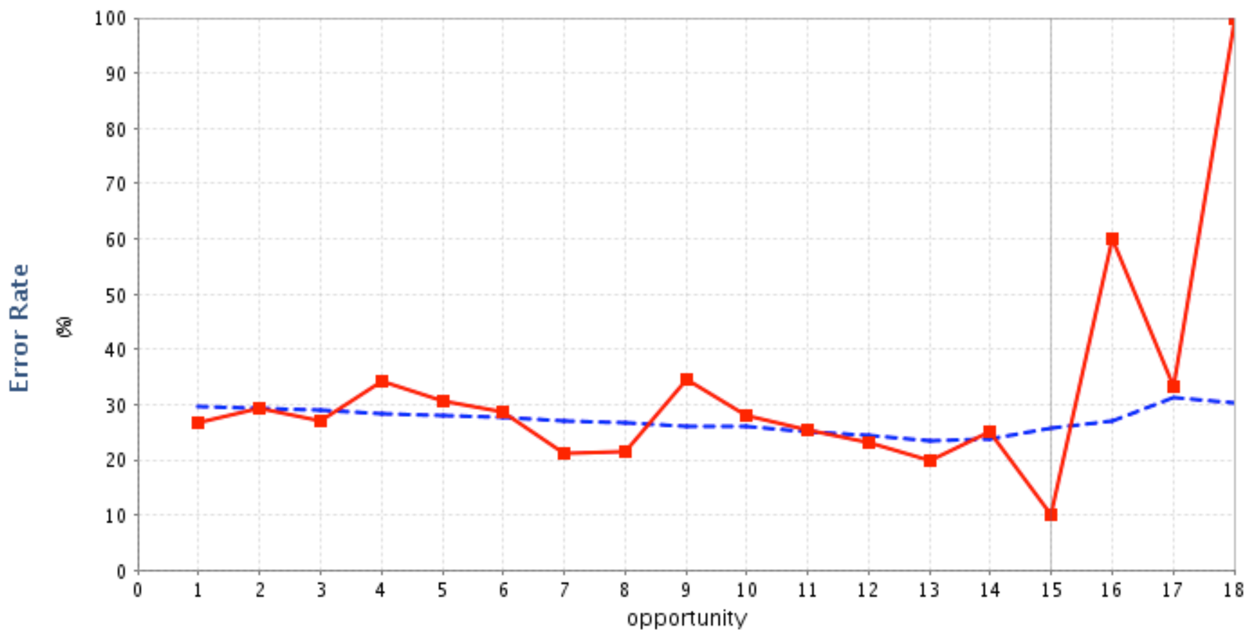

Figure 8. Percentage of students in error plotted as a function of opportunity (Learning curve) for a KC (combining non-variable terms in force summation) for which error rate is initially high and remains high.

Typical learning rates with existing tutors[23] correspond to slopes in the range of 0.05 to 0.15. To see the rate of improvement that such slopes imply, say the probability of a student first making an error is 0.5. With a slope of 0.1, the error probability drops to 0.40 at the fifth opportunity and to 0.29 at the tenth opportunity.

The fit of the statistical model to the data yields the initial error rate (a_j) and the slope (b_j) for each KC. Table 1 displays the results for the distinct KCs grouped according to the three phases of solution: selecting subsystems, drawing FBDs, and registering results. Within each phase, the KCs have been ordered by increasing intercept (in sample 1). The observations now described hold for both samples. A number of KCs with lower intercepts have quite high slopes, for example, section_as_subsystem, unknown_internal_consistent, and determined_support. The tutor is playing a valuable role if it helps students master skills, such as these, that they did not initially possess. Thus, high slopes are most critical in the case of low intercept KCs. By contrast, other skills tend have a low initial error, corresponding to high value of intercept. For a few of those skills, such as unknown_support_new_pin and unknown_support_new_roller, the slope is again high, but for other skills, the slope is low. In any event, rapid reduction in the error rate with practice (high slope) is less critical if the initial skill level is relatively high.

Table 1. Statistical fit of multiple Knowledge Component learning model: initial fraction correct (Intercept) and decrease of error fraction with practice (Slope) for different Knowledge Components as predicted by the fit.

	Sample 1		**Sample 2**	
KC (Select Subsystem)	**Intercept (a_j)**	**Slope (b_j)**	**Intercept (a_j)**	**Slope (b_j)**
section_as_subsystem	0.74	0.20	0.68	0.26
joint_as_subsystem	0.94	0.00	0.93	0.01
full_truss_as_subsystem	1.00	4.54	0.98	0.31
KC (Draw FBD)	**Intercept**	**Slope**	**Intercept**	**Slope**
unknown_internal_consistent	0.26	0.46	0.45	0.16
determined_support	0.51	0.32	0.64	0.30
determined_internal	0.79	0.06	0.67	0.10
unknown_support_new_pin	0.82	0.28	0.91	0.31
unknown_support_new_roller	0.89	0.12	0.87	0.38
unknown_new_internal	0.89	0.02	0.91	0.03
applied_force	0.90	0.04	0.81	0.10
free_pin	0.98	0.07	0.98	0.14
KC (Register Result)	**Intercept**	**Slope**	**Intercept**	**Slope**
register_support_force	0.87	0.06	0.91	0.03
register_internal_force	0.88	0.05	0.92	0.05

If one goes to the trouble of developing a model that distinguishes among different skills or knowledge components, one should expect it to be, in some sense, an improvement over a simpler model that treats each action by the student as an attempt to apply the same knowledge component. Table 2 shows the intercept and slope for the single KC model. The rate of learning is extremely low (slope = 0.002). A plot of the single KC learning curve is shown in Figure 9 – it can be seen that the error rate has enormous fluctuations. Treating all errors committed by the student as equivalent, that is, assuming there is a single "truss solving" skill, leads to the conclusion that little or no learning is occurring. By contrast, the multiple KC model identifies those steps that students find difficult, and that the practice improves their performance on those steps.

Table 2. Statistical fit of single Knowledge Component learning model; nearly zero slope indicates that improvement with practice cannot be detected when actions of all types are treated as instances of a single "truss-solving" skill.

	Cohort 1		**Cohort 2**	
Single KC Model	**Intercept**	**Slope**	**Intercept**	**Slope**
Single KC	0.86	0.002	0.89	0.002

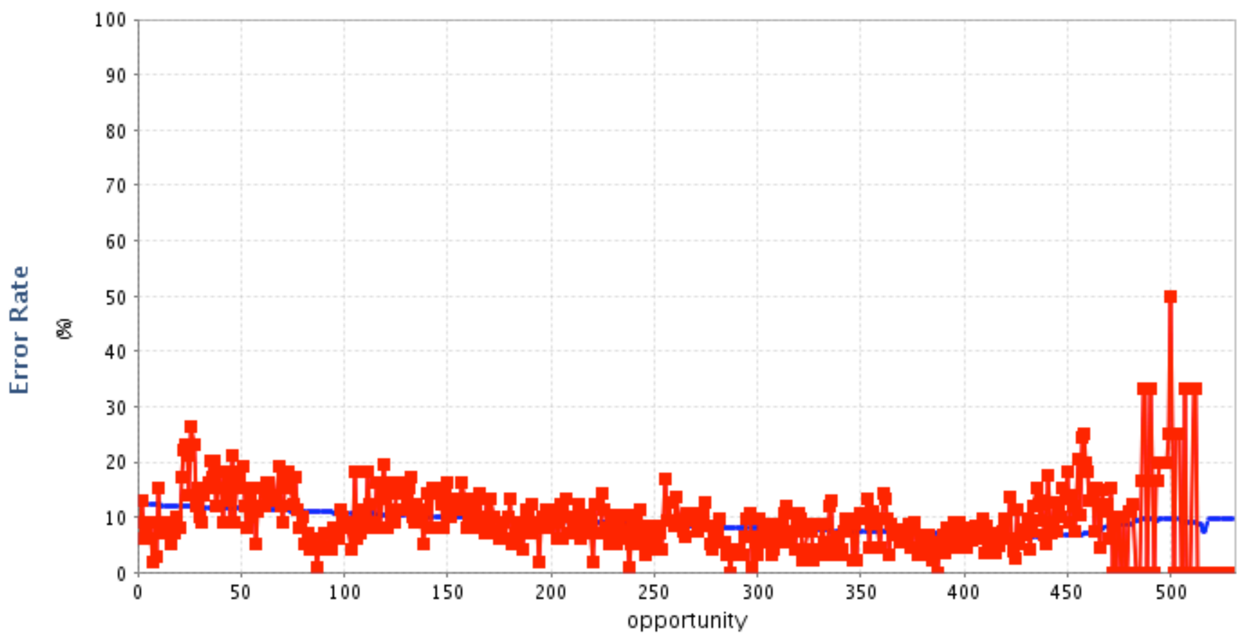

Figure 9. Percentage of students in error plotted as a function of opportunity (Learning curve) when treating all actions of a student as corresponding to the same KC; significant fluctuations and nearly zero improvement indicate that actions are poorly explained by a single KC.

We can also point to two quantitative measures suggesting that the multiple KC model is an improvement over the single KC model. When the datashop tools compute a fit of each statistical model to the data, several parameters pertaining to the goodness of fit are produced. In cross validation[24], the data set is separated into three groups. The model is fit to each group, the results for the other two groups are predicted using the fit parameters, and then the root mean square error (RMSE) between the prediction and the actual data is determined. In comparing two models, we can compare the RMSE from such three-fold cross validation. It was found that the RMSE of the multiple KC model is less than that of the single KC model (3.7% less for Sample 1 and 4.0% less for Sample 2).

A second means of comparing models is based on the Bayesian Information Criterion[25] (BIC). BIC is a measure of goodness of fit of the model to the data, but with a penalty that depends on the number of parameters in the fit. Namely, with two models that approximate the data equally well, the model with the fewer parameters (the quantities θ_i, a_j, and b_j here) will have the lower

BIC. Datashop computes a BIC value for each model; these values may be used to compare the statistical fits of different models to the same data set. To attribute meaning to BIC values for the different models, we appeal to a detailed study[26] of the statistical significance associated with a difference in the BIC values for two models applied to the same data. It was found[26] that differences in excess of 10 could be interpreted as implying a statistically significant difference between the two models. As seen from the BIC values reported in Table 3, the multi-KC and the single KC models differ significantly.

Table 3. Comparison of Bayesian Information Criterion (BIC) values for a multiple KC model and a single KC model, which treats all actions as corresponding to the same "truss-solving" skill. BIC differences in excess of 10 are treated as statistically significant.

Sample	Observations	BIC multi-KC	BIC single-KC	BIC Difference
1	9513	5762	6087	325
2	42827	23815	26372	2557

Finally we consider on what basis we should decide where to devote efforts to improve feedback. One can seek out in Table 1 those skills with insufficiently high intercept and insufficiently high slope. Most notable is the KC determined_internal: this corresponds to the skill of using a bar internal force that has been already determined in a new FBD where that internal force also acts. One must use the correct magnitude and interpret the earlier found tension or compression to draw the force in the correct direction in the new FBD. For both data sets, this KC does not have high intercept (0.79 and 0.67) and does not have a particularly high slope (0.09 and 0.10), at least not high compared to some of the other KCs. It is possible that feedback on this error can be altered; whether such alterations lead to improvement can be judged based on the new results for intercept and slope of this KC. As seen from the comparison with the single-KC model, the quality of the statistical fit is also strongly dependent on the choice of KCs and their assignment to student actions. Thus, improvements can also be sought in alterations to the KC model. These will be considered in future research.

8. Summary and conclusions

Learning to solve complex problems that involve analyzing multiple inter-related parts is a feature of many engineering courses. Such problems may have several or many solution pathways to correct answers. As with any learning, formative feedback to students on their initial efforts can significantly increase their ability to solve such problems. Students typically undertake such complex problems as part of homework, and their efforts are traditionally observed in the context of humans (instructors, TAs) grading those homework problems. Grading is usually to provide credit to the student; very little effective feedback to students results. Thus, we seek to determine if a computer tutor could be capable of providing feedback for complex problems, and how we might judge the effectiveness of that feedback. The challenge faced by such a computer tutor is to allow students to pursue multiple pathways to solution, and still be able to judge and give feedback on those efforts.

The approach taken here is for the computer tutor to have a cognitive model of a student engaged in solving the problems of interest. We explore this idea for a test case of trusses in statics.

Based on a task analysis of solving such problems, and a catalogue of student errors, a tutor for truss analysis was devised. The cognitive model consists of algorithms for correct forward steps that are applicable at any correct solution state, and a limited number of allowed incorrect steps that reflect typical student errors. By giving timely feedback after errors are committed, we can grant latitude to the student in solving while retaining ability to interpret work. Thus, in answer to the first research question, we have shown in this test case that it is possible to provide feedback on complex problems in which students can take a variety of solution pathways.

To respond to the second research question, how metrics can be devised to determine whether that feedback is effective, steps to solve truss problems with the tutor are hypothesized to involve a finite set of skills or knowledge components (KCs). Thus, each action by the student in solving any problem is categorized as an attempt to exercise one of the hypothesized KCs. From the saved student work we extract a sequence of opportunities to exercise each KC. The feedback from using the tutor may be deemed effective if the percent of students that commits an error in applying a KC decreases with successive opportunities. Since such learning curves (percent error vs. opportunity) are always noisy, we seek to determine how well a statistical model, derived from the power-law of learning and commonly used in other intelligent or cognitive tutors, fits the learning curves. Parameters in the statistical model include, for each KC, the intercept (fraction of students responding correctly at first attempt) and slope (rate at which the fraction in error decreases). Students in regularly scheduled statics classes at two institutions used the tutor for a weekly assignment in lieu of pencil and paper homework. Data from students in these two classes who consented to have their results used for research purposes were fit to the statistical models.

We found that the error rates for various hypothesized KCs differ significantly. From the fit of the statistical model, most of the KCs either had low error rates, or if the percentage of students who initially erred was higher, then that percentage in error decreased markedly with successive opportunities to practice. Thus, for most skills, students already had the skill at the start or developed the skill in the course of using the tutor. Such results constitute metrics that can be used to judge whether feedback is effective. The benefit of distinguishing between different KCs, rather than simply viewing all steps as associated with a single "truss-solving" skill can be seen from multiple comparisons to results from a fit to the alternative single skill model. Finally, in answer to the third research question, the learning curves themselves and quantitative results of the model fit point to specific parts of the tutor where the feedback could be improved or where an altered breakdown into knowledge components may give clearer evidence of learning. Future research will be aimed at determining whether such improvements can indeed be realized.

Acknowledgements

We thank Jackie Yang, Jeremy Jiang, and Rebecca Piston for their assistance in development and initial testing of Truss Tutor, and Ken Koedinger for insights into the learning curve analyses. 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. 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.
2. 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.
3. Lewis, M. W., & Anderson, J. R. (1985). Discrimination of operator schemata in problem solving: Learning from examples. Cognitive Psychology, 17, 26-65.
4. Lee, A. Y. (1992). Using tutoring systems to study learning: An application of HyperCard. Behavior Research Methods, Instruments, & Computers, 24(2), 205-212.
5. 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.
6. 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.
7. Roselli, RJ, Howard, L and Brophy, S. (2006) A computer-based free body diagram assistant. Computer Applications in Engineering Education 14: 281-290.
8. 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.
9. Sleeman, D. H., & Brown, J. S. (1982). Intelligent Tutoring Systems. New York, NY: Academic Press.
10. Wenger, E. (1987). Artificial Intelligence and Tutoring Systems: Computational and Cognitive Approaches to the Communication of Knowledge. Los Altos, CA: Morgan Kaufmann.
11. David J. Gunaratnam, A knowledge-based expert system for tutoring in structural engineering, Computers & Structures, Volume 30, Issue 3, 1988, Pages 767-773
12. 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
13. 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
14. Anderson, J. R., Boyle, C. F., & Reiser, B. J. (1985). Intelligent tutoring systems. Science, 228, 456-468.
15. Newell, A., & Simon, H. A. (1972). Human Problem Solving. Englewood Cliffs, NJ: Prentice-Hall.
16. P.S. Steif, 2004, An Articulation of the Concepts and Skills Which Underlie Engineering Statics, 34rd ASEE/IEEE Frontiers in Education Conference, Savannah, GA.
17. 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).
18. 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)
19. 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).
20. P.S. Steif, L. Fu, and L.B. Kara, "Development of a Cognitive Tutor for Learning Truss Analysis", Proceedings of the 2014 American Society for Engineering Education Annual Conference & Exposition, Indianapolis, IN, June 2014.

21. Koedinger, K.R., Baker, R.S.J.d., Cunningham, K., Skogsholm, A., Leber, B., Stamper, J., “A Data Repository for the EDM community: The PSLC DataShop.” In Romero, C., Ventura, S., Pechenizkiy, M., Baker, R.S.J.d. (Eds.) Handbook of Educational Data Mining. Boca Raton, FL: CRC Press. (2011).

22. Draney, K.L., Pirolli, P., & Wilson, M. “A measurement model for complex cognitive skill”. In P. Nichols, S.F. Chipman, & R.L. Brennan (Eds.), Cognitively diagnostic assessment (pp. 103–126). Hillsdale: Erlbaum. (1995).

23. Koedinger, K., McLaughlin, E., Stamper, J., “Automated Student Model Improvement,” In Proceedings of the 5th International Conference on Educational Data Mining (EDM 2012). Chania, Greece. Jun 19-21, (2012). pp. 17-24.

24. Model Values. In PSLC DataShop. Retrieved 11/18/13 from https://pslcdatashop.web.cmu.edu/help?page=modelValues

25. Schwarz, Gideon E. "Estimating the dimension of a model". Annals of Statistics 6 (2): 461–464 (1978).

26. Raftery, A.E., “Bayesian model selection in social research,” Sociological Methodology, 25, 111-196 (1995).