3 The Approach of Interactive Computer-Aided Design and Web-Based Structural Steel Design

3.1 PERSONAL COMPUTERS

Personal computers have gained substantial popularity among design engineers. In addition to low cost, personal computers provide a user-friendly programming environment, lend themselves to interactive programming very effectively and can often be expanded due to their open architecture.

3.2 PRINCIPLES OF STRUCTURED PROGRAMMING

Structured programming is a method of designing computer programs to minimize complexity (Kittner and Northcutt, 1984). Principles of structured programming are delineated in this section. Generally speaking, computer programs presented in this book have been developed with these principles in mind. Of course, these rules cannot always be followed rigidly, especially when efficiency or speed of computation is the major consideration.

1. The program should be “egoless”, that is, easy to read and understand. Due to high manpower cost, clarity and simplicity should be valued more than complicated efficiency.
   a. Choose variable names close to standard and customary notations.
   b. Insert spaces within the program statements for better readability.
   c. Use comment statements generously throughout the program for internal documentation.
   d. Use blank statements to skip lines between blocks of the program.
   e. Indent statements within a loop.
   f. Define the variables within the program.
   g. Use one statement per line.
2. The program should be easy to modify and maintain.
   a. Use variables for constant values that may change in the future. Place the constant values at the beginning of the program.
   b. Initialize all the variables at the beginning of the program.
3. The program should be designed to maximize programmer efficiency (Kittner and Northcutt, 1984).
   a. Use top-down programming; that is, design he program in stages from simple to complex.
   b. Use a flow chart in designing the logic of the program.
   c. Avoid GOTO statements as much as possible in procedural languages such as FORTRAN and BASIC. This will prevent unnecessary complexity (the so-called “spaghetti” effect) and make the program logic easier to follow.
   d. Use modular programming. This approach will help to use fewer GOTO statements.

   e. The relationship between modules (coupling) should be weak. Each module should be as independent of the others as possible.
   f. A module should preferably have a single entry and single exit.
   g. A module should not be big in size. A good size is a singe page of coded program.
4. The program should be reliable.
   a. State the limitation of the program.
   b. Incorporate as many error messages as possible
   c. Use security to prevent modification of the program by unauthorized users.

3.3 HOW TO WRITE EFFICIENT PROGRAMS

A list of guidelines for developing efficient software on personal computers is presented in this section. Some of these guidelines are in contradiction with the principles of structured programming presented in section 3.2. As the programs become larger, one may have to compromise some of the principles for the sake of increased efficiency.

1. On most available personal computers, division and multiplication take 10 to 100 times more CPU time than addition and subtraction. Therefore, unnecessary divisions and multiplications should be avoided.
2. On personal computers integer arithmetic is an order of magnitude faster than floating-point arithmetic. To increase the efficiency of the program, especially in graphics software, integer arithmetic should be preferred over the floating-point arithmetic.
3. In loops and iterations, the quantities that do not vary in the loop should be calculated outside the loop.

4. Personal computers can perform a decrement operation more quickly than a comparison.

3.4 INTERACTIVE COMPUTER-AIDED DESIGN

As of mid 1980's computer programs had not been used extensively for design of structures. A number of reasons were cited by Adeli and Paek (1986) for the lack of interest in conventional programs for design of structures:

1. In practical design cases there are a large number of alternatives whose selection needs the judgment of the experienced human designer.
2. Design specifications usually cover the general situations and leave the less frequent cases to the judgment of the human designer. In other words, they contain discontinuities and gaps to be filled by the human designer.
3. Human designers use their previously gained experience in design of new structures.
4. A human designer usually visualizes and sketches different structural forms and configurations before making the preliminary design, stress analysis, and the final design.
5. Design specifications change frequently – for example, every three to four years – even though not significantly.
6. Design specifications are based on years of experience gained by researchers and practicing engineers and contain rules of thumb and heuristics which may not be readily implemented in traditional computer languages.
7. Parts of design specifications and standards need interpretation by an experienced designer.

8. Design is an ill-defined and ill-structured problem, lacking a clearly defined goal, and not quite amenable to algorithmic procedures.
9. Design is a creative process.

Adeli (1987, 1988) developed the approach of interactive computer-aided design of steel structures. An interactive program should ideally perform the following:

1. Carry all the numerical calculations.
2. Check for the consistency of design according to the specified design specification.
3. Perform error checks.
4. Present possible alternatives to the user.
5. Prevent the user from entering the data in violation of the design specification.
6. Inform the user about the ranges of practical values.
7. Provide practical values for the final design.

3.5 WEB-BASED EDUCATION

Distance and web-based education has received increasing attention in recent years in Europe, specially United Kingdom, North America, and East Asia. Distance education potentially provides opportunities

- to educate students in remote areas without ready access to academic institutions,
- to offer continuing education courses to company employees at their sites without them having to leave the company and lose productivity due to commuting and work interruption, and
- to supplement the conventional classroom teaching.

Until recently distance education was based on exploiting the communication media such as the closed-circuit television, videotaped lectures, and audio-conferencing with extremely limited interaction between students and the instructor and peer students. The recent resurgence of the World Wide Web (WWW or the web for short) and the Internet technologies has provided fresh opportunities for effective distance learning and education. Compared with the traditional distance learning approaches of 1970's and 1980's the Internet provides two significant advantages. First, it provides a powerful and fast mechanism for interaction between the instructor and students. Second, students can have easy and instant access to a wealth of information through numerous on-line libraries and web sites.

The web can be used three different ways as an educational tool (Barrie and Presti, 1996). The first use of the web is to simply access information on the web itself such as the electronic academic journals available on the web. For example, the Electronic Journal Center of OhioLink (http://journals.ohiolink.edu/etext) provides a web-based service that delivers the contents of scientific and technical journals to end users via a web browser.

The second way is offering a course or a complete degree program on the web. For example, Open University in the United Kingdom (http://www.open.ac.uk) and University On-Line (http://www.online.edu) and CALCampus (http://calcampus.com) in the U.S. are providing regular university-level courses through the web. Lectures are recorded on video and then converted into special graphic files so that they can be replayed through a web browser. Current state of technology is still arguably too limited in terms of speed of transmission and bandwidth to make the total replacement of the conventional classroom viable.

The third and probably at this time the most attractive way of utilizing the web is to use it as an aid to the conventional classroom instructions. There are already a number of examples in this category such as

- http://sulcus.berkeley.edu/mcb/165_001: Developed at the University of California at Berkeley this project provides supplementary materials for the Molecular Neurobiology and Neurochemistry course.
- http://philae.sas.upenn.edu: Developed at the University of Pennsylvania this project provides a Virtual Media Language Lab on the web helping students to learn foreign languages.
- http://vizlab.beckman.uiuc.edu/chickscope: Developed at the University of Illinois at Urbana this project teaches K-12 students virtual experimentation by remote instruments such as using Magnetic Resonance Imaging (MRI) to demonstrate egg-to-chick embryology (Bruce et al., 1997).
- http://www.med.umn.edu/ISAP: Developed at the University of Minnesota this project provides self-assessment in Pharmacology (Konstan et al., 1997) .
- http://www.eerc.berkeley.edu/bertero/index.html: Developed at University of California at Berkeley, this site provides slides with commentary on damage due to earthquakes that can be used to complement a course on earthquake engineering.

It is believed that distance education on the web cannot fully replace the live face-to-face classroom instructions for design courses at least for the foreseeable future. But, it can be used as an effective tool to supplement classroom instructions. In that context, web-based education provides the following advantages and opportunities:

- Richer learning environment through easy access to a very large and growing number of on-line resources,
- Access to more up-to-date course materials,
- Flexible learning environment (students have more flexibility in scheduling their study time and can spend more time on the subjects they consider difficult),
- More opportunities for students with special needs such as physical impairment or language problem (for example, for students not quite fluent in English)

It must be noted that the web-based education is still limited in providing interactivity and communication.

3.6 ISSUES IN DEVELOPING COURSEWARE FOR WEB-BASED EDUCATION

Developing a web site to teach an engineering design course requires dealing with a number of problems and technological challenges. These problems along with potential solutions are discussed in this section.

3.6.1 Information representation

Information can be provided in the form of a) written texts, b) pictures, c) sound, and d) animation with increasing difficulty as we go from a to d. Most web-based materials today are in text and picture forms. But, the latter two are finding increasing applications. The course web page may include an on-line syllabus, course materials developed by the instructor, frequently-asked questions, and hyperlinks to other relevant and useful sites.

3.6.2 Human-Machine Interaction

Human-machine interaction can be through keyboard, mouse, or voice-recognition (Adeli, 1990). At this time, voice-recognition is still a nascent technology for interaction with the web in developing educational tools. Thus, the human-machine interaction has to be primarily through keyboard and mouse. An important issue is compatibility with various hardware and operating systems; that is, software developed on a Windows operating system may not be executed on a Macintosh or Unix operating system.

The web overcomes the compatibility problem by using a platform-independent or architecture-neutral protocol such as HyperText Markup Language (HTML) in cooperation with a web browser. All web pages are written in HTML. The word *HyperText* means the document has links to other local documents or web pages on the Internet (hyperlinks). HTML, however, provides no programming capability and can produce only static information like the materials found in a printed newspaper, thus providing only limited user interactivity. This limitation can be overcome by using Internet languages such as Java (Horstmann and Cornell, 1999), JavaScript , ActiveX , and DHTML (Homer et al., 1997) that allow dynamic creation of texts and images. Among these languages, Java appears to be the most powerful and consequently is the most widely-used. The web-based courseware in this work has been developed in Java. The unique features of Java will be discussed in the next section.

3.6.3. Communication and Interaction between Students and Instructors

Students can communicate with the instructor through asynchronous communications such as email and electronic bulletin boards and synchronous communications such as a *chat room*. The communication among the students themselves is considered as important as communication between the students and the instructor as students are encouraged to consult and collaborate with each other. Sometimes a student may grasp a difficult concept better from her peers than from her instructor because her way of thinking and reasoning is more similar to her peers than her instructor. As such, academic interaction among students is considered an important element of the learning process. Students can communicate with each other through email, electronic boards, and chat rooms.

Another way of communication and interaction by students is through *anonymous peer review*. Each student places her solved homework on the web anonymously which is then reviewed by a few other randomly selected students.

3.6.4 Presenting Mathematical Expressions

Engineering design is often based on commonly-used design codes and specifications which include a) mathematical expressions, b) texts, and c) figures. For a course on structural steel design, mathematical equations and symbols representing design constraints and structural analysis must be presented in a clear way. This creates challenges for the course web site designer. HTML has a very limited ability to display characters and equations on the screen. For example, displaying subscripted and superscripted variables is a very tedious editing job. Several solutions are discussed in this section.

One solution is to present all the mathematical expressions as figures since most web browsers can display figures without any special effort by the users. In this case, equations or whole documents containing equations can be converted to graphic files (JPG or GIF files) with a special tag identifying it as a figure to the web browser. This solution is a very time-consuming task for the web site designer. Further, this causes an undesirable increase in the size of the file slowing down its transfer from the server to the user. As a result the user may have to wait many minutes for the file to be downloaded. This can be particularly irritating for users connected to the web via a telephone modem. To minimize the web site development effort, web designers often use special software tools that can accommodate mathematical terms. For example, word processors such as Microsoft Word and Word Perfect provide a function for automatic conversion of mathematical expressions to figures.

The most favored software tool for displaying mathematical expressions among the web site designers is probably the Adobe Acrobat (http://www.adobe.com) using its PDF format for three reasons. First, Acrobat Reader can be downloaded free of charge. Second, its PDF format allows for a higher resolution than JPG and GIF formats. Third, common word files as well as hand-written notes can be converted to PDF files easily. Unlike an ordinary HTML file with equations included as figures, however, users cannot edit a PDF formatted file. They can only view and print it. The size of a PDF file is much larger than that of a corresponding HTML file and consequently it requires more downloading time, which can be an inconvenience for a user connected to the web via a telephone modem. Thus, it is common to provide PDF formatted files along with plain HTML (JPG and GIF files) to students so that they can have a choice depending on their web connection status.

More recently, a software package called Scientific Notebook (http://www.scinotebook.com) has received a lot of attention. It is similar to the popular mathematical software packages such as Maple, Mathcad, and Matlab for solution of mathematical problems such as partial differential equations. However, unlike those mathematical software packages, the user can browse the Internet using freely available Viewer Version of Scientific Notebook without using any web browser. Moreover, unlike the Adobe Acrobat, it allows the user to create and edit documents. Since the Scientific Notebook uses a unique file format, other popular web browsers such as Netscape or Microsoft Internet Explorer, however, cannot load the web pages written using this software

3.6.5 Homework Assignment on the Web

Design concepts are usually mastered through assignment of homework problems. How well the students perform in the homework assignments is used by the instructor to monitor the progress of students. There is no substitute for the traditional paper and pencil homework on the web. Most current web-based courses assign homework in the form of multiple-choice questions, questions with short (one line or so) answers, or essays. Such assignments often will not do the job for an engineering design course requiring manipulation of mathematical procedures, data, and physical drawings. Students cannot be expected to type in all of the traditional paper and pencil homework.

As such, one has to compromise between the amount of details the student has to include in her homework and the time she needs to enter in her solution. In the conventional homework assignments, students are normally expected to include all the details of their solutions line by line. In a web-based homework assignment, instead the instructor can ask the students to provide the results for a specified number of intermediate steps. This means the instructor has to develop a suitable set of homework for the web-based course.

3.6.6 Interactivity for Improving the Learning Process

Some students tend to print all of the web-based course materials at the beginning of the semester/quarter and return to the web site only occasionally. Strategies are needed to animate an idle site. One solution is scheduling group chats periodically. Participation in the chat can be part of homework assignment and grading system of the course. Another strategy would be having a high level of interactivity (including user-friendliness and on-line explanation) in both the web page and different design programs placed on the web. This will encourage usage by the students.

3.7 UNIQUE FEATURES OF JAVA

Java is a special programming language created to expand the user-machine interactivity on the Internet. It allows the users, for example, to play games, run spreadsheets, chat with other users in real time, and obtain continuously updated data on the Internet. The most unique feature of Java compared with other languages is that it allows the web site designer to write special programs, dubbed *applet*. A Java applet can be safely downloaded by the users using a web browser. The user needs only to have access to a web browser that can interpret and execute Java applets. Java compiler produces a platform-independent

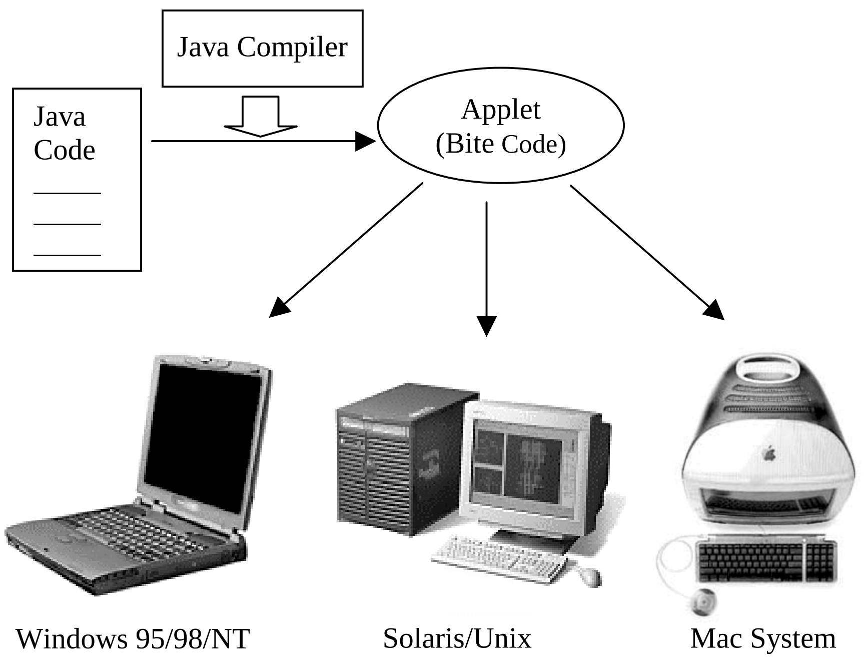

Figure 3.1 Process to develop an applet in Java for use on the web

bite code that can be interpreted and executed on the browser independent of any user's machine or its operating system.

An important consequence of the platform-independence of Java is effective distribution of executable programs (Figure 3.1). A Java programmer does not need to worry about writing various versions of the same program for different machines and operating systems. Users simply download an applet program whenever they need it rather than installing the program on their computer. As such, maintaining, updating, and redistributing the program is done centrally and effectively.

Java applets are executed on the web browser; they are not stored on the hard disk of the computer. Thus, the users do not need to worry about any potential loss of data or damage to the hard disk due to, for example, a virus. As an added safety feature, a Java applet, by default, is not allowed to delete or modify any file on the user's hard disk or perform other computer crashing operations.

Java is an Object-Oriented Programming (OOP) language with all its desirable characteristics of abstraction, encapsulation, and inheritance (Yu and Adeli, 1991, Adeli and Kao, 1996). Java draws upon features of OOP languages Eiffel, SmallTalk, and C++. Like other OOP languages, C++ is designed to support its predecessor language, C, which is not an OOP language. This adds complexity to its syntax and memory management. In contrast, Java does not provide such back supporting, thus freeing itself of unnecessary complexities.

Java also provides multi-threading. A single Java applet can execute different processes simultaneously, for example loading an image while performing numerical processing. Like symbolic processing languages such as LISP (Adeli and Paek, 1986) Java performs automatic garbage collection, that is the memory no longer in use is released automatically and continuously in the background. This relieves a Java programmer from worrying about memory allocation and de-allocation, a major source of bugs in computer programs.

3.8 CREATING INTERACTIVE STRUCTURAL STEEL DESIGN COURSEWARE ON THE WEB

In a pioneering work in the area of Computer-Aided Instruction (CAI) of structural design, Adeli (1987 & 1988) developed the approach of interactive computer-aided design (CAD) of steel structures. In this approach, the user/designer and the CAD system solve the design problem in a collaborative manner. The CAD system performs all the numerical computations and displays the designs. It stops after each step of the design, presents the intermediate design results and alternatives, and requests the user's input on the key design questions and parameters as the design progresses. In this synergistic man-machine approach, the CAD system does all the work in consultation with the user; but the user is in charge of the whole design process.

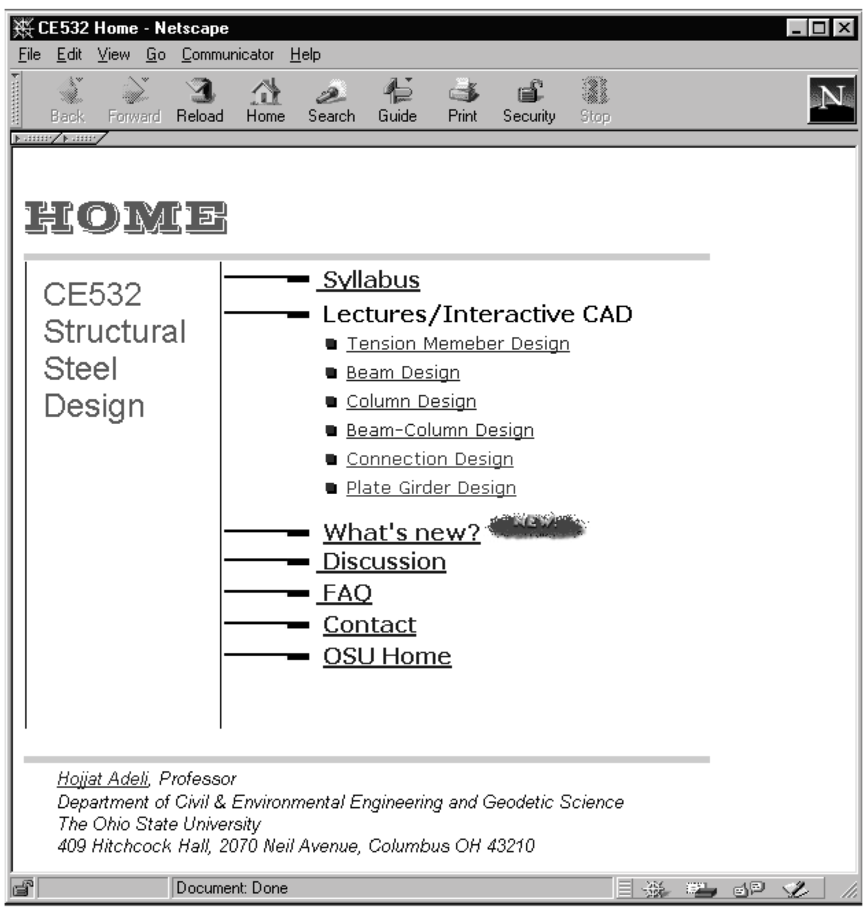

Figure 3.2 The homepage for the web-based steel design courseware

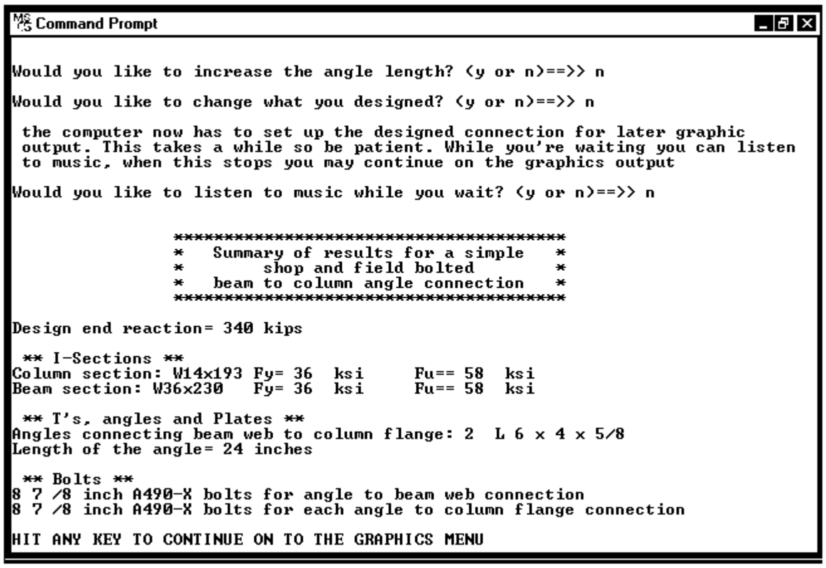

Figure 3.3 Text-based user interface (Adeli, 1988)

In this book, we build upon that approach by utilizing the new features provided by the web and platform-independent programming language Java (Adeli and Kim, 2000).

Figure 3.2 shows the homepage for the web-based steel design courseware. At this time, the subjects covered in this course are those normally covered in a first course on structural steel design and covered in Adeli (1988). They include design of tension members, beams, columns, beam-columns, connections, and plate girders. Since the size of the file is the key factor for the downloading time, and the goal is to reduce this time, one applet is created for each type of design.

In the earlier interactive design approach (Adeli, 1988) the user interaction was through a text-based user interface (Figure 3.3). The interaction in the new web-based approach is through a Graphical User

Interface (GUI) using Java applets. The result is a substantial improvement in the user interface in terms of degree of interactivity, user-friendly input procedures, and on-line help.