

2006-1697: AN INTEGRATED MODELING, ANALYSIS, AND AUTHORING ENVIRONMENT FOR STRUCTURAL/MECHANICAL ENGINEERING EDUCATION

Gregory Miller, University of Washington

Gregory R. Miller is a Professor of Civil Engineering at the University of Washington in Seattle. He received his B.S.C.E. from the University of Washington (1980), and M.S.C.E (1981) and Ph.D. (1983) degrees from Northwestern University in Chicago. He has been a faculty member at the University of Washington since 1984, working primarily in the areas of structural mechanics and computational modeling. He received a National Science Foundation Presidential Young Investigator Award in 1987, and has received numerous awards for his educational contributions at the college, university, regional, and national levels. In addition to producing various journal and conference publications, he has been involved in the development of numerous software packages for research, education, and professional applications. Several of these programs have evolved into commercial products and are used widely in education and practice.

An Integrated Modeling, Analysis, and Authoring Environment for Structural/Mechanical Engineering Education

Abstract

This paper presents an approach for technology-enhanced instruction based on embedding presentation capabilities within an interactive analysis environment. The idea is to combine the advantages of open-ended student exploration and experimentation with mechanisms for providing context, guidance, and formal instruction. The ultimate goal is to create a class of tools that are not only effective in helping students learn, but also are attractive to faculty in regards to constructing their own materials and activities. The paper presents both a description of the analysis/presentation environment and the evaluation outcomes of a pilot implementation focused primarily on the student learning side of the equation.

Introduction

It is generally accepted in the engineering education community that well-designed, technology-enhanced, student-centered learning activities can be effective in helping students learn. However, if one looks around at a typical university or junior college, one finds relatively few examples of such activities taking place compared to more traditional lecture/textbook approaches. There are many reasons for this, most of which revolve around the typical realities of time pressures, reward systems, and their effects on motivation. An argument can be made that an additional aspect of this is that faculty, like their students, are significant stakeholders in the educational process, and instructional approaches that give faculty the impression they are being marginalized are unlikely to take hold regardless of their effectiveness in regards to student learning outcomes. Adopting student-centered activities, materials, and exercises developed by others tends not to be particularly attractive proposition for many faculty, at least partially because this can be perceived as placing them more in the role of glorified Teaching Assistant than course author. One of the intrinsic attractions of lecturing is that it enables faculty members to frame a given topic in their own way based on their own understanding of the subject. Personification and personalization of knowledge and understanding are among the principal ingredients that faculty members bring to a course, and many are resistant to give this up. To the degree this argument is valid, one approach to help broaden the use of effective instructional techniques would be a model for technology enhanced, student-centered learning activities that also promotes faculty engagement, ownership, and authorship. This presentation describes work focused on the initial development of such a model within the specific context of structural mechanics and design.

Historically, there have been many examples of tools supporting faculty instructional authorship and communication that have been adopted readily once the barrier to entry has fallen low enough. Word processors, course web sites, and presentation software represent three ready examples. However, the nature of such generic tools is that they do not allow one to work within an environment that is discipline specific or discipline cognizant. Materials must be captured (frozen, in a sense) and imported from other discipline-specific sources, and therefore dynamic interaction with the resulting materials is limited, if available at all. Tools that are designed to support interactive content also typically work by requiring external content creation and capture. The work presented here is based on taking a different approach---rather than bringing

discipline-specific material into generic presentation environments, we are developing a model in which generic authoring capabilities are brought into a discipline-specific, rich simulation environment. In particular, we are using an advanced modeling tool with unique visualization capabilities coupled with presentation/scripting technology to enable the integration of guided explanation and conceptual development with unguided, student-centered exploration and experimentation. The idea is to provide enhanced mechanisms for communicating technical concepts by allowing flexible interplay of explanation and exploration.

The approach is intended to be general in concept, but the specific context of the current work is the field of structural and mechanical modeling, design, and analysis applications. The central component of the work is an analysis environment that supports *live modeling* of structural systems and components coupled with integrated scripting and presentation capabilities. Live modeling implies interaction with an analytical model directly via controls or mouse gestures, with the results of these actions presented immediately in an animated, quasi-real time fashion. In effect, the system being modeled appears “live” in the sense that it responds immediately and realistically to input from its environment. For example, to see how a structure responds to load, one can simply select a load tool, push on the structural model, and observe the response in parallel with one’s actions. The synchronized spatial and temporal cause/effect interaction makes it relatively simple to use this type of software and understand what one is seeing, while the advanced modeling capabilities and rich inspection mechanisms make it suitable for both basic and advanced applications. In instructional contexts, in particular, live modeling provides an excellent framework around which one can design educational activities that encourage experimentation and exploration, and that can accommodate various learning styles.

The work described in this paper is part of an ongoing project, and to date the focus has been on developing the necessary technology and ensuring that the student learning aspect of the approach is adequate to move on to considering the faculty side of the equation. This paper gives an overview of the computational environment in its current state of development and describes an initial evaluation based on a pilot learning activity. The scope of this pilot activity presented below is necessarily limited: by definition, general purpose authoring tools can be used in many ways and in many contexts, and it is nearly meaningless to try to make general conclusions about effectiveness devoid of some specific implementation or application example. However, as soon as one chooses a specific implementation or application example, the linkage back to the underlying tools and processes becomes muddled. For example, it is essentially meaningless to ask the general question: “are web sites effective in helping students learn?”, because the variety of web sites and possible associated activities mean that there is no simple answer, and any conclusions drawn based on the present state of the technology cannot be assumed to be valid for future developments. At best, one can determine whether a particular web site or class of web sites is or is not effective under given conditions, but this does not really answer the general question (the fact that a given web site is or is not effective does not inherently imply that a different site will or will not be effective). The trend in such contexts is to go the “Best Practices” route, but this is itself an ongoing, evolving process that does not necessarily provide a definitive answer to the general question. The author makes no claim that the technology presented here is anywhere in the same league as the web in terms of potential influence, but as an enabling platform for communication, it shares some of the same issues in regards to assessing effectiveness and potential. Therefore, the presentation here is perhaps best interpreted as trying to do two modest things: (1) to address the simple question, “Can the approach be effective?” via a specific, focused example; and (2) to illustrate the nature of the communication

mechanisms and overview the underlying technologies, again in the context of a particular example.

Live Modeling with Embedded Presentation

Figure 1 depicts the combined modeling/presentation environment in typical screenshot form. As the figure shows, there is a modeling window and a presentation window. The modeling window can be thought of generally as any suitable interactive analysis program, but in this specific case it is a structural analysis program that supports live modeling: Dr. Frame2D¹. This window and its associated tools and commands remain active at all times, so that a student can carry out any kind of modeling task at any time, driven either by their own curiosity or by instructions contained in the presentation. The static screen shot does not make it possible to illustrate the live modeling aspect of the program, but as described earlier, one can remove, add, or modify members, loads, and supports directly and observe the effects of such actions immediately. The particular model shown is a simple truss, but the range of modeling capabilities includes finite elements and nonlinear structural response, and there is also a 3D version of the program. It is advantageous to have relatively advanced modeling capabilities available even when considering less advanced material, because often student exploration will take them beyond the limits of simple models, and observing the breakdown of simplifying assumptions can be an important learning experience.

The presentation window is the key addition, since it allows one to augment the analysis environment with context, additional media, and storytelling. The presentation window has a paging mechanism that controls progress through a given script, which can contain both display information and software commands and actions. This enables the presentation script to operate the analysis program while providing commentary and explanation. Combined with the live modeling window, this can lead naturally to a relatively seamless blending of showing/demonstrating and student exploration. For example, displayed text might read “Observe what happens when we remove a support”, the program can then remove the support and display the results, followed by a subsequent display page adding observation commentary appropriate to the resulting model behavior, such as images of actual collapsed or damaged structures that were subjected to the same kind of action. Furthermore, unlike any kind of canned presentation (even if nonlinear navigation is available), because the modeling environment is always live, students can interact with the model as they wish at any time during the presentation sequence. This can allow either the spontaneous question “I wonder what happens if I do this now?” to be explored immediately, or students can be guided by the presentation to try things on their own (“Try putting a load at joint *A* and watch what happen”).

The scripting and presentation capabilities currently are based on an Extensible Markup Language (XML)² scripting framework that combines standard HyperText Markup Language (HTML)³ with a set of command tags that allow one to coordinate the display of a series of local web pages in an integrated browser window with corresponding program actions. Figure 2 contains a short excerpt of such a script illustrating the basic structure of the format. The script is composed of a series of steps, with each step containing a set of program actions (optional) and associated HTML display text. One can either embed HTML directly as shown in the second step, or simply refer to HTML files as shown in the first step—in either case, the HTML can be composed using any HTML editing tool. Currently the program commands must be composed manually: in the long run the plan is to make a higher-level tool for this task.

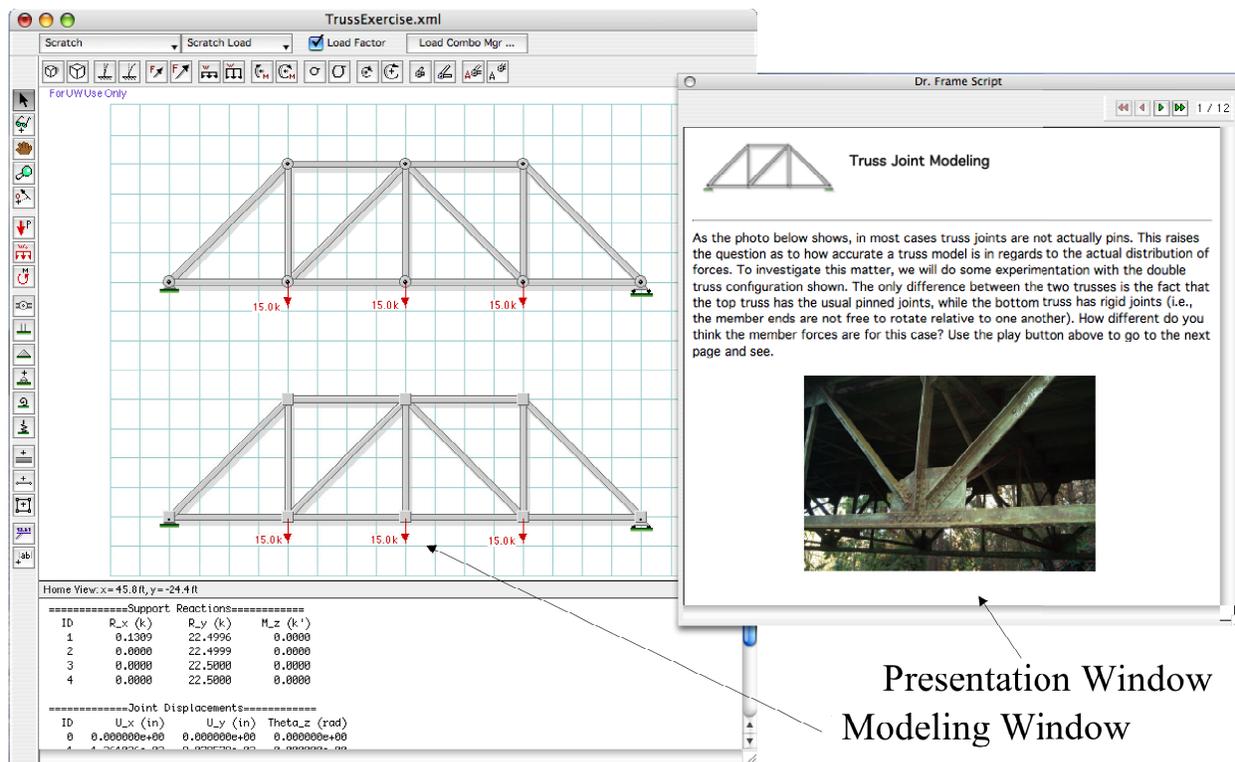


Figure 1: The combined modeling/presentation environment

Using this command/display framework, one can combine the familiar and generic presentation capabilities of HTML-based browsing with a full-featured modeling environment. With a system like this, the work required to create a discipline-specific, content-rich presentation or tutorial is reduced dramatically because the environment is already populated with the behaviors and objects one is trying to explain. The ability to intersperse guided and unguided exploration can provide expanded opportunities for faculty (and student) authoring at varying levels of formality. This can range from constructing a brief and simple sequence for a lecture (pure presentation) to putting together a standalone tutorial (part presentation, part student-directed learning) to assigning open-ended design or analysis tasks (fully student-centered). Among other things, this kind of flexibility allows an instructor to incorporate different styles of instruction incrementally.

Pilot Implementation Description

To gather some initial formative data concerning student impressions and learning outcomes, a pilot activity was implemented in a junior-level structural engineering course (CEE 379). As part of a standard assignment, students were asked to complete an online pre-survey/quiz, go through a lesson based on the presentation/modeling environment described above, and finally to complete a post-survey/quiz. The pre- and post-surveys both contained the same three questions designed to assess changes in students' understanding of particular concepts, and in both surveys students were also asked to rate their level of confidence in their answers. The post-survey included additional questions concerning students' impressions of the value of the overall activity.

```

<?xml version="1.0" ?>
...
<ScriptTitle> Dr. Frame Script </ScriptTitle>
...
<StepList>
  <Step>
    <Action> read DoubleTruss.drf </Action>
    <dfurl>DoubleTruss1.html</dfurl>
  </Step>

  <Step>
    <Action> cmd cmd_ToggleMemberForces </Action>
    <Action> cmd cmd_ArrowTool</Action>
    <Action> 2D mouse ClickDrag2 (12.5 10) (12.5 5) (12.5 5) </Action>
    <html>
      <h3> Truss Joint Modeling</h3>
      <hr>
      <p>With member force display turned on, it is easy to compare member forces for
      the two configurations. As you can see, the axial force values are nearly equal for
      the two cases. What would you say the difference is as a rough percentage?</p>

      <p>Go ahead and try changing the loads and see what difference this makes. You
      can toggle displacement display (type the 'u' key) to see how the displacements
      compare.</p>
    </html>
  </Step>
  ...
</StepList>
...

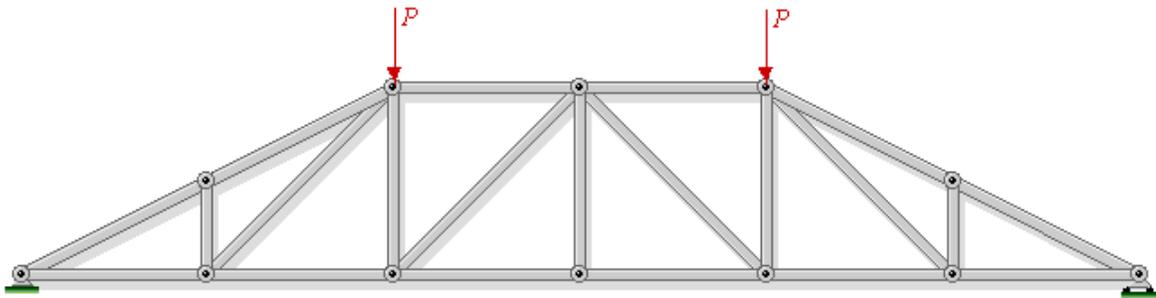
```

Figure 2: XML-based scripting framework combining program actions and HTML content.

The learning objectives for the pilot activity focused on practical behavioral issues associated with trusses. The course itself is primarily an analysis class, with a heavy emphasis on the stiffness method in the context of truss, beam, and frame structures. In such a class, it can be difficult to bring in qualitative aspects of structural behavior, and so the exercise was set up to provide an opportunity for students to investigate these kinds of issues in a self-study mode. Two specific aspects of truss behavior were singled out for the pilot exercise: (i) the effect of joint restraint and related practical details on truss response; and (ii) the form/function alignment that occurs for trusses whose shape matches a loading's funicular curve. In more general terms, this pilot study could be categorized as an exercise aimed at having students learn material that would otherwise not be covered/discussed at all due to time limitations. Therefore the baseline level of instruction would be approximately zero, and the pretest part of the exercise indicates the associated level of student baseline expertise. Consistent with the target scenario, other than a brief overview similar to typical homework assignment comments, no class time was taken to discuss the exercise, explain the use of the software, or go through the underlying technical concepts.

The three questions the students were asked are given below:

-
1. The joints in a real truss are rarely actual pins, which you are likely to have noticed when looking at typical bridge or roof trusses. For a typical truss, what kind of error would you expect in regards to computing axial force values based on using a pinned-joint assumption versus a restrained-end (i.e., rigid connections) assumption?
 - a. Less than 1%
 - b. About 5 %
 - c. About 10%
 - d. About 20%
 - e. About 40%
 2. Which of the following factors would you expect have the biggest influence on the difference between pinned versus restrained end axial force values for a truss?
 - a. The span of the truss.
 - b. The materials used to make the truss.
 - c. The joint details.
 - d. The slenderness of the members.
 - e. The shape of the truss.
 3. Without doing any calculation, about how many zero-force members are there in the truss loaded as shown below?



- a. 0
- b. 1
- c. 2
- d. 3
- e. 9

The learning exercise consisted of students going through a script that consisted of two sections. The first section looked at the issue of joint restraint, presenting two trusses with identical geometry, loading, member properties, and support conditions, but with different joint restraint in the members (one model had fully pinned joints and the other fully restrained joints: see Figure 1). The loads and members were coupled in such a way that whatever action one took on one truss, the same effect would apply to the other, so one could conveniently modify loads, geometry, and member properties and compare immediately the effect on the response between the two models. The script provided some brief background information, illustrated the effect of some simple modifications, and then encouraged students to try their own variations. A conclusion was then suggested, and students were given the opportunity to consider whether they agreed or disagreed.

The second section of the script introduced the idea of the funicular curve for a load and presented some simple examples. Example trusses were introduced, and students were then shown and asked to perform by themselves geometric modifications such that when the trusses matched the funicular shape, all non-chord members became zero-force members. Once again, a conclusion was offered for student consideration based on their observations.

The entire exercise was set up to take about 15-20 minutes on average, with the pre- and post-surveys adding about 5-10 minutes to the overall process. Students were given no formal training with the software, and in most cases this was their first exposure to it. The ability to use a show/do mechanism was key in enabling new users to do nontrivial tasks quickly and without undue learning curve distraction.

Pilot Implementation Evaluation

The evaluation instrument described above was set up in collaboration with the Office of Educational Assessment (OEA) at the University of Washington, and all data gathering and evaluation were performed by OEA independently of the author. The majority of this section reflects verbatim summaries of the preliminary evaluation report provided by OEA.

The surveys were presented as online questionnaires at the beginning and the end of the exercise. Within the assignment, an introduction to the evaluation project was presented along with a link to the survey itself. Students' participation was voluntary, although to receive credit for the assignment, they had to at least answer whether they consented to participate in the surveys. A total of 30 students completed a pre-survey, 32 completed a post-survey, and 28 completed both a pre- and post-survey. The total class enrollment was 38.

Figures 3-6 present summaries of student responses to the questions common to both the pre- and post-surveys. Based on these results, the OEA evaluator was able to report the following:

Participants did make conceptual gains before and after going through the Dr. Frame exercise. For each of the conceptual questions, students were most likely to provide an incorrect answer, with percentage of correct responses increasing significant on the post-survey. Participants also demonstrated more confidence in their responses on the post-survey as compared to the pre-survey, with ratings for all three ratings increasing significantly (according to matched-pairs t-tests, $p < 0.001$).

Although these results do not provide any information about how this approach compares to other instructional methods, they do indicate that the approach in and of itself can be effective in regards to student learning. For the specific case of structural engineering, the results imply that certain kinds of topics that are not typically covered in normal lectures and textbooks could be introduced to students via this kind of mechanism.

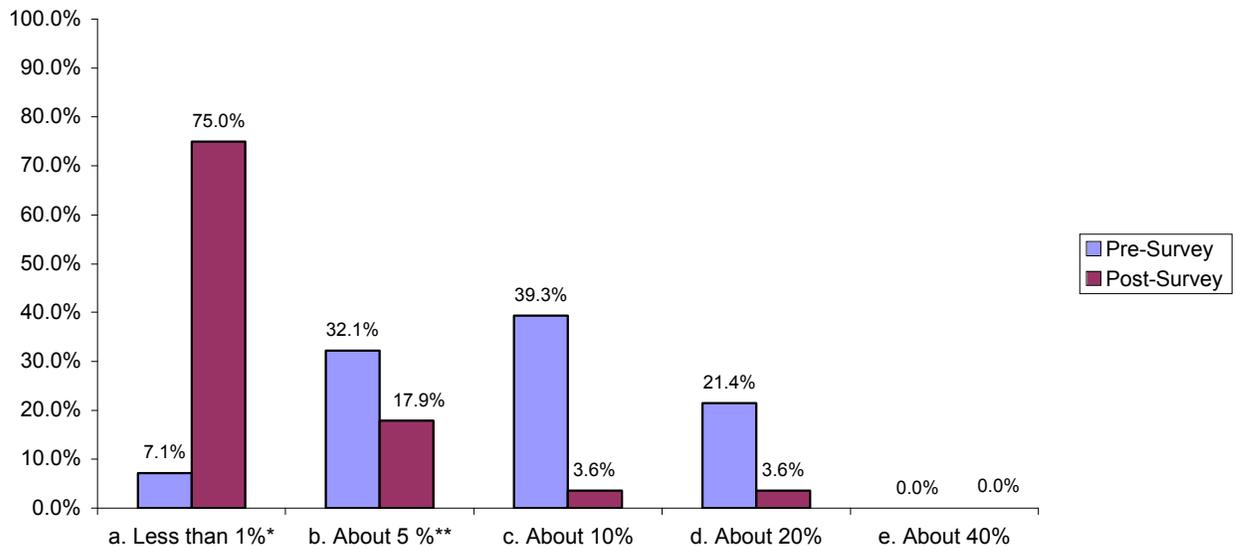


Figure 3: Percentage of individuals providing each response to Question 1 on the pre- and post-survey. A (*) indicates the best answer, with (**) representing an acceptable answer.

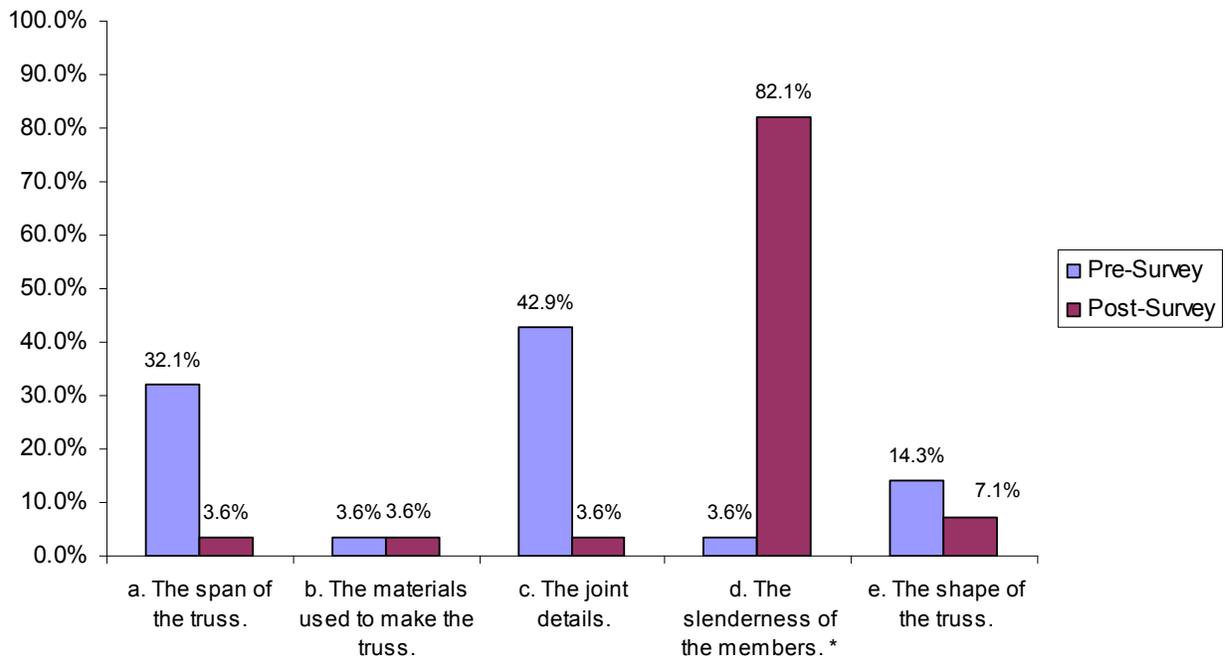


Figure 4: Percentage of individuals providing each response to Question 2 on the pre- and post-survey. A (*) indicates the best answer.

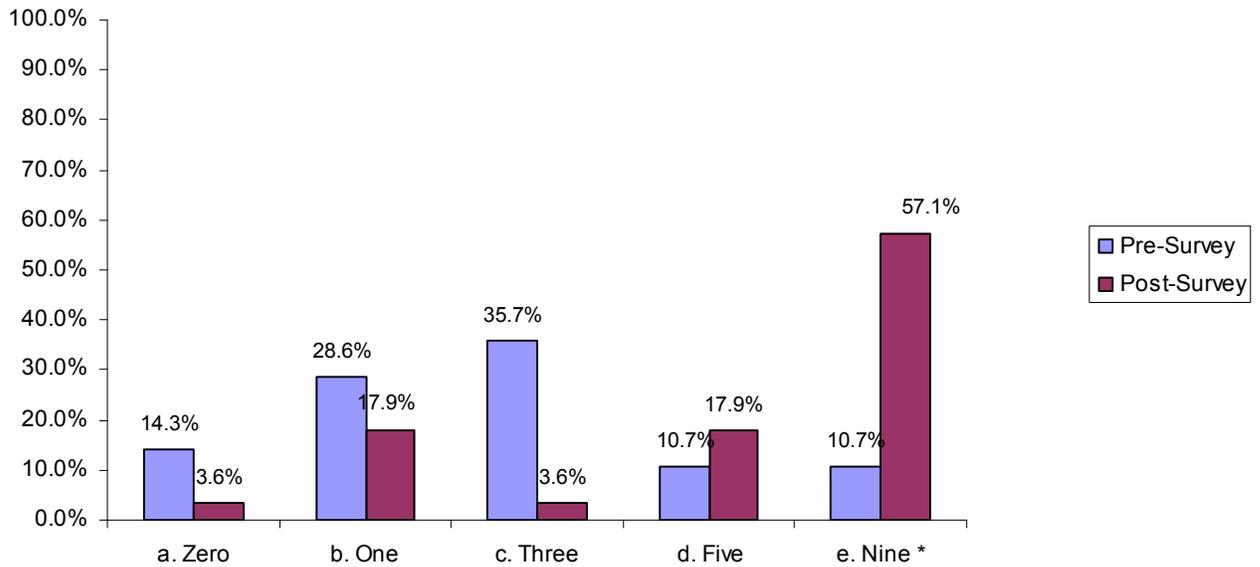


Figure 5: Percentage of individuals providing each response to Question 3 on the pre- and post-survey. A (*) indicates the correct answer.

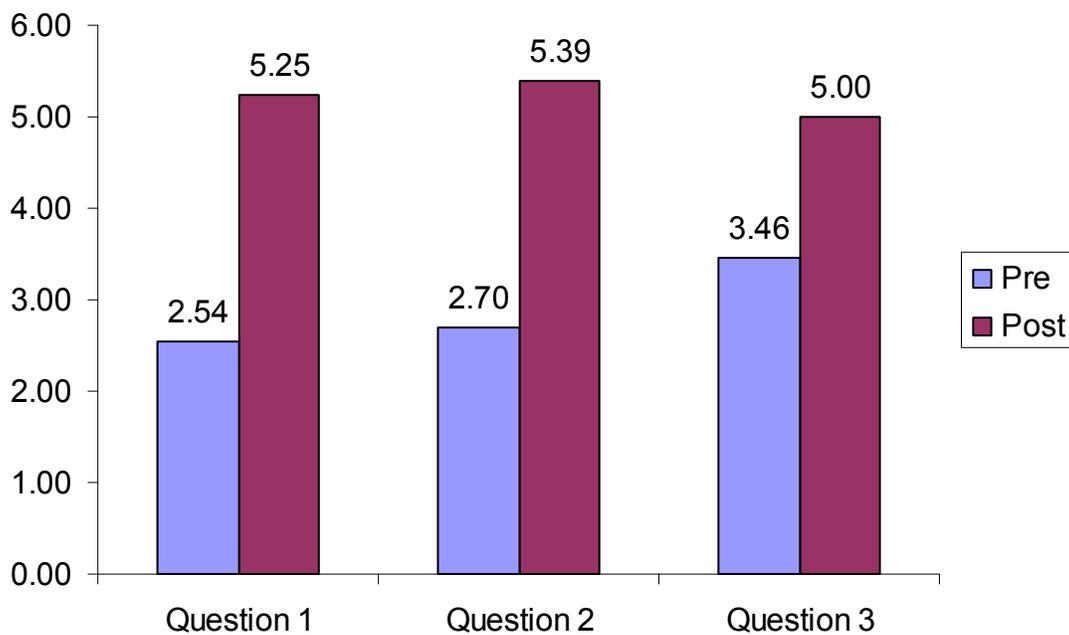


Figure 6: Mean confidence ratings (on a scale of 1: Not at all confident to 6: Completely confident) provided by participants about their responses to three conceptual questions on the pre- and post-surveys

The evaluative questions on the post-survey included a set of six statements for which participants were asked to report the degree to which they agreed or disagreed with the

statements. Table 1 shows the frequency, mean and standard deviation of their responses to these questions, and Figure 7 summarizes the mean response in a more visual form.

Table 1: Descriptive statistics for six evaluative statements.

	Strongly Disagree	Disagree	Neutral	Agree	Strongly Agree	Mean	SD
	1	2	3	4	5		
This activity enhanced my understanding the behavior of trusses.	0 (0.0%)	0 (0.0%)	2 (6.1%)	19 (57.6%)	12 (36.4%)	4.30	.585
The software was easy to use.	0 (0.0%)	0 (0.0%)	0 (0.0%)	17 (51.5%)	15 (45.5%)	4.36	.783
I enjoyed this activity.	0 (0.0%)	2 (6.1%)	1 (3.0%)	17 (51.5%)	13 (39.4%)	4.24	.792
This activity was a good use of my time.	0 (0.0%)	1 (3.0%)	3 (9.1%)	15 (45.5%)	14 (42.4%)	4.27	.761
Now that I've completed this assignment, I will probably have to study less for the final exam.	10 (30.3%)	10 (30.3%)	6 (18.2%)	6 (18.2%)	1 (3.0%)	2.33	1.190
If given the opportunity, I would like to return to this activity as a study resource.	0 (0.0%)	3 (9.1%)	5 (15.2%)	14 (42.4%)	11 (33.3%)	4.00	.935

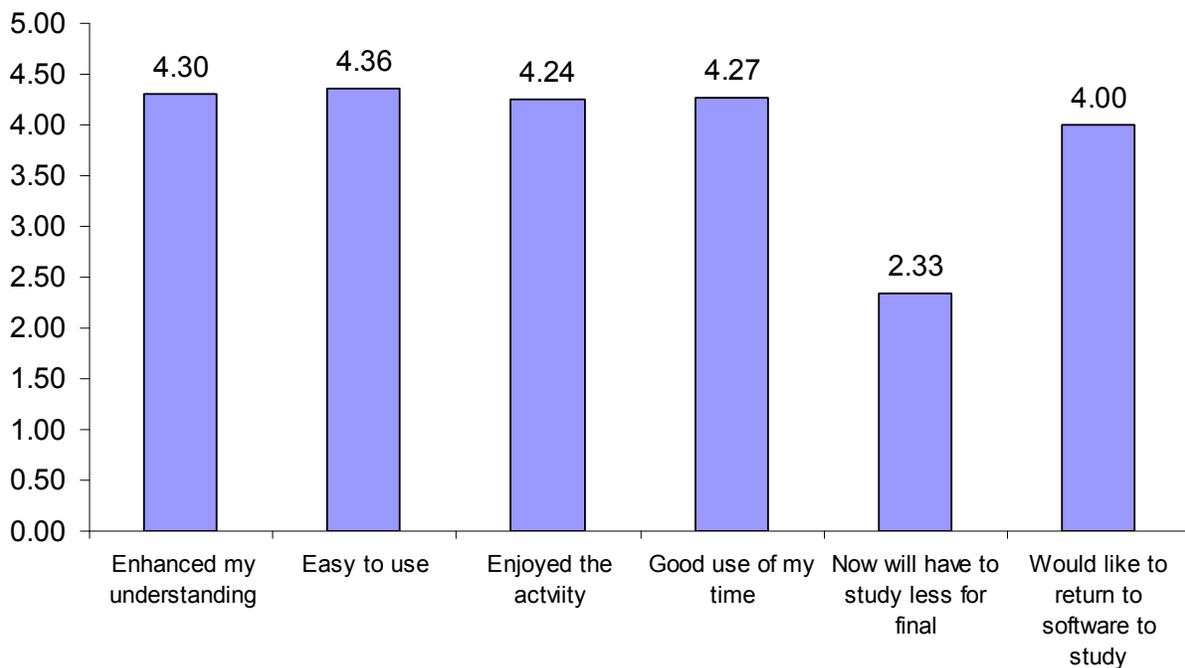


Figure 7: Mean ratings for each of the rated evaluation items on the pre-survey.

In addition to the directed evaluative statement summarized above, participants were asked four open-ended questions:

1. What, if anything, did you find particularly valuable about this activity?
2. Did you have any technical problems with the software? If so, please describe these in detail below.
3. How, if at all, do you think this activity could have been improved?
4. Any additional comments about the activities?

In the interest of space, the individual responses will not be reproduced here, but the OEA evaluator's comments were as follows:

For the most part, responses to the Dr. Frame software were overwhelmingly positive. Almost all students agreed that they enjoyed using it, that it helped enhanced their understanding of certain concepts, it was easy to use, and was a good use of their time. There was some indication that students did not necessarily recognize the activity as a replacement for studying. Students provided a number of suggestions for how the activity might be improved and it appears there were only a few minor glitches in using the software, but nothing that stood out as frustrating for all students; rather they were idiosyncrasies of the individuals' knowledge or the computer they were using.

Similar to the case of the learning outcomes, these remarks suggest that the approach can be effective in regards to student response.

Conclusions

This paper has presented a strategy for computer-enhanced instruction based on combining presentation with simulation and analysis in such a way that guided and unguided activities can be blended in a relatively seamless and flexible manner. The results of an evaluation of a pilot implementation indicate that the approach potentially can be used to construct activities that are effective in terms of enhanced student learning and student response. This is a necessary but not sufficient criterion for such an approach to be effective generally: it still remains to be determined if faculty respond positively to this kind of tool in their instruction. Efforts to address this question are currently underway.

Disclosure

It should be noted that the author of this paper has a financial interest in Dr. Software, LLC, the company that produces Dr. Frame and related products.

Acknowledgments

This material is based upon work supported by the National Science Foundation under Grant No. 0442746.

Bibliography

1. Dr. Frame, Dr. Software, LLC, <http://www.drframe.com>, 2005.

2. Extensible Markup Language (XML), World Wide Web Consortium (W3C), <http://www.w3.org/XML/>, 2003.
3. HyperText Markup Language (HTML) Home Page, World Wide Web Consortium (W3C), <http://www.w3.org/MarkUp/>, 2003.