Int. J. Engng Ed. Vol. 13, No. 6, p. 397–406, 1997
Printed in Great Britain.

0949-149X/91 $3.00+0.00

Using Java to Develop Interactive Learning Material for the World-Wide Web*

BENJAMIN 'QUINCY' CABELL V, JOSEPH J. RENCIS and HARTLEY T. GRANDIN Jr
Mechanical Engineering Department, Worcester Polytechnic Institute, Worcester, MA 01609-2280, USA
JAVED ALAM
Department of Civil and Environmental Engineering, Youngstown State University, Youngstown, OH 44555, USA

Java has emerged as a powerful programming language for developing platform-independent, interactive and computational-based software that can be used on the World-Wide Web through a Java-enabled Web browser. The paper introduces the Java programming language, its advantages and disadvantages, and its characteristics for developing interactive instructional applications on the WWW. The interactive and computational capabilities of Java are demonstrated through a simple matrix assembly Java applet. With this applet, students assemble element equations into the global equations for structural analysis using the bar element. The matrix assembly applet features a graphical user-friendly interface, on-line help and interactive feedback.

INTRODUCTION

THE DEVELOPMENT of the Internet, a network of networks, allows the interconnection of computers separated by tens of inches or thousands of miles. The number of users and Internet hosts are growing at an exponential rate and there is no reason to expect this rate will do anything but continue to increase. The Internet offers a great potential for the rapid and cost-effective exchange of massive amounts of information. The World-Wide Web (WWW, W3, or the Web [1, 2]) is that part of the Internet which exists and is included in Hypertext Markup Language (HTML) and its method of exchange, the Hypertext Transport Protocol (HTTP). The WWW has allowed the finite element method universal resource (FEMur) development team to create and disseminate information in hypermedia format, a combination of hypertext [3, 4] and multimedia. HTML allows for the combination of diverse information, including plain text, pictures, sound clips and animation/video clips. These hypermedia documents are placed on an Internet-connected computer running a WWW server, which allows the documents to be accessed easily and instantly by anyone with a computer, an Internet connection, and a WWW browser (such as Netscape/Mosaic/Lynx/Internet Explorer/Hot Java [5–9]).

A significant part of the teaching process involves the delivery of course-related information to students. Recent developments in computer networking technologies offers new ways to perform this function. There have been several attempts to use the development of the WWW for engineering instructional purposes. These efforts are directed towards creating course material in multimedia format using HTML [10–14] and making it accessible world-wide through the Internet. Thus far, the student's role in this learning process has been largely passive; the student navigates through the information on the Web pages by clicking on hypertext links, making an education on the Web no more interactive than the reading of a textbook. This use of WWW for teaching and learning lacks the interactive component essential for the learning process.

The HTML language is continuously evolving to incorporate new features needed to produce multimedia documents. This constant evolution makes it impossible for the makers of Web browsers to incorporate data handlers for all available WWW content. Web browsers use helper applications, third-party software applications the user can configure to work with his/her Web browser, to process/handle unrecognized Web data (such as Microsoft Word documents, PostScript files, Shockwave multimedia files, RealAudio audio files, and MPEG video).

The introduction of the Java programming language [15] extends the capability of HTML documents by allowing developers to integrate platform (computer hardware and operating system) and browser-independent features into Web documents that were formerly available only through user-installed platform-dependent helper applications. Java provides programmers with the freedom to create interactive content for the Web by developing new data types and the methods to operate on them. When a Java-enabled browser (e.g., Netscape Navigator version 2.0 or higher or Microsoft Internet Explorer version 3.0 or higher) encounters the HTML Java applet tag, the browser

* Accepted 1 June 1997.

requests the necessary Java code from the Web server on which the code resides. The Java code, passed around the WWW in compiled form as a 'class', encapsulates data and the functions necessary to manipulate that data.

The beauty of Java is that there are no specific limitations on what the Java language can be used to do. For security reasons, however, there are restrictions on how Java code can access the computer on which it is run (in this way the user is protected from misguided and/or dangerously unskilled programmers).

Java applets can display graphics (still or animated), play audio files, and receive user input from the mouse and keyboard. Web documents come alive because the computer can respond instantly to a user's input; no longer is the user forced to supply the computer with responses via fill-out forms submitted through common gateway interface (CGI) scripts/executables. The computer can now become an active participant in and computational tool for the information dissemination process rather than a clumsy CRT 'book'.

The authors are currently using Java to develop a prototype interactive learning tool for the one-dimensional bar element. The interactive learning tool for the finite element method is called FEMur-CAL (Finite Element Method universal resource for Computer-Assisted Learning). This FEM tool does not replace the conventional classroom experiences, but provides supplementary instruction to students who need extra help. The prototype will be integrated into the 'Learn the Finite Element Method' component of the Finite Element Method universal resource (FEMur). FEMur is a World-Wide Web site maintained by the authors of this paper and its URL (Uniform Resource Location) is http://femur.wpi.edu/.

OVERVIEW OF JAVA

Java is both a program environment and a programming language developed and maintained by JavaSoft, the Sun Microsystems company responsible for Java [15]. Java has been tailored specifically for networked computing, such as that which takes place on the Internet. Languages such as C++, Pascal, and FORTRAN require extensive libraries and experienced programmers to achieve the same degree of networked functionality. Any program that is written in Java can run on any computer as long as the 'Java Virtual Machine' is inside. The Java Virtual Machine (JVM) [16] is a programming standard for an imaginary machine. The standard reflects the basic capability of all computer platforms (all CPUs as well as all windowed operating systems). When writing code in Java, a programmer is not writing software for a particular hardware/software platform but instead for the JVM. Any computer can execute Java code by becoming (through software) this Java Virtual Machine.

The Java programming language allows the developer to build both applets and standalone applications. Applets are pieces of compiled code that are referenced within Web pages (HTML documents) on the WWW. An applet requires a Java-enabled browser such as Netscape Navigator 2.0 or higher [5] or Internet Explorer 3.0 or higher [8] to run. An example of an applet is the matrix assembly program in Section 5. Standalone programs are written in almost the same way as applets, but do not require, nor do they use, a Web browser. Almost all computer programs in common use are examples of standalone applications, including Microsoft Word, CorelDRAW!, and Lotus Notes.

The Java source code is written in the same way that code for any other language is written. Java is an object-oriented programming (OOP) language. The roots of Java syntax lie in C++, but it should be stressed that Java is not merely a refinement of C++; Java is at its core quite different from C++. The Java programmer's job is being made easier as new and powerful tools, libraries, and integrated development environments (IDE) are becoming available for every platform. Java source code is compiled using a Java compiler, which creates platform-independent Java bytecode. This bytecode is unlike normal machine code in that it is written for the JVM and is not, therefore, native to any particular processor or operating system. Web pages that contain Java applets include the applets through the use of the HTML tag ⟨APPLET⟩ (an extension to HTML originally proposed by JavaSoft) [16]. This HTML tag refers the Web browser to the location of the applet's code. A Java-enabled Web browser (which is platform-dependent software) reacts to an ⟨APPLET⟩ tag by downloading the Java class that is referenced in the tag. If the specified class requires code that is contained in one or more additional classes (which generally is the case), these supporting classes are then retrieved from the server. As each class is retrieved it is examined and any additionally required supporting classes are retrieved. This process continues until the browser has downloaded all the classes that have been referenced within any of the already downloaded code. Once the classes are downloaded, the browser executes the code through a Java interpreter (which is the software that emulates the JVM).

Java is often referred to as an 'interpreted and compiled' language. There is a performance penalty resulting from the required code interpretation, which has been reduced by the recent incorporation of just-in-time (JIT) compilers into Web browsers. JIT compilers transform the platform-independent classes into native machine code for improved run-time performance. The only Web browsers that currently support Java are Netscape Navigator 2.0 or higher [5], Microsoft Internet Explorer 3.0 or higher [8], and Sun's Hot Java 2.0 or higher [9].

Version 1.0 of the Java Development Kit (JDK)

[17] appeared in mid-January 1996. At the time of this writing, the current version of Java is 1.1 Beta 3. Versions of the JDK are available from Sun for SPARC/Solaris, X86 Solaris, Microsoft Windows NT/95, and MacOS. Third-party versions of the 1.0.2 JDK have been made available for Windows 3.11, IBM OS/2, Linux, and other UNIX platforms. The JDK as supplied by Sun includes: Java classes, source, compiler, interpreter, appletviewer, debugger, documentation generator, and application programming interface (API) documents. The most complete archive of Java applets, tutorials, and links to sites is Gamelan [18]. Official information and links to various Java applets can be found at http://www.JavaSoft. com/applets/index.html. Current USENET newsgroups include: comp.lang.Java (the original and still most trafficked Java newsgroup), comp.lang. Java.advocacy, comp.lang.Java.announce, comp. lang.Java.setup, comp.lang.Java.programmer, comp.lang.Java.security, comp.lang.Java.tech, and comp.lang.Java.misc. The documentation distributed with the JDK is very useful, but not comprehensive. Additional material must be read and understood before anyone can be proficient in Java; this material can be found in Java newsgroups, Java books [19, 20], and the many excellent tutorials [21] on the Web.

Java is not the only programming option open to Web developers wishing to bring interactive and dynamic documents to the WWW. On the heels of Java's success other competing and co-operative technologies have been developed. Included in these alternative technologies are: JavaScript [22], JavaBeans [23], OpenDoc [24, 25], and ActiveX [26].

JavaScript is Netscape's Java-like platform-independent, object-based scripting language. JavaScript is particularly useful for those with limited programming experience as it uses a simplified syntax and specialized built-in functionality to provide a wide variety of form verification and interaction options.

The technologies of JavaBeans, OpenDoc, and ActiveX were all created using the increasingly popular component model. This new approach 'is based on the concept of component software: self-contained, reusable software modules. You might think of component software in the same way you think of components for a home stereo. When you create a home stereo system, you can buy different companies' stereo parts—a cassette deck, receiver, CD player, or speakers—because you know that they all have a standard interface and will work together. Components in a [component model] software application—for example, a text editor component, database component, online network component, or multimedia service component—all share a standard interface, and will work together ...' [27].

JavaBeans is JavaSoft's portable, platform-independent Java-coded component model technology. JavaBeans components will run in ActiveX, in OpenDoc, and in Netscape. OpenDoc is the platform-independent component model technology developed by Apple, IBM, Novell, SunSoft, Taligent, and the XSoft Division of Xerox Corporation. 'OpenDoc offers a document-centric (as opposed to application-centric) model for computing' [28]. ActiveX is Microsoft's platform-semi-independent component model technology. ActiveX includes support for Java. ActiveX controls can be developed through programming tools such as Visual Basic, Visual C++, Visual J++, Borland C++, and Borland Delphi or through desktop applications including Microsoft Word, and PowerPoint. ActiveX does not yet approach the platform-independence of Java, though Microsoft claims this is ActiveX's goal.

ADVANTAGES AND DISADVANTAGES OF USING JAVA

The authors feel that Java has the following advantages for the development of software in an academic environment:

- *Tailored for the Internet.* The Java language is unique in its built-in handling of the network protocols required to send and receive data across networks such as the Internet. For the programmer, Java has made accessing a file stored remotely (on a Web or FTP server) as easy as accessing a file stored locally (on a user's floppy or hard disk). Programs can easily be written to allow world-wide access to and interaction with various information resources.
- *Platform independent.* The major strength of Java is that most software developers are no longer forced to choose between computer platforms, or expend great resources porting code from one platform to another. Further, code/class resources developed by others are no longer unavailable for use by programmers simply because the platform on which they were developed is different. The authors find platform independence extremely attractive. Academic communities are littered with a vast assortment of computers and operating systems. Personal preference and differing computing needs have led to campuses with operating systems that include MS Windows, MacOS, and the many flavors of UNIX. Educators' resources are rarely large enough to allow for the porting of code from one platform to another. Choosing one platform means excluding others and thereby limiting the usefulness of the resource to students and contributing faculty. Beyond the local campuses, platform independence greatly increases the usefulness of developed resources to the world-wide Internet community.
- *Interactive content.* Java is making Web pages dynamic. The user provides his/her input with the familiar computing tools of keyboard and

mouse and receives feedback through live/still video and sound.
- *Computation aspects.* Java allows a Web environment in which numerical analysis can be performed on computationally intensive engineering problems. This capability gives educators the freedom to present students with open-ended questions, where the user can be involved in every aspect of problem definition, exploration, and solution; in the past, interaction has only been possible through a fixed set of options presented in fill-out forms.
- *Java applications run locally.* Java runs the Java applets locally, rather than requiring a constant and slow continual transfer of data from a remotely-located Web server. An applet is downloaded in its entirety before it is run; this does not mean that data cannot be passed between the applet and remote data servers as the applet is run. The old method of interaction, CGI programs, required the Web serving computer to make every content displaying decision; with this arrangement, interactive Web content was both difficult to manage and slow.
- *Security.* Security is a significant concern whenever files are being received from an unknown or untrusted site. Harm can come to a computer in many forms, including a computer virus (a program which propagates by inserting its code into the executable code of other programs often ending in the damage/destruction of floppy or hard disk files) and a Trojan horse (a program which claims to perform one desirable function while instead providing an undesirable function). The Java language and program environment have tried to address these concerns by restricting the ways in which a misguided programmer can access the resources of the computer on which his/her code is executed.

Some other advantages are discussed in [16]. Some disadvantages of using Java include the following:

- *Programming knowledge.* Java is a programming language; programming experience and/or great patience and free time is required to turn ideas into Java code. Java is syntactically based upon C++. Anyone familiar with C++, or any object-oriented programming language, will find the transition to Java programming fairly natural [19]. Those with little knowledge of C/C++ may find the alternate technologies of JavaScript, JavaBeans, OpenDoc, and ActiveX to be more useful. The power of Java will become increasingly available to novice users as Java programming tools become more available. Visual tools, visual resource workshops and integrated development environments are available from companies including Symantec [29], Borland [30], Microsoft [31] and JavaSoft [32].
- *Performance concerns.* Since Java must be interpreted as it is executed, there are valid concerns regarding the code's execution speed. Various remedies for this have been proposed. The current remedy being most widely employed is the use of JIT compilers. In this way, platform independence is maintained and performance is improved.
- *Available code.* Java is in its infancy. The availability of third-party classes and programs is still weak when compared to the availability of code for C++/FORTRAN/Visual BASIC. Programmers working in Java must create many of their own classes to handle tasks for which they would be able to obtain third-party libraries in other languages. This is common for any new programming language.

Some other disadvantages of Java are addressed in [16].

USING JAVA TO DEVELOP INTERACTIVE INSTRUCTIONAL COURSEWARE

Educators have been using the Web this past academic year in courses to post course guidelines, homework, etc., and to develop courseware (instructional modules) [10–14, 33]. The second author has worked with two undergraduate students in developing learning modules for bar and beam elements [14]. These modules can be used as part of a introductory finite element course, matrix structural analysis course or any other course in which the instructor would like to introduce basic FEM concepts. The modules include sections on element assumptions, sign convention, solution characteristics, formulation, rigid body modes and example problems. Almost all of the instructional courseware that can be found on the WWW today is completely static and offers no significant advantages over textbooks. New advances in technology demand an evolution from this out-dated approach to instruction. Java has given the Web the power of continuous, interactive, real-time, visual, and aural instruction.

When accessing a Web page, instead of clicking on a passive screen of images and text the user can be prompted for a response by offering an interactive menu while on-line instructions are delivered through an audio/animation sequence. The method of presenting courseware will pique students' interests and understanding of the course material. The rich multimedia flavor of CD-ROMs can now be found in the academic/corporate networks and the Internet [34].

To both demonstrate and test Java's usefulness in an academic environment, the authors have developed a Java applet that allows a student to learn the steps involved in the stiffness matrix assembly process found in the finite element method (Section 5). A student using any Java-enabled browser could load the finite element method tool and interact with it through an input device such as a mouse. The assembly

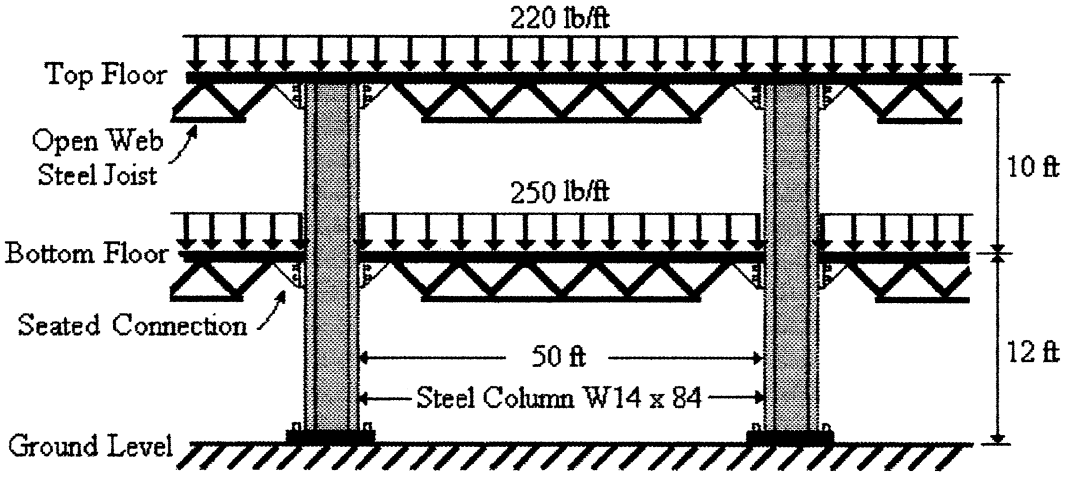

Fig. 1. Problem definition.

applet presented in this paper provides a different learning experience than Java Beam [35] and Java FEA applet [36]. Java Beam and Java FEA applet are programs similar to commercial FEM codes and do not focus on the mechanics of learning the finite element method.

A JAVA APPLET EXAMPLE FOR THE FINITE ELEMENT METHOD

A matrix assembly applet was developed to demonstrate how Java could be used to develop an interactive learning environment for the finite element method on the WWW. The matrix assembly applet can be found at location http://femur.wpi.edu/Interactive-Learning-Tools/. The JDK 1.0 for Windows 95 [17] was used in applet development. A Java-enabled Web browser such as Netscape Navigator 2.0 or higher [5] is required to run the applet on the Web. If your browser does not support Java, the matrix assembly applet will not load. The authors would like to emphasize that the matrix assembly applet can be used on any computer platform, i.e., MS Windows 95, MS Windows NT, Macintosh MacOS, Sun SPARC Solaris, Linux, as long as the computer has a Java-enabled browser. The source code for the assembly applet can be found at http://femur.wpi.edu/Interactive-Learning-Tools/Source/. The processing power of a 486 66 MHz computer with 8 Mb of RAM is sufficient to run the applet.

Consider the example problem in Fig. 1 where steel columns are used to support the symmetric loads from the two floors of a building. A finite element model consists of two one-dimensional bar elements as shown in Fig. 2. Buckling is neglected in this example. The student will use the Java-enabled Web browser to go through the interactive stiffness assembly procedure. Once the appropriate HTML document is loaded, the assembly applet is executed on the host computer and on the terminal screen the student is presented with the element stiffness matrices. The matrix assembly applet is

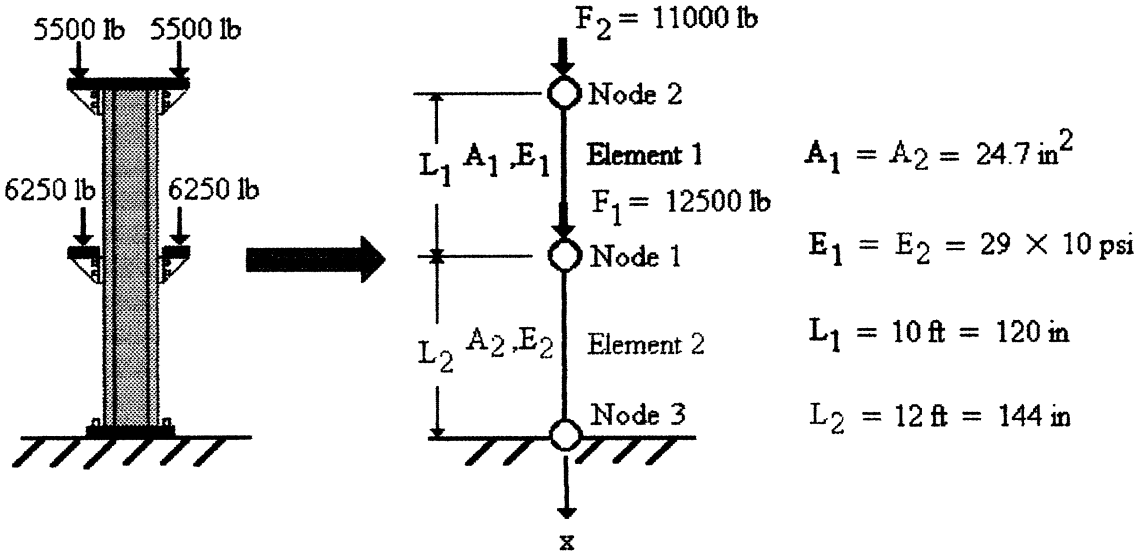

Fig. 2. Finite element mesh.

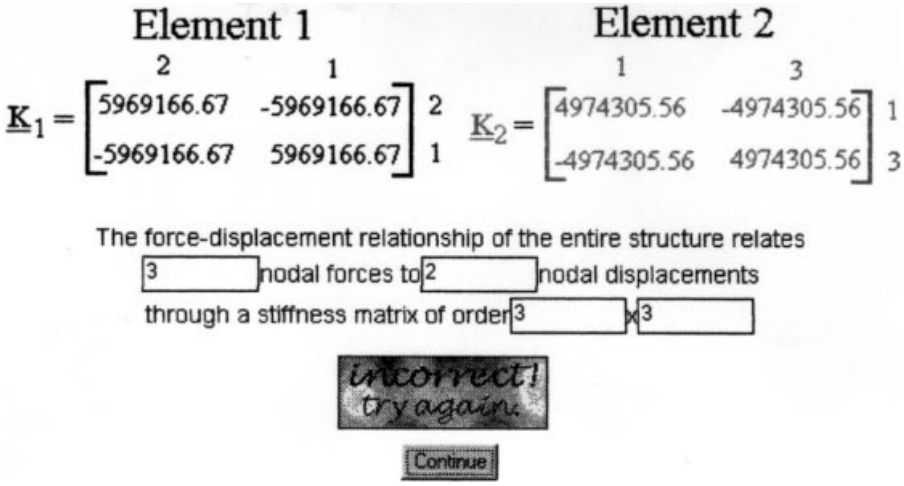

Fig. 3. FEMur-CAL matrix assembly, applet one.

shown in Fig. 3. The force/displacement relationships of elements 1 and 2 are shown in Figure 3 in terms of global node numbers. The element stiffness matrix is of order 2×2 since it relates two nodal forces to two nodal displacements, i.e. $f_e = K_e u_e$. The student then defines the size of the global force/displacement relationship for the problem. Since the mesh contains 3 nodes with one degree of freedom per node, the force/displacement equation relates 3 nodal forces to 3 nodal displacements through a 3×3 global stiffness matrix, i.e. $f = Ku$. Once the equation size is defined by the student, the Java applet will generate the empty global force/displacement relationship as shown in Fig. 4.

The stiffness matrix K of the structure under consideration can be found by using the direct assembly method [37]. It states that the contribution of all elements are simply added into the global stiffness matrix according to the degrees of freedom (node numbers in this case) associated with the elements. This process leads the formation of the global stiffness matrix for the entire structure. The assembly process using the applet is carried out by dragging an element stiffness term and dropping it into the appropriate position in the global stiffness matrix. The drag and drop feature of the learning environment is shown in Fig. 5.

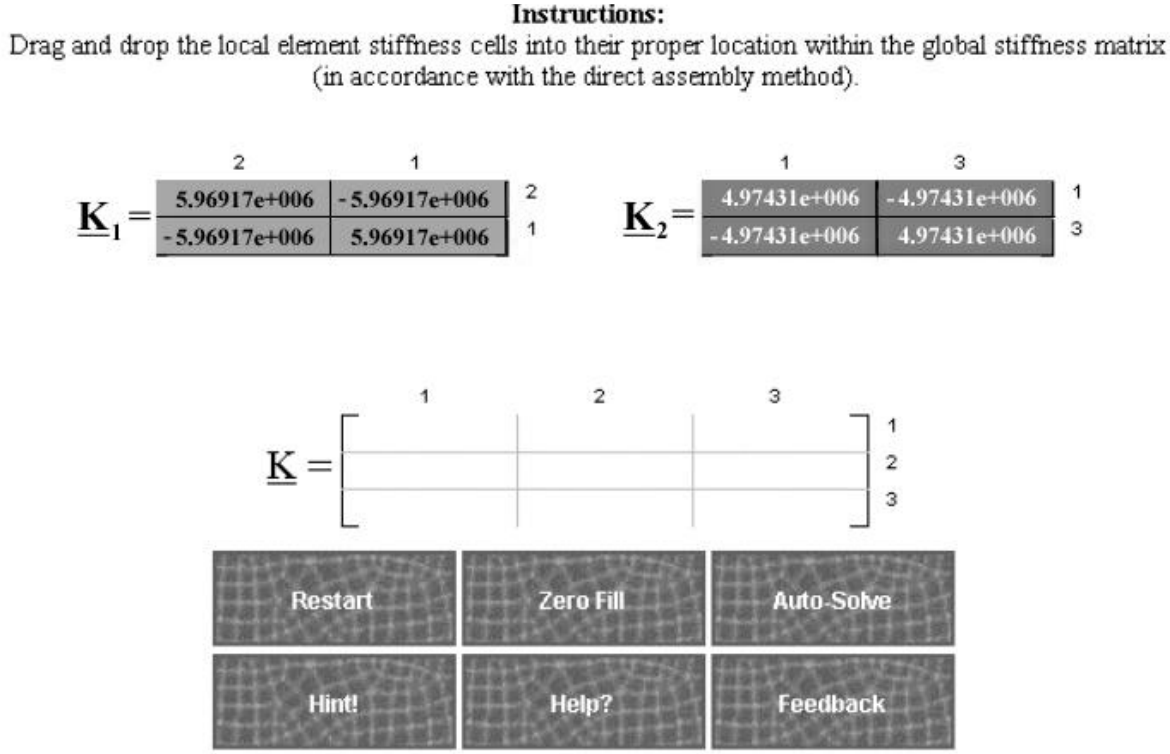

Fig. 4. Empty global force-displacement relationship, applet two.

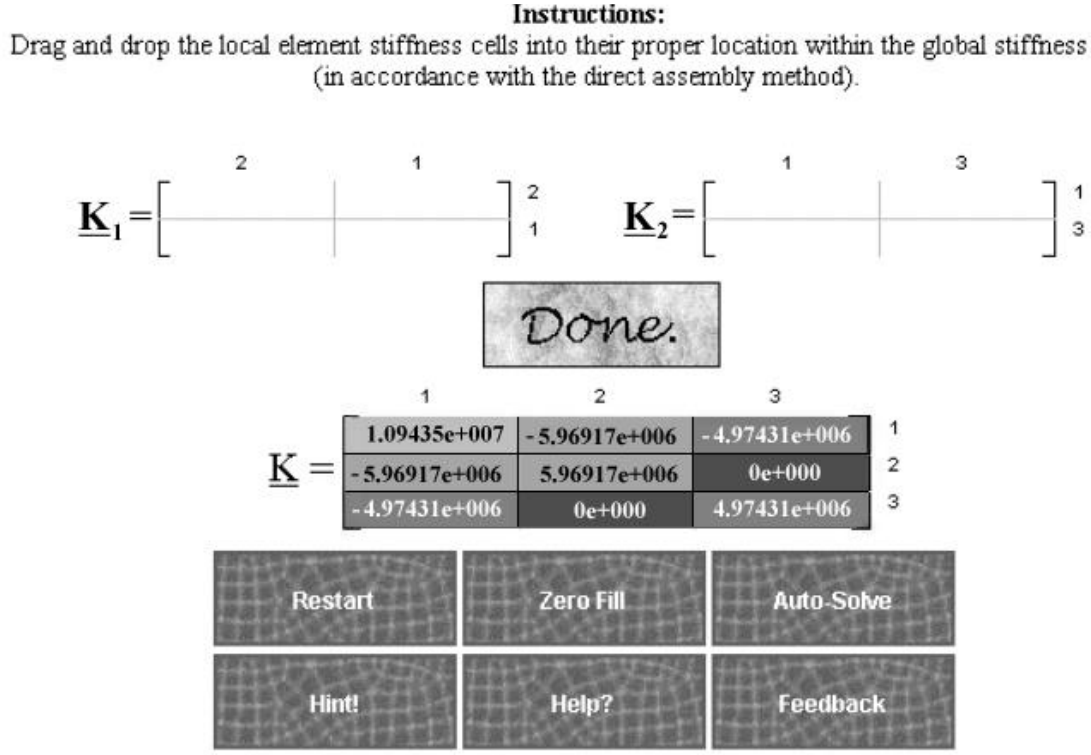

Fig. 5. Done.

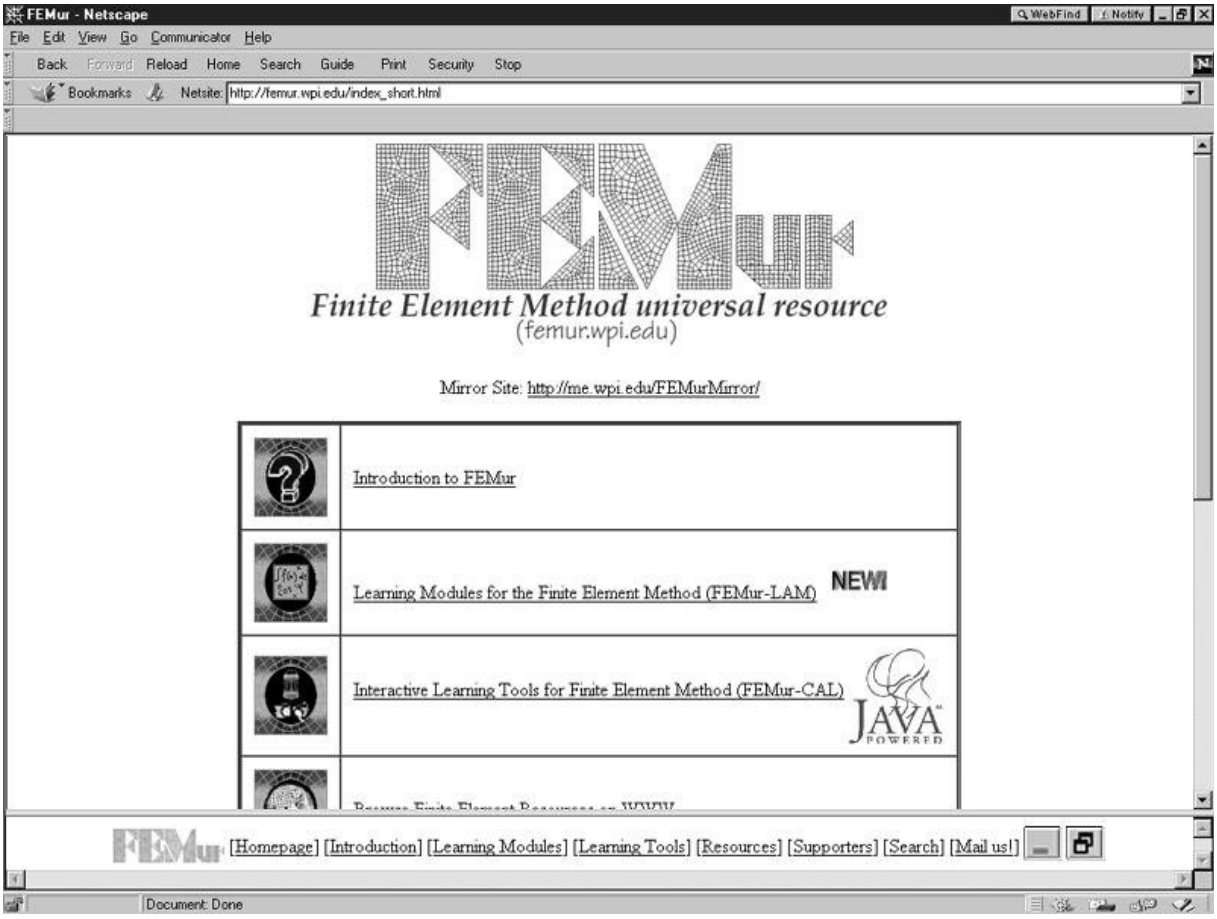

Fig. 6. FEMur homepage.

The matrix assembly applet has been developed with the following educational features in mind:

- *Interactive learning capabilities.* When an element stiffness term is assembled into the global stiffness matrix the student is informed whether or not the term was placed correctly. If an incorrect assembly step is done, the student has the option of trying again or allowing the software to show the correct location. The user can also request that the assembly process be done entirely by the software on an element stiffness term basis or element stiffness matrix basis. The restart button can be used to restart the assembly process.
- *Color coding of element stiffness matrices.* Each element stiffness matrix has been assigned a unique color so that the student can more readily develop an intuitive feel for the relationship between the local stiffness elements and their global stiffness matrix placement. When stiffness terms from different elements are added together (combined) in the global stiffness matrix the color of the matrix position is represented by a combination of all element colors.
- *Zero stiffness terms.* The 'zero fill' button will zero out all unoccupied and non-zero positions in the global stiffness matrix.
- *Developer's feedback.* The feedback button provides an electronic mechanism for the student to notify developers regarding any technical and content problems that arise during the execution of the applet.
- *On-line help.* On-line help has been included if any additional information or clarification is needed regarding the features described above.

FUTURE DEVELOPMENT OF AN INTERACTIVE LEARNING TOOL FOR THE FINITE ELEMENT METHOD

The authors are currently developing a Web site for the finite element method called the Finite Element Method universal resource (FEMur) and its Internet address is http://femur.wpi.edu/. The FEMur homepage is shown in Fig. 6. FEMur includes: *Resources for FEM on WWW* and *Learn the Finite Element Method*.

Resources for FEM on WWW is a collection of organized links to FEM on the World-Wide Web. Some resources include commercial FEM codes, textbooks, other FEM homepages, consulting services, newsgroups, university, government and personal homepages, and professional societies. *Learn the Finite Element Method* is a modular resource that will include theoretical aspects and design/analysis applications of an element in an electronic textbook form. The authors are currently developing a prototype FEM interactive learning tool for the one-dimensional bar element using Java and JavaScript that will be integrated into the *Learn the Finite Element Method* component of FEMur. The interactive learning tool for finite element method is called FEMur-CAL (Finite Element Method universal resource for Computer-Assisted Learning). The matrix assembly applet in the previous section demonstrates on a small scale how Java can be used to develop an interactive learning environment for the finite element method that has a friendly graphical user interface (GUI) and on-line help. Future updates of the FEM interactive learning tool can be found at URL address http://femur.wpi.edu/Interactive-Learning-Tools/.

The student will begin an on-line interactive session in FEMur-CAL by starting with the problem definition and the mesh topology. The student will then proceed to solve the problem in a step-by-step manner according to FEM. Fill-out forms will be used to implement the interactive interface within the WWW browser. The student will also go through the process of working with and solving the finite element matrix equations on-line. Results will be displayed in graphical and tabular forms.

The authors would like to emphasize that the objective of FEMur-CAL is to be a learning tool and it is not intended to be used for commercial purposes in analyzing and designing industry-sized problems. Furthermore, FEMur-CAL is not a substitute for the conventional classroom experience. FEMur-CAL is intended to supplement the classroom as well as be available for examination review and virtual make-up classes.

CONCLUSION

The Java programming methodology is new and there remain some inadequacies and bugs. However, Java supports a new paradigm in computer programming by allowing programmers to create platform independent and Web browser executable software. The future of development effort on the Web utilizing Java appears to be very promising. Since its introduction in the summer of 1995, major software development companies such as Microsoft, IBM, and Oracle [38] have licensed the technology to integrate the Java programming methodology into their WWW-based software products.

This paper provides an overview of Java programming language and its use in the development of interactive content for the WWW. A demonstration applet written in the Java language has been developed for the stiffness assembly procedure. It will be used as an interactive learning tool for the finite element method. The applet program can be executed on any computer connected to the Internet and running a Java-enabled Web browser by accessing the URL http://femur.wpi.edu/Interactive-Learning-Tools/. General information and learning resources on the Finite Element Method may be found at the same site by browsing http: // femur.wpi.ed/. Up-to-date information regarding this paper can be found at http://femur.wpi.edu/Journal-Papers/IJEE/.

REFERENCES

1. K. Hughes, *Entering the World-Wide Web: A Guide to Cyberspace*, Enterprise Integration Technologies (May 1994); http://www.hcc.hawaii.edu/guide/www.guide.html.
2. K. Hughes, *From Webspace to Cyberspace*, Enterprise Integration Technologies, Version 1.1 (July 1995); http://www.jsp.fi/kevinh/.
3. J. Neilson, *Hypertext and Hypermedia*, Academic Press Inc., San Diego (1994).
4. V. Balasubramanian, *State of the Art Review on Hypermedia Issues and Applications*, Graduate School of Management, Rutgers University, Newark, New Jersey (March 1994); ttp://www.isg.sfu.ca/~duchier/misc/hypertext_review/index.html.
5. Homepage for Netscape Navigator; http://home.netscape.com/.
6. Homepage for NCSA Mosaic; http://www.ncsa.uiuc.edu/SDG/Software/SDGSoftDir.html.
7. Homepage for Lynx; http://http1.brunel.ac.uk:8080/depts/cc/www/texts/unix_lynx.html.
8. Homepage for Microsoft's Internet Explorer 3.0; http://www.microsoft.com/ie/.
9. Homepage for JavaSoft's, a Sun Microsystems Company, Mtn. View, California, Hot Java; http: //www.JavaSoft.com/HotJava/.
10. T. G. Shawki, *TAM 221 Mechanics of Materials*, Department of Theoretical and Applied Mechanics, University of Illinois at Urbana-Champaign, Champaign, Illinois (May, 1995); http://e2.tam.uiuc.edu/TAM221/index/.
11. J. Kayser, *Statics Tutorial*, Department of Civil Engineering, Lafayette College, Easton, Pennsylvania (1995); http://www.lafayette.edu/kayserj/statics/cover.htm.
12. T. Martin, *The Development of Interactive World-Wide Web Courseware for Students of Engineering and Technology at Deakin University*, Faculty of Science and Technology, School of Engineering and Technology, Deakin University, Australia (1995); ttp://www.scu.edu.au/ausweb95/papers/education1/martin/.
13. J. R. Bourne, A. J. Brodersen, J. O. Campbell, M. M. Dawant, and R. G. Shiavi, *ES 130: Engineering Science 130: Introduction to Engineering*, Center for Innovation in Engineering Education, Vanderbilt University (1996); http://wwwfp.vuse.vanderbilt.edu:8888/es130/.
14. P. Kwok, E. Flory, and J. J. Rencis, *Bar and Beam Element Learning Modules for Finite Element Method*, Mechanical Engineering Department, Worcester Polytechnic Insitute, Worcester, Massachusetts (1996); http://femur.wpi.edu/Learning-Modules/Stress-Analysis/.
15. JavaSoft, *What is Java?* http://www.javasoft.com/nav/whatis/.
16. J. Gosling and H. McGilton, *The Java™ Language Environment: A White Paper*, JavaSoft (1995); http://Java.sun.com/doc/language_environment/.
17. JavaSoft, *Downloading the Java Developers Kit;* http://www.JavaSoft.com/download.html.
18. EarthWeb, *Gamelan: The Directory and Registry of Java Resources;* http://www.gamelan.com./.
19. BeaconRay's Great Books; *The Java Shelf;* http://jollyroger.com/java.html.
20. *The Java™ Series*, Addison-Wesley Publishing Company, Reading, Massachusetts; http://www2.awl.com/cseng/javaseries/index.html.
21. E. R. Harold, *Brewing Java: A Tutorial*, (January, 1996); http://sunsite.unc.edu/javafaq/javatutorial.html.
22. Netscape Communications Corp., *JavaScript Guide;* http://search.netscape.com/ko/eng/mozilla/3.0/handbook/javascript/index.html.
23. JavaSoft, *JavaBeans Component APIs for Java;* http://splash.JavaSoft.com/beans/.
24. IBM Corporation, *The Club for Enterprise Open Technology;* http://www.software.ibm.com/clubopendoc/.
25. Apple Computer, Inc., *Apple Computer's OpenDoc Web;* http://www.opendoc.apple.com/.
26. Microsoft Corporation, *Microsoft Site Builder Workshop;* http://www.microsoft.com/activex/.
27. Apple Computer, Inc., *Apple Computer's OpenDoc Users' Site*; http://www.opendoc.apple.com/users/users.html.
28. Academic Press, *OpenDoc Programmer's Guide;* http://www.viamall.com/softpro/0-12-624665-3.html.
29. Homepage for Symantec Corporation's Symantec Café; http://cafe.symantec.com/cafe/index.html.
30. Homepage for Borland International, Inc.'s JBuilder; http://www.borland.com/jbuilder/.
31. Homepage for Microsoft Corp.'s Visual J++; http://www.microsoft.com/visualj/.
32. Homepage for JavaSoft's JavaWorkshop; http://www.sun.com/workshop/.
33. J. R. Bourne, A. J. Brodersen, J. O. Campbell, M. M. Dawant, and R. G. Shiavi, a model for on-line learning networks in engineering education, *Journal of Engineering Education*, pp. 253–262, (July, 1996).
34. Engineered Multimedia, Inc., http://www.engmm.com/.
35. *Java Beam*, Coalition of Schools for Excellence in Education and Leadership, College of Engineering, University of Washington, Seattle, Washington (199); ttp://ecsel.engr.washington.edu/JavaBeam/.
36. S. Sinclair, *Finite Element Analysis—Live on the Web!* (1996); http://members.aol.com/trickys/Java/JFEA.html.
37. H. T. Grandin, Jr., *Fundamentals of Finite Element Method*, Waveland Press, Inc., Prospect Height, Illinois, (1991).
38. IBM Corporation, 'IBM licenses Java technology from Sun Microsystems for use in Internet products,' press release, (December 6, 1995); http://www.ibm.com/Java/news/.

Benjamin 'Quincy' Cabell V received his BS from Worcester Polytechnic Institute in Mechanical Engineering. He is currently pursuing graduate school at the same institution, hoping to further develop his skills in mechanical design, writing software for instructional purposes, and developing new software for use by mechanical engineers. e-mail: quincy@besiex.org; WWW: http://www.cabell.org/Quincy

Joseph J. Rencis is an Associate Professor in the Mechanical Engineering Department at Worcester Polytechnic Institute. His research focuses on the development of boundary and finite element methods for analyzing solid, heat transfer and fluid mechanics problems. He received his BS from the Milwaukee School of Engineering in 1980, a MS from Northwestern University in 1982 and a Ph.D. from Case Western Reserve University in 1985. e-mail: jjrencis@wpi.edu; WWW: http://jjrencis.wpi.edu

Javed Alam is a professor of Civil and Environmental Engineering at Youngstown State University. He obtained his BS in Civil Engineering from Indian Institute of Technolgy at Kanpur, India and received his MS from Asian Institute of Technology at Bangkok, Thailand. He pursued further studies at Case Western Reserve University in Cleveland, Ohio to obtain a Ph.D. degree. His research interests are in the area of Structural Mechanics and Computer Applications in teaching. e-mail: jalam@cc.ysu.edu; WWW: http://jove4.eng.ysu.edu/civil/javed.html

Hartley T. Grandin Jr. is a Professor Emeritus of Engineering Mechanics and Design in the Mechanical Engineering Department at Worcester Polytechnic Institute. He received his BS in 1955 and an MS in 1960 in Mechanical Engineering from Worcester Polytechnic Institute and a Ph.D. in Engineering Mechanics from the Department of Metallurgy, Mechanics and Materials Science at Michigan State University in 1972. e-mail: hgrandin@wpi.edu; WWW: http://me.wpi.edu/Documents/People/Faculty/hgrandin.html