DOCUMENT RESUME

ED 421 172 IR 018 892

AUTHOR Knee, Richard H.; Cafolla, Ralph
TITLE Bringing Interactivity to the Web: The JAVA Solution.
PUB DATE 1998-00-00
NOTE 6p.; In: "SITE 98: Society for Information Technology & Teacher Education International Conference (9th, Washington, DC, March 10-14, 1998). Proceedings"; see IR 018 794.
PUB TYPE Reports - Descriptive (141) -- Speeches/Meeting Papers (150)
EDRS PRICE MF01/PC01 Plus Postage.
DESCRIPTORS Computer System Design; *Computer Uses in Education; *Internet; Online Systems; *Programming Languages; Teacher Education; World Wide Web
IDENTIFIERS Browsing; *Java Programming Language; *Object Oriented Programming; Web Sites

ABSTRACT

Java is an object-oriented programming language of the Internet. It's popularity lies in its ability to create interactive Web sites across platforms. The most common Java programs are applications and applets, which adhere to a set of conventions that lets them run within a Java-compatible browser. Java is becoming an essential subject matter and teaching tool within corporate and academic settings. JavaScript was developed to fill the considerable gaps in functionality between HTML, CGI and Java. One of Java's great advantages is its inherent security. There is a challenge involved in integrating Java into traditional education programs. The following outline is suggested for providing educators with the Java skills to meet their instructional objectives: (1) draw up a detailed outline for the class; (2) create a template for content, demonstrations and projects; (3) research content for topics that apply to education, note change; (4) configure technological environment, know your lab and its limitations; (5) design technology demos--use of student work from previous courses; (6) develop the lecture content; (7) develop a resource list of tools, Web sites and projects; (8) develop student and instructor notes via presentation tools; (9) create a plan for updating course materials; (10) develop a list of student objectives or outcomes. (AEF)

Bringing Interactivity to the Web: The JAVA Solution

By:

Richard H. Knee
Ralph Cafolla

Bringing Interactivity to the Web: The JAVA Solution

Richard H. Knee
Florida Atlantic University

Ralph Cafolla
Florida Atlantic University

Java is the object-oriented programming language that has taken the Internet by storm. Java's popularity lies in its ability to create interactive Web sites across platforms. Because of Java, the Internet no longer need remain a static vehicle. It becomes a truly interactive tool for the user.

The most common Java programs are applications and applets. Applications are stand-alone programs, such as the HotJava browser. Applets are similar to applications, but they do not run stand-alone. Instead, applets adhere to a set of conventions that lets them run within a Java-compatible browser. Applets are essentially embedded in HTML pages for Web viewing (Musciano & Kennedy, 1997).

Java was first "brewed" in 1990 when a handful of developers at Sun Microsystems set out to build a device that could control everyday consumer appliances ranging from coffeepots to VCRs. Soon the vision expanded to include a device that could potentially serve as an interface to the Internet. Late in 1991, an object-oriented programming language called Oak was developed (Scott, 1997).

Sun wanted to build a system that could be programmed easily without a lot of esoteric training. While the developers found the existing programming language, C++ to be unsuitable for their needs, they designed Java as closely to C++ as possible in order to make the system easier to understand. In January 1995, just as it began to attract serious attention from the greater Internet community, the Oak programming language was renamed Java.

Java is fast becoming an essential subject matter and teaching tool within corporate and academic settings. Corporations are sponsoring Java classes for their diverse employee pools. Universities are integrating Java into curriculums as a result of the increasing demand for Java developers. Primary and secondary schools are using Java technology to teach students everything from physics equations to biological processes. Java could very well become one of the foundational tools for the classroom of the future, one that allows for a rich, interactive education. Integrating the Web into the curriculum is greatly enhanced when the interface to the Internet is interactive. Java is making that a reality. In this paper, we will explore the growth of the Java language as a tool for use in teacher education as well as the K-12 classroom.

Bowen (1996) states that "The Java programming language is gaining popularity in university and community college curriculums-as a first programming language, as a general programming language better suited to conveying advanced concepts such as concurrency and distributed objects, and as an Internet-related toolset important in continuing-education programs."

The term "object-oriented" is generally used to describe a system that deals primarily with different types of objects. The actions you can take depend on what type of object you are manipulating. For example, an object-oriented draw program might enable you to draw many types of objects, such as circles, rectangles, triangles, etc. Applying the same action to each of these objects, however, would produce different results. If the action were Make 3D, for instance, the result would be a sphere, box, and pyramid, respectively (Musciano & Kennedy, 1997).

Java and JavaScript

Up to now we have looked at Java applets, which are small programs that are separate from the browser and the HTML document. Java applets run in their own space via a separate execution engine. JavaScript is different! JavaScript is a scripting language that taps the native functionality of the browser. You can sprinkle JavaScript statements throughout an HTML document and Java Script-enabled browsers interpret and act upon the JavaScript statements (Musciano & Kennedy, 1997).

JavaScript was developed to fill the considerable gaps in functionality between HTML, CGI and Java. JavaScript is well suited to enabling users to interact with HTML pages. JavaScript code can respond directly to user interaction with <FORM> elements like text boxes, buttons, check boxes, drop-down selections lists and more. It is excellent for client-side image maps and hypertext links. It can also be used to "spice up" Web pages with the date, time, status bar messages and scrolling banners

What about ActiveX?

One of Java's great advantages is its inherent security. Because it was created to run applications over the Internet, its designers were very concerned about security. They had to be careful to make it impossible for malicious programmers to build Java applets that could reach over the Internet

into users' machines to steal passwords, shut down systems, or perform other unfortunate hacks.

To guard against this, Sun developed the "Java sandbox," (referred to as a virtual machine) inside which Java applets are forced to run. This virtual machine blocks all access to potential danger zones on your computer, such as the hard disk and serial ports. While the sandbox makes Java very safe, it also makes Java work slower and somewhat less efficiently than native applications that run directly on your machine without any intermediary processing (Rowley, 1995).

The sandbox security model is one of the most important political differences between Java and its nearest competitor, Microsoft's ActiveX. ActiveX allows access to these "danger zones" and relies on a model of trust, simply telling users that a given Web site wants to run an ActiveX control and asking if this is okay. Simply asking for your permission to introduce foreign, potentially malignant files is viewed by some people as inadequate security protection. As of this writing, Java applets are installed on approximately 1,000 times more Web sites than ActiveX (Scott, 1997). Security assurances are one big reason why.

Java in Education

There is a challenge involved in integrating Java into traditional education programs: Java is not a static technology—yet. It is not "written in stone" the way Pascal and C++ are. Instructors and students must be prepared for changes both in the technology and in the way that it is taught. In his Java Report article*Using Next Generation Technology for Java Education*, Holger Opderbeck, CEO of MindQ Publishing, puts it best: "It's about time we started using the power of modern computing to more effectively teach the complexities of modern computing" (Opderbeck, 1997).

Java is an exciting, dynamic technology that is challenging to teach. With new specifications, new classes, and general updates, one must accept the fact, when teaching Java, that the course will never be the same because the subject matter is in a never-ending state of change. In the past, programming classes were created using the model of "develop the course and then teach." The teaching cycle was typically much longer than the course development cycle. In today's technological environment, curriculum development must be iterative; in other words, it is an ongoing repetitive process that is required due to the constant change of the subject matter and the technology. In order to be Java-compliant, we must follow Sun specifications, which are continually changing. Today, new releases of host-based products are issued every few months, so class material will probably need to be updated on a regular basis.

At Florida Atlantic University, we decided that graduates of the Educational Technology program should have nine graduate hours of programming exposure and its application for education. There are a significant number of institutions of higher education that have made the Java language a core course required prior to C or C+. (McCauley & Manaris report as cited in Culwin, 1997.) Our second course is centered on Java. The purpose of our course was to provide educators with the Java skills to meet their instructional objectives. It takes a lot of planning to put together a course of this nature. We suggest the following outline:

1. Draw up a detailed outline for the class
2. Create a template for content, demonstrations and projects
3. Research content for topics that apply to education – note change
4. Configure technological environment – know your lab and its limitations
5. Design technology demos – use of student work from previous courses
6. Develop the lecture content
7. Develop a resource list of tools, Web sites and projects
8. Develop student and instructor notes via presentation tools
9. Create a plan for updating course materials
10. Develop a list of student objectives or outcomes

We adopted the concepts of object-oriented programming in our course design. Demos are used because a single demo can replace many screen captures. Each new demo file corresponds to a new chapter or topic, and dependencies between chapters and between labs are kept to a minimum. Creating self-contained modules (just like creating objects or reusable components in programming) means that topics can be written independently and rearranged or altered with minimal impact to the other sections of the course.

An FTP site was created for the course so that material can be shared and updated by the students. Student work can be put on the server for all to explore as the course develops. Java students also developed a Web site as a project, an exciting way of using the medium to teach the medium.

Apple's EOE -Educational Object Economy Web site (Apple Computer, 1997) is a community of educators, learners, developers, and businesses, focused on the creation and collaboration of educational activities which include pieces of Java software in them. The EOE is also intended to help educators and learners access this material and the creators of the applets. The EOE has a library of over 1000 pointers to Java applets, over 25% of which make source code available. Working together, the educators, learners, and developers can collaborate to enhance existing material and produce new innovations. Educators with little or no programming background have access to an ever growing library of interactive applets to

use as stand alone teaching tools or as enrichments to Web-based instruction.

The EOE has an exciting future for the educator. Educators are providing information about their favorite applets that they use in their classrooms, what applets work best with the textbooks they are using, hints about lesson plans and materials used to effectively use the Java applets to achieve learning objectives. Teachers also make suggestions for what applets they would really like to see created (teacher's wish lists). The EOE also encourages partnerships between teachers and university Java programming classes, so that teachers have access to programmers and developers who are creating the kinds of learning material the teachers need.

EOE provides an excellent scenario of a fifth grade teacher who finds an applet to teach her class a botany lesson. By working with the Java programmers on the listserv she is also able to have the program modified to fit her specific needs within four days. Java is making the Web an interactive pedagogical tool within the reach of all educators' especially those with no programming experience (Bastiaan, 1997).

Who is building Web sites today? Entrepreneurs, writers, hobbyists, educators and students from the elementary grades and up are building them, not Java programmers. In fact, very few Web sites are actually built by professional programmers. That is why programs like Lotus BeanMachine are important: It brings the power of Java to non-programming Web-builders like teachers and their students.

There are three groups of parts that come with BeanMachine. There are Multimedia parts, such as Animation, Audio, Ticker Tape, and Text. There are Networking parts, such as a Database component, a URL Link for linking to other Web pages, and an e-mail part. The third category of parts is called Controls, which includes all the basic things you need to build data entry forms such as buttons and text fields. Anyone that has experienced a modest amount of Visual Basic will find himself or herself at home with this software but we stress once again, no programming knowledge is required.

Every JavaBean has a certain set of properties, methods, and events. Properties describe the part, its speed, its width and height, etc. Every bean has a different list of properties that makes it unique. Beans are active objects, as they know how to do things. Animations can play and stop. Buttons can show and hide themselves. Again, every kind of bean has a different set of actions it knows how to perform; these actions are referred to as methods. Events help a bean interact with other beans. An event is a signal that something important has happened: a button was clicked, a transition has finished, or a ticker tape was refreshed. In BeanMachine, every part on the palette is a JavaBean, so it has properties, methods (which BeanMachine calls actions), and events (Lotus, 1997). The BeanMachine is just one example of software that will allow you to "brew a perfect cup of Java" for your next Web page without really learning how to program in Java.

Conclusion

Java is becoming a basic skill for a wide range of students and teachers, not only for programmers. Educators are beginning to recognize that today's students need to learn logic, or "higher order process skills," at an early age. Java helps students develop their logical or "object oriented cognitive skills," an understanding of how, for example, the "objects" of car, truck, and sports car relate to each other. This notion of objects and their relationships, of classification and sub-classification, is important in disciplines ranging from art to chemistry, and it is fundamental to the Java environment. The skills it teaches are crucial, for logic is the essence of cognition, and language is its tool (Spohrer & Wolpert, 1997).

References

Apple Computer (1997).*EOE - Educational object economy.* [On-line]. Available: http://trp.research.apple.com/FMPro?-db=DNDdb&-format=info.html&-view

Bastiaan, M. K. (1997).*Educational object economy.* [On-line]. Available: http://trp.research.apple.com/index.html

Bowen, B. (1996).*Educators embrace Java.* [On-line]. Available: http://www.javaworld.com/javaworld/jw-01-1997/jw-01-education.html

Culwin, F., (1997). Joining Java*Monitor Editoral.*[On-line]. Available: http://www2.ulst.ac.uk/misc/cticomp/editorial.html

Lotus Corporation. (1997). [On-line]. Available: http:// www2.lotus.com/home.nsf/ welcome/beanmachine

Musciano, C. & Kennedy, B. (1997). HTML - the definitive guide. Cambridge, MA: O'Reilly & Associates.

Opderbeck, H. (1997). *Using next generation technology for Java Education*. New York: MindQ Publishing.

Rowley, K. (1995). Understanding software interoperabiity in a technology-supported system of education.*Cause/Effect.* [On-line]. Available: http://cause-www.colorado.edu/information-resources/ir-library/pdf/cem9535.pdf

Scott, R. (1997).*Introduction to JavaScript scripting, logical operations.* New York: Ziff Davis.

Spohrer, J. & Wolpert, J. [Online]. Available: http://www.ibm.com/java/education/educational-resources.html

Additional Resources

- Doug Lea's Q&A on using Java at SUNY-Oswego http://g.oswego.edu/dl/html/javaInCS.html
- Sun Microsystems resources on Java in Education http://www.sun.com/edu/java
- Selected Java newsgroups comp.lang.java.programmer — Java language discussion comp.lang.java.misc — IDEs, books, etc.
- Java FAQ - includes selected Books and language information http://sunsite.unc.edu/javafaq/javafaq.html
- Java Booklist http://lightyear.ncsa.uiuc.edu/~srp/java/javabooks.html

- Web technologies, including Java, in teaching
  http://www.npac.syr.edu/users/gcf/webwisdomapr96/
- Sun Microsystems Java CourseWare Page
  http://www.sun.com/edu/java
- Sun Microsystems resources on Java in Education
  http://www.sun.com/edu/hot/java/index.html
- Footprint Software
  http://www.footprint.com

Richard H. Knee is an Assistant Professor of Educational Technology and Research in the College of Education, Florida Atlantic University, Davie, Florida 33314.
E-mail: Knee@fau.edu

Ralph Cafolla is an Assistant Professor of Educational Technology and Research in the College of Education, Florida Atlantic University, Boca Raton, Florida 33431.
E-mail: cafolla@fau.campus.mci.net

U.S. DEPARTMENT OF EDUCATION
Office of Educational Research and Improvement (OERI)
Educational Resources Information Center (ERIC)

ERIC

NOTICE

REPRODUCTION BASIS

☒ This document is covered by a signed "Reproduction Release (Blanket)" form (on file within the ERIC system), encompassing all or classes of documents from its source organization and, therefore, does not require a "Specific Document" Release form.

☐ This document is Federally-funded, or carries its own permission to reproduce, or is otherwise in the public domain and, therefore, may be reproduced by ERIC without a signed Reproduction Release form (either "Specific Document" or "Blanket").