Abstract

This paper will discuss ten different Statics instructional technology programs that were considered for classroom use and the positive aspects of each as well as some of their shortcomings.

While in the process of a larger research project investigating the effects of technology use in the classroom, a number of different readily available instructional programs designed for use in Statics courses were studied. The programs were being considered for use in some sections of Statics being taught at Georgia Tech to assist students with the truss analysis portion of the course. Current and former teachers of the course were surveyed to see what type of characteristics they sought in an instructional program for trusses. Ten popular and readily available software titles were then obtained from developers, textbooks, and bookstores. The titles were then evaluated to determine the extent to which they matched the desired pedagogical characteristics outlined by the instructors and supported by learning theory. The findings revealed that while all of the programs did meet some of the evaluation criteria, no one title fulfilled all of the instructors’ needs. This paper will describe desirable software features based upon pedagogical theory and present the list of software characteristics requested by faculty members. In addition, the functionality of each of the pieces of software is discussed and the program that was chosen for the aforementioned larger research project is identified as well as the reasons why it was chosen.

Introduction

An ongoing research project investigating the long-term effects of technology use in the classroom is being conducted at Georgia Tech. A portion of this project involved selecting software to use in the classroom in order to study the effects of its use. The course in which the technology was implemented was Statics; a course with which many engineering students and instructors are very familiar. Furthermore, this technological intervention involved just one portion of the course, the truss analysis topic. A number of readily available pieces of software exist for use in assisting with the instruction of truss analysis and so a method by which to decide which title to use in order to best serve the students was employed. This method involved comparing the software programs to established theories of learning as well as assuring that the programs contained characteristics deemed to be desirable by the instructors of the course.
brief review of learning theory will be presented here and will be followed by a list of desirable software characteristics compiled by Statics instructors. The software programs considered for class use will then be described and finally the programs will be compared to the theories of learning and the list of characteristics to see if any of the titles meet the needs of the research.

**Learning Theories**

There are a number of different theories about how learning is actually accomplished. New developments in learning started over half a century ago with rise of behaviorism\(^1\), which was built upon and followed up by instructional design theory\(^2\). In both of these theories, the instructor possesses knowledge and skills, which he or she then passes on to the students. In opposition to these theories is the theory of constructivism, which states that knowledge cannot be passed on or even taught but must be constructed by the learner. Many leaning models have evolved from, or in opposition to, constructivism. Some of these include constructionism\(^3\), social constructionism\(^4\), situated cognition\(^5\), cognitive flexibility\(^6\), and case-based reasoning\(^7\). These theories differ in the recommended approach to teaching and learning in major and minor ways depending on the theory to which one subscribes. There are some, however, who believe that differing theories of learning can be integrated. Greeno (as cited in Sulbaran\(^1\)) proposes options for integrating behaviorism, constructivism, and situated cognition. Dick, on the other hand, acknowledges major differences between constructivists and instructional designers but does mention that both sides could be improved by learning more about the other\(^2\).

The purpose of this paper is not to describe or promote any or all of these learning theories but to explain how tenets of these theories were used in evaluating instructional software. Despite the many different theories of learning, there are several common practices among most of the more popular models. Most agree that authentic contexts, activities, or problems are required to increase learning, retention, and transfer. The amount of real-world context necessary for good instruction varies between theories but most agree that it is necessary. Many theories also agree that students should be allowed to experiment within these real-world contexts. Open-ended models, problems, or environments that can be manipulated, created, or experimented on allow the students to construct their own knowledge about a particular context. The authenticity of this context facilitates in the transfer of this knowledge to future situations.

Based on these learning theories, this paper postulates that for a piece of educational technology to be pedagogically sound it must include the aforementioned characteristics. Namely, it should include real-world scenarios, situations, problems, and environments. Also, students should be able to experiment within these real-world contexts to construct their own knowledge. Furthermore, Mayer and Chandler\(^8\) cite evidence and experiments that show that multimedia explanations, consisting of narration and animation, results in improved performance and problem-solving transfer. Narrated animations have been understood to be effective because they reduce cognitive load by utilizing the student’s auditory channel as well as the visual channel thus leaving more room in the cognitive working memory to process the information. This use of auditory information in conjunction with visual presentations has long been supported by leaders in engineering education as an effective use of technology\(^9\).
Desirable Characteristics as Determined from Faculty

In addition to identifying pedagogically sound technology characteristics supported by learning theory, input from statics instructors was also an important part of the decision-making process. There are approximately eight sections of statics taught each semester by a number of different faculty members and instructors at Georgia Tech. Because these are the classes wherein the technology would be implemented, six faculty members who teach statics were interviewed and given a survey about the characteristics they would most like to see in an instructional program for use in statics classes. A list of desirable characteristics was compiled based on this survey. Then three statics instructors were chosen to review the list for completeness and accuracy. This same group of instructors then evaluated a number of software programs to determine if the programs had these characteristics.

The list of desirable characteristics as determined by the faculty includes usability issues, such as attractive interface, in addition to pedagogical issues, such as models or problems that can be designed, played with, or experimented upon. This list of desirable software characteristics follows and has been broken into a few categories for simplification.

- **General Characteristics**
  - Interactive
  - Self paced
  - Easy to Use
  - Available and Accessible

- **Interface Characteristics**
  - Easy Navigation
  - Clear Instruction
  - Clear Links
  - Attractive Interface
  - Standard or intuitive buttons, menus, and icons
  - Appropriate and helpful feedback

- **Content/Context Characteristics**
  - Integrates With the Curriculum and Profession
  - Appropriate vocabulary and subject matter
  - Real world problems

- **Style Characteristics**
  - Multimedia which adds to the presentation
  - Visual

- **Constructivist Characteristics**
  - Open-ended models or problems that can be designed, played with, or experimented on

There is some overlap in these characteristics, for instance if a program does not contain clear instructions then it likely will not be easy to use. All of the instructors that were interviewed and surveyed agreed that, while all these characteristics were important, the single most important characteristic was the ability to manipulate objects and see how these manipulations affected the results. Specifically, with respect to trusses, the instructors wanted a program that enabled students to design trusses and to adjust certain parameters and see the results of these adjustments. An example of this type of adjustment would be the ability to reverse the direction of a point load on a truss and see the results of this change on the internal member forces. This important characteristic is consistent with the learning theories outlined previously.

There is also overlap between the instructors’ list of characteristics and the list of features outlined by the learning theory literature (the open-ended constructivist activities, for example).
The complete list of characteristics that was used for this evaluation included the bulleted list above with the addition of some of the tenets from learning theory that were not put forth by instructors; namely the use of multimedia explanations of statics theory and audio explanations. These software characteristics were then used to evaluate a number of different software titles that were developed for use in assisting with the teaching of truss analysis.

**Software Titles**

While an attempt was made to evaluate all readily available software titles for use in statics, some lesser-known programs or programs that are not widely distributed may have been overlooked. Listed below are the programs that were evaluated with a brief description of each.

**Multimedia Engineering Statics**

This is an ebook complement to an entire online courseware management tool for statics and dynamics build by Kurt Gramoll at The University of Oklahoma. The ebook is broken up into sections and subsections. Each subsection follows the same general outline and is made up of four web pages, the first of which introduces a real world case, problem or situation and is called the case introduction. The second web page is where theory is presented, the theory that is used to solve the case. The third page is the case solution; here the theory is put into action to solve the case that was previously introduced. These three pages all have multimedia content such as pictures, short movie clips, and audio explanations. The last page in each section is a simulation page. In the simulations, the students are allowed to do some constructivist activities by

![Multimedia Engineering Statics screenshot](image-url)

Figure 1. Screenshot from Multimedia Engineering Statics.
constructing or manipulating situations. For example, in one of the truss simulation pages, the students are allowed to build a small truss of their own design and load it up and see the results. They can then change the loads, supports, or members and see how this affects the results.

Figure 1 is a screenshot from one of the theory pages of this program; notice the section menu on the left and the four pages (case intro, theory, case solution, and simulation) menu below the subsection menu at the top of the page. This page uses two movies, three audio clips, and text to explain the theory behind the method of sections for truss analysis.

**MDSolids**

MDSolids\(^{11}\) is the Premier Award (ASEE annual award for excellence in engineering education courseware) winning software developed by Timothy Philpot for use in mechanics of materials courses though a few of the modules are useful for statics topics also. MDSolids is divided up into ten different modules that each solves a particular type of problem. The determinate beam module, for example, allows the user to design a simple determinate beam and the program draws the shear and moment diagrams. The user chooses the types and locations of supports and loads. Once the diagrams are drawn, the parameters can be manipulated and the diagrams updated. This allows the students to experiment with different conditions and see, visually, how changing parameters effects the results. The modules that are appropriate for use in statics are trusses and determinate beams.

**Statics Tutor**

Statics Tutor\(^{12}\) is a CD that contains explanations of statics topics and includes examples and problems for users to solve. For the most part, the explanations and examples follow a slide show format for presentation of information not unlike turning the pages of a textbook. The problems on the CD are slightly more interactive and involve a number of steps; the user must get one step of the problem right before moving on to the subsequent steps. This type of format allowed the program to give intermediate feedback on problems instead of simply telling the user whether the final answer was right or wrong.

**Shaping Structures: Statics**\(^{13}\)

Although *statics* is in the title of this software, this title is more of an instructional program to teach structural analysis. This software gives excellent step-by-step examples of graphical solutions to eight different types of structural analysis problems. While trusses were one of the eight topics discussed, the method described in this software, the load line method, is one that is not normally taught in statics courses at the sophomore level but is reserved for more advanced structural analysis courses.

**BEST Statics**

This is a work in progress being developed by faculty and students in the Instructional Software Design Center at the University of Missouri Rolla. BEST\(^{14}\) is done in Flash, is available online,
and has a continually growing database of theory, examples and problems. There are three separate sites, one each for statics, dynamics, and mechanics of materials. Each individual site then has a table of chapters and sections with links to theory, examples, and problems for each section within a chapter. The Flash animations make some complex visuals easy to understand and especially illustrate 3-D graphics well.

Statics Tutorial

The Statics Tutorial by Beer and Johnston\textsuperscript{15} is included as part of the New Media Version of their popular statics textbook. This tutorial is very similar to the Statics Tutor mentioned previously and uses slide-show format and simple animations to present theory and examples. This tutorial also has drill-and-practice quiz banks that students can take but, unlike DeVore’s Statics Tutor, these problems do not include intermediate feedback but does provide students with correct final answers.

Working Model Simulations

Three popular textbooks include CDs with pre-set Working Model simulations. Bedford and Fowler’s Statics Study Pack\textsuperscript{16} and Hibbeler’s Statics Study Pack\textsuperscript{17} both include simulations using Working Model 2D and Beer and Johnston’s New Media Version of their statics text\textsuperscript{15} includes a CD of simulations that use Working Model 3D. In all three cases, the simulations are based on examples from the book and allow the users to adjust the parameters of the model and see the results. Users are not, however, allowed to design their own models using these pieces of software.

Problem Banks

Two textbooks, the aforementioned Bedford and Fowler study pack and the Hibbeler study pack, both include a password to an online problems website. These websites include problems and solutions from former versions of the textbooks that students can try on their own to solve. The problems are categorized according to the chapters in the accompanying textbook. Both versions also include multiple choice and true/false questions to test theory and application; these are available for use without a password. The Bedford and Fowler companion site also has online homework capabilities where homework is completed and graded online and grades are reported to the instructors.

Dr. Frame

Dr. Frame\textsuperscript{20} is a graphical structural analysis program that allows students to build and analyze any type of structural frame, including trusses. The analysis is performed and results are returned in real time. This real-time analysis enables students to instantly see the results of any changes they make. For instance, a student analyzing a beam can grab a point load, using the graphical interface, and drag it along the length of the beam to see instantly and graphically how changing the location of the load affects the reactions, internal forces, and internal moments of the beam.
Multimedia Learning Environment for Statics

This program was developed by Siegfried Holzer and Raul Andruet as an instructional aid for students in statics classes. The program presents theory, problems, and examples for a number of different statics topics. Navigation throughout the program is generally accomplished via links connecting one page of information to another. There are photos, diagrams, and simple animations throughout the program. There are also simple problems throughout that allow students to enter answers and then compare their results to the correct answers.

Evaluation Results

Evaluation of these ten software programs took place during the summer of 2002. As with many types of software, there are a number of different versions of these programs available; the references section lists the release date of the version that was evaluated. Please note that updates, additions, or corrections to the programs may have taken place since the listed release date or since this evaluation took place.

While all of the software titles evaluated had some of the desired characteristics presented previously, none of the titles met all of the criteria. It is not the intent of this paper to publicly disparage or criticize these programs and, as such, shortcomings will be discussed in general terms and not in reference to any particular piece of software. Also, some positive aspects of each program have previously been mentioned. It is not within the scope of this paper to discuss each of the titles in detail but the authors would be happy to give more information or answer any questions about any of the programs. Please feel free to contact them for more information.

The most serious shortcoming that was encountered during the evaluation of the programs was incorrect solutions to examples or problems. Two different programs evaluated had errors in the truss analysis portion of the program. While perfection is impossible to reach, in both cases they were very simplistic errors that should have been caught and corrected during editing.

All of the programs were self-paced and allowed the students to control the presentation of information although in a number of programs it was not always clear what the next step should be or what linking options were available at a particular time which, obviously, tended to hinder navigation. All programs contained some amount of interaction though the amount varied from program to program. With some programs, the only interaction the students had with the program involved pressing the Next button while others required students to design or create a model or problem. Unfortunately, not all programs were easy to use. Ideally, a student should be able to install a piece of software and being using it immediately. Some programs, however, contained no clear instructions as to what was expected of the user and, to complicate matters, some did not contain a help menu or file to assist users. Also, some programs were not self contained and required users to find and download additional fonts, extensions, or programs in order to run properly, which can tend to be frustrating to users.

Many other characteristics varied widely from program to program as well. Some programs had interfaces that were attractive, intuitive, and easy to use while others had very poorly designed
interfaces. Some programs had excellent multimedia simulations or visualizations that made complex problems easier to understand. In contrast, multimedia content of other programs consisted solely of static pictures, thus rendering them no more useful than a textbook.

Many of the programs lacked any useful feedback, either positive or negative, in response to user input. One program, for example, in response to an incorrect answer tells the user that the answer that they have entered is not correct because they chose the wrong answer choice. Clearly this type is feedback is completely useless to the user and does not help students identify and correct any misunderstandings.

With one exception, all programs integrated well with the curriculum and presented theory and analysis methods similar to those used in most statics classes.

Very few of the programs supported the constructivist-like activities that were identified by both learning theory and instructor interviews as being extremely important. Many of the programs simply presented theory and examples but did not allow students to experiment. At the opposite end of the spectrum are the programs that did allow students to adjust parameters on models or problems but presented no theory to students; these programs are mainly analysis tools.

**Software Selection**

In the middle of this spectrum is Multimedia Engineering Statics (MES). Although MES did not meet all of the evaluation criteria, it was the only program to fulfill the major requirements outlined by the instructors and supported by learning theory. MES presents theory in a multimedia rich format using text, narrated simulations, and audio commentary. MES uses real-world cases to introduce theory and then guides the user through the solution of these real-world cases. In addition to this, MES has simulations pages, which allow students to engage in constructivist-like activities. None of the other programs evaluated were as feature rich and yet still easy to use as MES. As a result of this evaluation, MES was chosen for use in a number of classes at Georgia Tech to supplement traditional instruction. The results of this supplemental technology use are still being examined but will be reported in the future.

**Conclusion**

The combination of instructor input and learning theory provide a desirable list of pedagogical features for instructional software and this list has been articulated in this paper. A basis for software evaluation can be formed around this feature list to provide an objective measure of educational technology programs. Such an evaluation was performed in this case, which led to the selection of a software title that is now being used in classes at Georgia Tech. Perhaps this list can help future software development activities improve engineering education.
References


**Biographies**

SEAN W. ST.CLAIR
Sean St.Claire is a Ph.D. candidate in the School of Civil and Environmental Engineering at Georgia Tech. He received a B.S. in civil engineering from Utah State University and a M.S. in civil engineering from Georgia Tech. Under the advisement of Dr. Nelson Baker, he has conducted numerous projects in the areas of engineering education and educational technology and assessment. Sean can be reached at sean.stclair@ce.gatech.edu.

NELSON C. BAKER
Dr. Nelson Baker is the Associate Chair for Information Technology and an Associate Professor in the School of Civil and Environmental Engineering at Georgia Tech. He is a nationally recognized leader in educational technologies designed for and applied to engineering having won numerous awards in this area. Dr. Baker’s research has led to the development and assessment of unique technologies such as virtual reality interfaces for education.