

2003

# Guidelines for the development of computer-based instruction modules for science and engineering

Jessica R. Lux

NASA Dryden Flight Research Center, [jessica.r.lux@nasa.gov](mailto:jessica.r.lux@nasa.gov)

Barry D. Davidson

Syracuse University, Department of Mechanical, Aerospace & Manufacturing Engineering, [bddavids@syr.edu](mailto:bddavids@syr.edu)

Follow this and additional works at: <http://surface.syr.edu/mae>



Part of the [Computer Engineering Commons](#), and the [Curriculum and Instruction Commons](#)

---

## Recommended Citation

Lux, Jessica R. and Davidson, Barry D., "Guidelines for the development of computer-based instruction modules for science and engineering" (2003). *Mechanical and Aerospace Engineering*. Paper 2.

<http://surface.syr.edu/mae/2>

This Article is brought to you for free and open access by the College of Engineering and Computer Science at SURFACE. It has been accepted for inclusion in Mechanical and Aerospace Engineering by an authorized administrator of SURFACE. For more information, please contact [surface@syr.edu](mailto:surface@syr.edu).

## Guidelines for the Development of Computer-based Instruction Modules for Science and Engineering

### Jessica R. Lux

Test Information Engineer  
NASA Dryden Flight Research Center  
PO Box 273 / MS 3302  
Edwards, CA 93523-0273, USA  
[jessica.r.lux@nasa.gov](mailto:jessica.r.lux@nasa.gov)  
Tel: +1 661-276-3820  
Fax: +1 661-276-2060

### Barry D. Davidson

Meredith Professor for Teaching Excellence  
Department of Mechanical, Aerospace & Manufacturing Engineering  
Syracuse University, 165 Link Hall  
Syracuse, NY 13244, USA  
[bddavids@syr.edu](mailto:bddavids@syr.edu)  
Tel: +1 315-443-4201  
Fax: +1 315-443-9099

### Abstract

The design and development of a prototype computer-based instruction (CBI) module is described. The module covers introductory structural analysis techniques and demonstrates the practicality of CBI for undergraduate engineering education. The module's generic learning objectives are used to develop a set of guiding principles for CBI development. Choice of authoring software and the module architecture are presented, followed by the details of the user interface, logic, and screen layout. The module design and software development process outlined here are easily extended to computer-based tutorials that present problem solving methods for engineering, mathematics, and the natural sciences.

### Keywords

CBI, Learning object, Engineering education, Statics, Structures, Structural analysis, Strength of materials, Problem-solving, Macromedia Authorware

### Introduction

Computer based instruction (CBI) is increasingly used in science, mathematics, and engineering education, and the reported benefits include higher learner interest, increased participation in coursework and improved outcomes (e.g., Croft et al. 2001; Edwards et al. 2001). CBI is generally most effective when used as a supplement to, rather than a replacement for, traditional engineering and mathematics education (Croft et al. 2001; Hummer et al. 1996), and there is certainly increased interest in using CBI in this fashion. As an integrated curricular component, CBI can be used in nearly any stage of the instructional process, including concept introduction and explanation, individual tutorial practice, and testing. For example, Holzer et al. (2000) reported using multimedia-rich CBI as a component of an engineering workshop learning environment, and students and teachers both reported favorably on the learning experience and educational outcomes.

In view of the above, the goal of the present work was to develop a CBI module that incorporated emerging "best practices" for module development that could also be used as a prototype for other modules in engineering, science and mathematics. That is, within the scientific literature, there appears to be a general consensus on those attributes that successful CBI modules should contain, i.e., modules that are received favorably by students and which result in a demonstrated higher level of learning. Our aim was to document these, incorporate them into a specific module and, in so doing, develop a generic architecture that could readily be reproduced in other content areas. Engineering statics, which is the study of rigid bodies at rest, and which is primarily concerned with the determination of the force distribution within bodies (structures) subjected to various external forces and constraints, was chosen as the initial focus area. Statics is perhaps an ideal application for CBI. Statics courses are taken by engineering students in a wide variety of disciplines and form the basis for many higher-level engineering subjects. Further, understanding statics is problematic for a reasonably large percentage of students.

Due to the level of demand, class sizes are typically quite large, thereby making high levels of student-instructor interaction outside of the classroom quite difficult. A quality CBI module focusing on statics therefore would have the potential of significantly improving outcomes for a high percentage of engineering undergraduates, not only in their statics courses, but in a variety of upper division courses that rely on this material. Also, a significant portion of statics education involves the learning of a specific problem solving methodology and then applying this method to a variety of generic applications. Thus, the subject matter is well suited for CBI, yet there are relatively few statics CBI modules reported in the literature. This is perhaps in part due to the fact that a commonly used introductory statics textbook is packaged with an interactive computer program for students to practice their problem solving skills (Beer & Johnston, 1997). However, although this program is quite good, it serves only as a database of problems; it does not review the concepts required to complete the practice problems, nor does it contain related and important reference information. There are other statics or statics-related CBI modules that have been reported (e.g. Holzer et al. 2000; Negahban 2000; Rais-Rohani 2001; Rojiani et al. 2000), but these demonstrate specific theory rather than emphasizing the problem solving methodology that was our focus. Based on our perception of the need for supplemental CBI in the area of engineering statics and our limited findings of existing modules that addressed this, it appeared that this subject provided an ideal pilot project for our work. We expect to subsequently expand this module to include more advanced topics in structural analysis, and for this reason the name "Structural Analysis Review and Tutorial" (START) has been chosen. The current module is a prototype, both in the local sense that it only covers a portion of the field of structural analysis, as well as in the more global sense that developers of learning modules in other areas of engineering, science and/or mathematics may readily adopt its architecture.

In what follows, the design of the START module is first described in a manner that makes the approach readily applicable to a wide variety of content areas. To this end, the module's generic learning objectives are used to develop a set of "guiding principles." These are extracted from the literature and may also be used to guide the development of other modules that focus on teaching problem solving methods. Next, the first level of development is described, which includes choice of authoring software, content and architecture. Finally, the details of the user interface, logic, and screen layout are presented, followed by a discussion of the practical issues surrounding implementation and assessment.

## **Learning Objectives and Guiding Principles**

The first step in module design is to define the learning objectives; these objectives should then drive the design of the final application. In this case, the goals for the module are for users to learn the general method for obtaining reaction forces on a statically determinant rigid body subjected to a set of known loads and boundary constraints, and then to demonstrate the ability to apply this knowledge in a variety of applications. Fundamentally, these goals are that users learn, practice and retain a specific skill set. Kristof & Satran (1995) indicate that realization of these goals is best accomplished through simple, clear presentation of content that provides remediation when deemed necessary by the user. Maintaining user orientation with a cohesive screen design is key to providing this straightforward content presentation. This is accomplished, in part, by a screen design that brings new content to a user's current screen, instead of sending the user to a new content screen (Kristof & Satran, 1995). In addition, a graphically embellished screen design should be avoided, i.e., one with excessive use of loud or contrasting colors, patterned backgrounds, and/or animated features, as this distracts the learner from both the content and the learning goals. As pointed out by Tufte (1990), such a design places the emphasis on the data container rather than the data. In extreme cases, such a design can even discourage the learner from using the module.

Balanced against the goal of a straightforward design is the goal of maintaining user interest. While the design should avoid distracting users from the content, it must at the same time interest users enough to keep them focused. One way to keep users engaged is to make them an active part of the learning process rather than passive recipients (Schank, 1993). Thus, a successful CBI module must have a high level of interactivity in order to maintain user interest and therefore successfully achieve its learning goals. The user's main interaction and learning control lies in his or her ability to navigate the content. Navigation must provide direct access, with a minimal number of steps between content, yet at the same time not overwhelm users with an excessive number of navigation options. Kristof and Satran (1995) suggest two design rules that should be used for each content link created. The first of these applies to cases where the link sends a user to a different location, in which case navigational controls must be provided so that the user can return to their original position without getting lost. The second rule states that if the link brings in additional content material to the current location, then that content must be designed to fit within the current location.

To maintain a high level of interactivity, an intuitive interface is necessary. Otherwise, users will be distracted from the subject matter content by spending time learning how to interact with the module. In addition, it has been shown that CBI modules can reach a wide range of learners at varying ability levels by allowing users to customize their learning experience. CBI should therefore provide an interactive interface that balances program control, in which users follow dictated learning sequences, and learner control, in which users select both the path and the pace through the material using self perception of their needs (Lin & Hsieh, 2001). If users are given complete control without guidance, then a novice user may become lost in the material. Conversely, if experienced users are forced to review material which they deem too easy or irrelevant, they are likely to avoid using the module. Thus, learner control and software pacing must be carefully balanced to maintain the interest of both novice and experienced users. Hannafin (1984) performed a review of the above issues and concluded with the following set of criteria for successful implementation of learner control: the audience must be older (success was reported with college-age students) and generally familiar with the content, the educational goals should be those of teaching a skill set, not imparting factual information, and advice must be provided to assist users in making decisions about the path and pace of instructions. Finally, like any other design process, CBI development requires an iterative approach. After a module is developed, feedback from both professionals and end-users must be solicited so that appropriate revisions can be made. Once a module has been implemented, continual assessment is necessary, which allows for the quality of both the content and the presentation to be increased. Table 1 summarizes the issues described above and provides a set of “guiding principles” for module design.

**Table 1.** Summary of guiding principles for module design

Attribute	Approach
Goal for Users	Teach a skill set
Presentation	Simple, clear, straightforward without the use of loud colors or patterns; consistent screen design that brings new content into the current screen
Architecture	Default path for novices, high degree of learner control for advanced users
Navigation	Controls so that user can return to original position; direct access to all content
Interface	Intuitive; high level of interactivity
Design Method	Iterative; solicit feedback from both professionals and end-users

### **Authoring Software, Content, and Architecture**

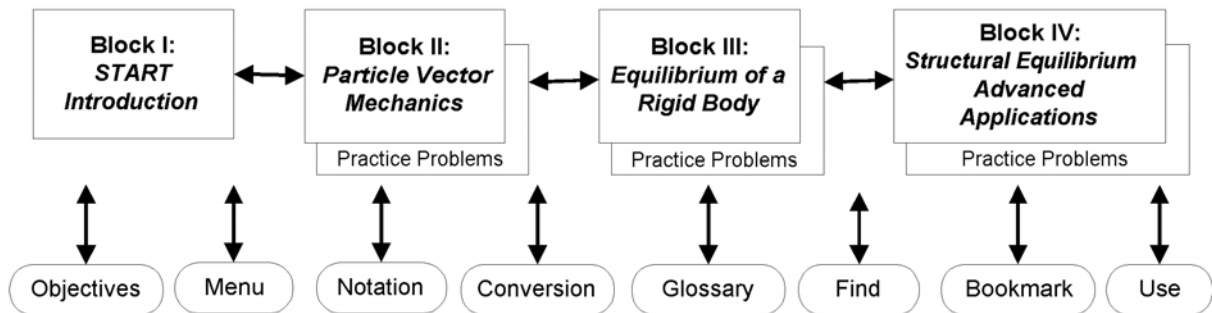
With the module’s learning objectives and guiding design principles defined, Macromedia Authorware 6 was selected as the authoring software. Authorware allows a designer to create stand-alone media-rich executables that accept user input, provide feedback based on this input, and foster total user control through the use of hyperlinked screens. In addition, user data and progress can be stored for later retrieval. Authorware executables can be delivered via the World Wide Web, a local area network, or CD-ROM. One of the most powerful features of Authorware is the easy access to the underlying program structure afforded to future designers and other members of a design team. Any designer familiar with the software can view the flowchart that was used to create a module. This is in sharp contrast with the lines of code that must be considered when using a low-level programming language. Also, as a result of the uniform data structures provided by this flowchart-authoring environment, Authorware readily allows integrating of components from different applications. Thus, members of a design team can work on components and integrate them seamlessly. These aspects of Authorware will allow future designers to use the architecture of START as a baseline template to create modules in other content areas.

Once Macromedia Authorware 6 was selected as the authoring software, the START module’s static content was organized for delivery. START reviews introductory structural analysis, illustrates the method for analyzing problems, and provides a series of practice exercises that gradually increase in complexity. START is not intended as the only educational medium on these introductory engineering topics, but rather is envisioned as an integrated component of a “typical” statics class that meets for lecture and perhaps recitation (problem solving sessions). The analytical method taught in START is described in Table 2. It is applicable to all statically determinant bodies at rest, and represents the building block for all advanced structural mechanics analyses to which students will be exposed.

**Table 2.** Analytical method taught in START

Step No.	Procedure
1	Resolve external forces into vector components
2	Set up a free-body diagram (F.B.D.) of the body to be analyzed
3	Set up the equations of equilibrium for the body
4	Solve the equilibrium equations for the unknown reaction forces

The next step was to develop the module architecture, which is shown in flowchart format in Figure 1. By determining a cohesive top-level architecture, the design and formatting of individual screens is a natural progression within that structure. The top-level architecture is defined by the general organization of information within the module, and the navigation between different blocks (or chapters) of the module. The “default path” through the module is the horizontal path through the four blocks along the top of this figure. Users do not have to follow this path, but for those that do, the progression is the introductory block, particle vector mechanics, equilibrium of a rigid body, and structural equilibrium & other advanced applications. The introductory block serves to introduce the entire module, and successive blocks introduce analysis methods for problem classifications of increasing complexity. Each block that follows the introductory block has its own introductory screen that describes its content and learning goals. Each of these blocks then presents content-specific information, e.g., the problem solving methodology and associated background material and definitions, followed by an example problem. Users then navigate through a series of practice problems. There is also a default path for practice problems associated with any particular block, which presents that series of problems in order of ascending complexity. In this manner, the learner is allowed to master a method for solving problems, rather than simply memorizing the module’s content. In many respects, the architecture of the module parallels chapters in a typical statics textbook, and for this reason should fit in readily with the preconceived notions of its users; this aids in providing an intuitive navigational framework. However, unlike a textbook, there is the ability for a user to tailor his experience to his own needs and learning style, as well as to obtain immediate feedback regarding his mastery of the subject matter. Additional details on the blocks and the way in which they adhere to the guiding design principles are provided below.



**Figure 1.** START system architecture

The introductory screen (Block I) describes the overall goals of the module to the user, provides a brief description of the module architecture so that the user knows what to expect when navigating through START, and defines materials that the learner may want to have on hand before starting (scratch paper, a calculator and a reference textbook). In keeping with the guiding principles, the introductory screen is straightforward with the minimum required amount of text. This screen is perhaps of critical importance to a successful module: if the introductory screen is too complex, users may choose to skip it, missing important information about the module, or perhaps dismiss the module in its entirety.

Forces on a particle are the simplest form of structural analysis: by definition, all forces act through the same point in space and no moments are introduced. These types of analytical problems are solved using only particle vector mechanics, i.e., Step 1 of Table 2, and START’s second block therefore covers this topic. In Block II’s introductory section, the concept of a vector is reviewed, and the principles and procedures for resolving the numerous external forces that may act on a body into a single vector are presented.

The third block of START introduces the mechanics of rigid bodies. Here, external forces no longer act through the same point; moments may therefore exist in the body, and all four steps of Table 2 are required for the problem's solution. Only beam-type bodies are considered in this block. Beams are chosen as they allow the basic method to be conveyed without unnecessary geometric complexity. In addition, beams are a commonly occurring engineering structure, and beam theory is often used for problems as diverse as a construction I-beams, an aircraft wing or a bridge deck; as such, most engineering students see beam problems repeatedly throughout their education. This block first introduces the concept of static equilibrium due to forces only. This is followed by the definition of a moment, a review of the calculation of the magnitude and direction of a moment due to an outside force, and the addition of the moment equilibrium equation. Standard boundary support conditions for beams and their meanings are also introduced. As in the other blocks, an example problem and a set of practice problems of increasing difficulty are then provided.

The fourth and final block introduces distributed loads as well as bodies of more complex shapes than the beams used in Block III. The local architecture of this block is identical to the previous two: an introduction, a description of the necessary analytical methods, an example problem, and practice problems of increasing difficulty.

As indicated by the lowest row of encapsulated text in Figure 1, the default path through START is also supplemented with pop-up reference screens. This material constitutes an unnecessary review for many advanced users, but inexperienced users may require supporting information. These windows, which “pop-up” from the main content screen when a user selects the corresponding navigation button, consist of screens that list the START objectives, describe the unit systems used in START (SI and US customary), include a table of standard boundary condition notations, provide a multimedia glossary of terms for structural analysis, and provide general instructions on module interface usage and providing feedback to the module administrator. Figure 2 provides a typical reference screen; it illustrates the forces that exist at two standard beam supports used within START.

## User Interface Design

Learners interface with the START module through a variety of buttons that control navigation, hints, and pop-up screens, as well as by typing numerical answers in response to practice problems. With respect to this latter mechanism, users are expected to perform numerical calculations outside the START executable; most students are proficient at performing calculations on their personal calculators, so there was no desire to force them to adhere to a new calculation method within START. Once answers are calculated, users type their results into a text box located immediately below the problem statement. The standard requested input for START is the magnitude of a single force, although a variety of other input options can be utilized with Authorware.

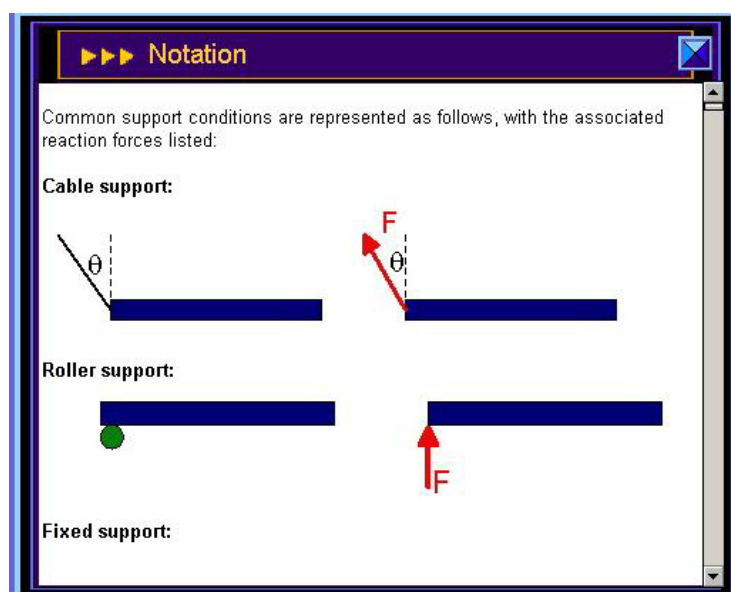


Figure 2. Structural boundary condition notation screen

The logic governing START's evaluation of user input is presented in Figure 3. This is perhaps best understood with the aid of Figure 4, which shows a typical problem presentation where the final solution has been displayed. When any problem is first presented, only the uppermost diagram, the problem statement below it, and "hint buttons" are displayed. This all represents the "Begin" ellipse and "Request Input" rectangle within Figure 3. At this point, users can enter a numerical answer, press a hint button, or press a navigational control button along the left hand panel. Upon receiving any input, the first determination made by START logical scripts is whether the type of input is a button press or if numerical data has been entered. In this latter case, answers are determined to be correct if they fall within a designer-specified accuracy. This is intended to account for errors produced by unit conversions and/or round-off during the solution phase, with a default of  $\pm 1\%$  of the correct answer. A random number generator is used to select a positive feedback phrase from five possible phrases for those answers that are found to be correct. The feedback phrase and the answer are then displayed, which allows a user to see how close his or her input came to the correct result.

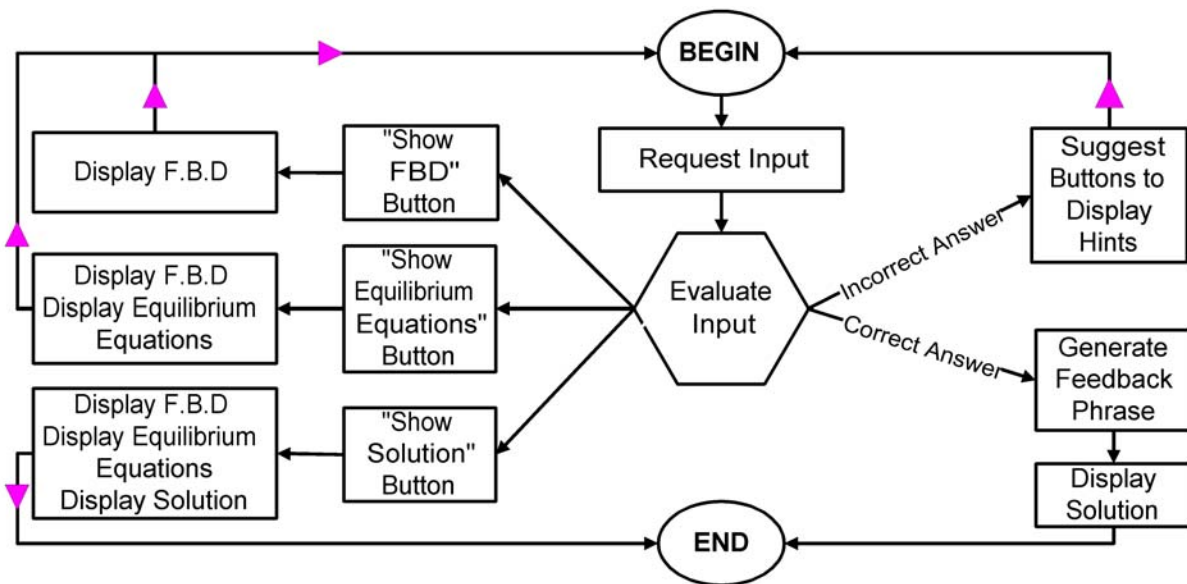


Figure 3. START text interaction logic

If the input is not within the acceptable range, the evaluation script generates the phrase "Not quite." The computer then suggests that the learner may want to use the answer hint system, which is presented as a series of hint buttons on the screen. Hint options include seeing a free body diagram, the equilibrium equations, or the final answer. That is, hints are arranged hierarchically according to the analytical method that is taught and which is displayed in Table 2. The hint buttons appear directly below the problem statement. If the "Show FBD" button is pressed, then the free body diagram below the hint buttons appears. If the "Show Equilibrium Equations" button is pressed, then these appear below the FBD. The "Show Solution" button will cause the solution at the bottom of the screen to be displayed. Figure 4 illustrates the problem presentation when the final level hint button ("Show Solution") has been pressed. Once this third tier of the answer hint system has been initiated, a user no longer has the option of entering answers into the input box. In Figure 4, one can see that the buttons and input box have been shaded grey, to indicate that they are no longer available. When the final answer has been presented, the user interactions on the screen are frozen to allow only navigational control. However, since the goal of this module is to build proficiency in a skill set, not to test user expertise, the user is under no requirement to use these hints, and may make an unlimited number of solution attempts until he or she obtains the correct answer or is no longer interested. A user may also navigate to the next problem regardless of whether a correct answer for the exercise has been accepted. This allows the requisite user control to approach the material in accordance with their own personal learning style, pre-existing knowledge, and learning goals.

Structural Analysis Review and Tutorial Page 2 of 2

### Equilibrium of Forces and Moments: SE-01: Box on a beam with rope supports

The 12 ft. beam above is supporting a 80 pound box, which is located 2.5 feet from the right side. At both the left and right sides, the beam is supported by ropes, as shown. The beam weighs 10 lb./ft. What is the maximum tension in the ropes?

Here is the free body diagram for this problem:

The sum of the axial forces and the moments about the left hand side of the beam are as follows:

$$\Sigma F_x = T1_x = 0$$

$$\Sigma F_y = T1_y + T2 - W1 - W2 = 0$$

$$\Sigma M_o = -W2 (6 \text{ ft.}) + -W1 (9.5 \text{ ft.}) + T2 (12 \text{ ft.}) = 0$$

The sum of the forces acting in the x-direction is zero, so  $T1_x = 0$ .  
 The weight of the box ( $W1$ ) was given as 80 lbs., so  $W1 = 80 \text{ lb.}$   
 The weight of the beam ( $W2$ ) acting at the midpoint of its length can be found by multiplying the length of the beam by the weight per unit length.  $W2 = 10 \text{ lb./ft.} (12 \text{ ft.}) = 120 \text{ lb.}$   
 Using the moment equilibrium equation, the known values of  $W1$  and  $W2$  can be used to determine that  $T2 = 1\text{b.}$   
 This value of  $T2$  can then be substituted back into the y-direction equation, along with the known values of  $W1$  and  $W2$ , to determine that  $T1_y = 1\text{b.}$

Figure 4. Typical START problem presentation

In addition to accessing hints, learners use a variety of other buttons to navigate and access content. This makes the experience inherently interactive; to produce an intuitive interface (Table 1), it is common to associate screen icons with physical objects that are familiar to users. For example, most word processor users recognize that the floppy disk icon will save their document and the icon depicting a printer will print it. The task of associating icons that are intuitively obvious to users becomes more difficult, however, for tasks that are more mental than physical. The START interface icons were chosen to be as intuitive as possible, but for some tasks there was no choice that would readily be apparent to all users. Thus, all icons are also labeled so that users are not required to memorize which icon depicts each pop-up feature. All interface icons appear in the left side navigation panel of each screen. As shown in Figure 4, readily associable objects are used for certain tasks, such as a magnifying glass to represent the “Find” option, and book-shaped icons and geometric designs are used for other tasks for which no obvious physical objects could be associated. Referring to Figure 4, the Objectives icon brings a pop-up window to the user’s screen which contains a brief list of the module objectives. The Menu icon brings up a hyperlinked hierarchical menu window, through which the user can navigate to any screen of the START module. The availability of the menu screen balances learner control with the program control provided by the default path. When learner and program control are balanced, the user is neither a passive recipient of the content, nor required to make complex decisions about the order of presentation of material he or she has not yet mastered. As described in the section on guiding principles, the combination of a default path with a high degree of learner control is essential to maintaining user interest. Also available is the Formulae icon, which initiates a pop-up window of common engineering formulas that might aid the user in completing the module. If the user presses the Glossary button, an interactive hyper-linked glossary is available in a pop-up window. Similarly, pressing the Find icon initiates an interactive search feature for the module. Finally, the Bookmark icon, when pressed, brings up a virtual bookmark system in which the user can mark screens and navigate to previously marked screens.



## **Presentation Design**

The final step in module development consisted of presentation design, which included the choice of the screen template used for the relative size and position of all the screen components, screen aesthetics such as font and resolution, and the multimedia components integrated into the module. All of the presentation design steps were completed in light of the principles presented in Table 1. For example, in order to give the impression that the various links cause new content to be brought into the current page, every screen of the module displays identical anchor components. With reference to Figure 4, these anchors include the panels containing the left side icons, the START name and block name on top, and the page number. Equally important to achieving this perception, the content area is always the same size, positioned as shown in Figure 4 in relation to the anchor components. Each screen devotes precisely the same area to each logical portion of the presentation, from the problem statement and diagram, to the textual input box, to the space for displaying the various hint options.

The color palette chosen for START is relatively small and was chosen for accurate reproducibility within the widest possible variety of computer color display settings. The screen-to-screen color selection is consistent, allowing users to instantly recognize the components of each exercise. Yellow and red texts are used against the dark background provided by the anchor panels. Content is presented in dark colors, ranging from black to red, with a white background area for maximum contrast. Different font colors are used to differentiate the various textual components of START, such as problem descriptions, numerical equations, and problem answers. For all drawings, dark blue is used for the body under consideration, red for the arrows indicating external forces, and black for dimensions and other geometric notations. There is also an intentional minimal use of multimedia. This keeps the presentation uncluttered and streamlined, avoids distracting users from the technical content, and reduces the overall storage size of the module.

## **Implementation and Assessment**

Initial feedback from selected users of the START module has been quite positive. After the module is introduced to engineering students, instructors with a knowledge of the Authorware program will be able to further customize and increase the number of problems presented in the START module. The longer the lifetime of the software, the more extensive the problem database will be. However, prior to a more widespread introduction, we wish to complete the development of the module's assessment and evaluation tools and associated metrics, as well as to determine and design into the module the way in which tracking data is to be used as part of the assessment process. That is, in addition to solicited and unsolicited user feedback, the module can record such information as the frequency and duration of logins, screens viewed, student responses, time spent per exercise, or the number of answer attempts made per exercise. The challenge is to combine all of the information into a quantitative measure of the module's effectiveness to establish, for example, whether (1) students access, and report favorably on, the resource, (2) students report that the resource is enriching, and (3) educators report that the resource promotes a higher level of learning (Soon et al. 2001). We are currently working to develop an initial version of this, and plan to shortly introduce the module in a limited fashion to assess both the module and the evaluation tools.

## **Conclusions**

A number of guiding principles were extracted from the relevant literature and used to develop what are believed to represent a set of best practices for developing CBI that will generate positive learning outcomes. START, the computer-based instruction module designed using this approach, achieves the design goals and provides a clear, simple, and straightforward user interface, imparts the appropriate balance between learner and program control, and offers a high level of interactivity. In addition to developing assessment and evaluation tools, future work on this module will include expansion of the content to more advanced structural analysis techniques. It is envisioned that this process will both improve the specific content and efficacy of this module, as well as help to develop generic approaches for the development of highly effective computer-based instruction modules in a variety of problem solving applications.

## Acknowledgements

The authors gratefully acknowledge the support of NASA Langley Research Center through Cooperative Agreement No. NCC-1-01004. We would also like to thank Elizabeth Kahl for her contribution of authoring software expertise.

## References

- Beer, F.P. & Johnston, E.R. (1997). *Vector Mechanics for Engineers: Statics and Dynamics*, 6th ed. Boston, MA: WCB McGraw-Hill.
- Croft, A.C., Danson, M., Dawson, B.R. & Ward, J.P. (2001). Experiences of using computer assisted assessment in engineering mathematics. *Computers & Education*, 3, 53-66.
- Edwards, D.W., Lamb, F.M., Ahmed, V.S. & Rothberg, S.J. (2001). ASTutE: Computer-aided teaching of materials balancing. *Chemical Engineering Education*, 34 (3), 258-263.
- Hannafin, M.J. (1984). Guidelines for using locus of instructional control in the design of computer-aided instruction. *Journal of Instructional Development*, 7 (3), 6-10.
- Holzer, S.M. & Andruet, R.H. (2000). Experiential learning in mechanics with multimedia. SUCCEED. Retrieved 19 December 2001, from <http://www.succeednow.org/papers/Expmechmult.pdf>.
- Hummer, J.E., Baugh, J.W., Konuru, B.P. & Click, S.M. (1996). A controlled comparison of traditional classroom instruction with computer-based instruction in an engineering class. American Society for Engineering Education Annual Conference & Exposition, 1996, American Society for Engineering Education, Washington, D.C.
- Kristof, R. & Satran, A. (1995). *Interactivity by Design: Creating and Communicating with New Media*. Mountain View, CA: Adobe Press.
- Lin, B. & Hsieh, C.-T. (2001). Web-based teaching and learner control: a research review. *Computers & Education*, 37, 377-386.
- Macromedia Authorware. <http://www.macromedia.com/software/authorware/>
- Negahban, M. (2000). Results of implementing a computer-based mechanics readiness program in statics. *International Journal of Engineering Education*, 16 (5), 408-416.
- Rais-Rohani, M. (2001). On development, application, and effectiveness of a computer-based tutorial in engineering mechanics (statics). American Society for Engineering Education Annual Conference & Exposition, 2001, American Society for Engineering Education, Washington, D.C.
- Rojiani, K.B., Kim, Y.Y., Kapania, R.K. (2000). Web-Based Java Applets for Teaching Engineering Mechanics. American Society for Engineering Education Annual Conference & Exposition, June 2000, American Society for Engineering Education, St. Louis, MO.
- Schank, R.C. (1993). Learning via multimedia computers. *Technology in Education*, 36 (5), 54-56.
- Soon, M.H.B., Chan, H.C., Chua, B.C. & Loh, K.F. (2001). Critical success factors for on-line course resources. *Computers & Education*, 36, 101-120.
- Tufte, E.R. (1990). *Envisioning Information*. Cheshire, CT: Graphic Press.