Improving the Interactivity and Functionality of Web-based Radiology Teaching Files with the Java Programming Language[1]

John Eng, MD

Java is a programming language that runs on a "virtual machine" built into World Wide Web (WWW)-browsing programs on multiple hardware platforms. Web pages were developed with Java to enable Web-browsing programs to overlay transparent graphics and text on displayed images so that the user could control the display of labels and annotations on the images, a key feature not available with standard Web pages. This feature was extended to include the presentation of normal radiologic anatomy. Java programming was also used to make Web browsers compatible with the Digital Imaging and Communications in Medicine (DICOM) file format. By enhancing the functionality of Web pages, Java technology should provide greater incentive for using a Web-based approach in the development of radiology teaching material.

■ INTRODUCTION

The exponential growth of Internet use has largely been the result of the implementation of data transfer protocols and file servers constituting the World Wide Web (WWW). The Web's compelling combination of textual information, digital images, and digital audio creates a natural medium for institutional activities such as the radiology teaching file (1-4).

The Web-based approach adds much to the venerable radiology teaching file in terms of accessibility and archiving efficiency. The Web's hypertext paradigm also allows easy navigation between related screens of information (Web "pages"). Despite these benefits, however, the appearance and presentation of radiology teaching cases on the Web are functionally identical to the format found in printed textbooks: static images accompanied by static text. This format presents a dilemma to an author: If a case is presented as an unknown, the user will be required to perform tedious navigation between the image and its "answer," which by definition cannot appear on the same screen as the image. One solution would be to create a second version of each

Abbreviations: DICOM = Digital Imaging and Communications in Medicine, HTML = hypertext markup language, JVM = Java virtual machine

Index terms: Computers, educational aid • Computers, multimedia • Education • Internet

RadioGraphics 1997; 17:1567-1574

[1]From the Department of Radiology and Radiological Science, The Johns Hopkins Hospital and School of Medicine, 600 N Wolfe St, Baltimore, MD 21287. Presented as an *info*RAD exhibit at the 1996 RSNA scientific assembly. Received March 4, 1997; revision requested May 14 and received July 16; accepted July 24. **Address reprint requests to** the author.

image in which the findings and diagnoses are labeled as they would be in a textbook. However, a second image doubles both the time required to download the case over a computer network and the storage requirements of the image server and significantly increases the amount of effort required for case preparation.

Ideally, what is needed is a method of going beyond the textbook paradigm and overlaying user-selectable arrows and labels on a single version of the image. The current version of the Web page description standard, HyperText Markup Language (HTML) version 3.2, provides no such capability. A recently developed programming language, Java, does provide a way of placing this interactive capability into otherwise standard Web pages. In this article, the basic concepts of Java are introduced and applied to the "quiz display problem" and other related difficulties of Web presentation, thereby demonstrating how Java programming fundamentally improves the content, interactivity, and functionality of Web-based radiology teaching material.

■ JAVA PROGRAMMING

Java is a programming language developed by Sun Microsystems (Mountain View, Calif). The language's syntax is roughly a hybrid between C and C++. Since its initial release in May 1995, Java has received unprecedented attention from programmers and the public alike (5,6) due to several unique features of the language, including cross-platform compatibility, integration with a standard programming library, and Internet distribution capability.

• Cross-platform Compatibility

Java programs run on an imaginary computer called the Java virtual machine (JVM). Therefore, a Java program file will run on any real computer that emulates the JVM properly. The JVM thus serves to standardize the different hardware platforms. JVM emulators (which are platform dependent) are available free of charge for most computers running the Macintosh (Apple Computer, Cupertino, Calif), Windows (Microsoft, Redmond, Wash), or UNIX operating systems.

• Standard Programming Library

Unlike other programming languages, Java is tightly integrated with a programming library that includes standard subroutines for the graphical user interface (GUI), graphics display, networking, and many other basic functions. The library is a required part of the language and a crucial component of Java's cross-platform compatibility.

• Internet Distribution of Programs

The JVM has been incorporated into popular Web-browsing programs such as Navigator (Netscape Communications, Mountain View, Calif) and Internet Explorer (Microsoft). This allows Java programs to be embedded into Web pages and distributed widely over the Internet. These embedded Java programs are known as applets. Java applets are embedded into Web pages with a special HTML tag in much the same way as images and graphics are embedded. The freedom provided by a true programming language allows applets to greatly enhance the interactive features of a Web page. Because applets are downloaded along with their associated Web pages, the user is always guaranteed of having the latest version of an applet. A number of security features are built into the JVM to prevent downloaded applets from damaging the user's system, whether by accident or malicious intent.

■ CREATION OF THE WEB SITE

The Web site described in this article was created on a Power Macintosh 7200 desktop computer (Apple Computer), although other platforms could have been used with similar results. Images for the quiz cases began as standard black-and-white photographic slides produced from the original films. These slides were then converted to the PhotoCD format (Eastman Kodak, Rochester, NY) by a standard processing laboratory. The PhotoCD files were read directly into Photoshop (Adobe Systems, Mountain View, Calif) for editing and composition. Other images were acquired as Digital Imaging and Communications in Medicine (DICOM) image files and converted with NIH Image (National Institutes of Health, Bethesda, Md) into a format readable by Photoshop. Except for the DICOM display portion of the project, all final images were stored in graphics interchange file (GIF) format.

Figure 1. Interactive image annotation for unknown cases. Screen shows an HTML document that combines an embedded Java applet (consisting of a scrolling image display and mode selector buttons) with borderless HTML frames (one each for "History," "Findings," and "Diagnosis") to give the user complete control over the quantity and sequence of information shown for each case. Each unknown case is presented as an image within a scrolling display area. A brief clinical history is also shown for each case. The user can select one of the three display modes, depending how much "hinting" is desired. This screen is in the "Hide Findings" mode. For cases involving multiple images, the separate images would be combined into a single composite image presented within the scrolling display area.

The Web site's HTML documents were created with a simple yet powerful freeware text editor, BBEdit Lite (Bare Bones Software, Bedford, Mass). Although more specialized software packages are available for preparing Web pages, such software was not required for this project.

Java applets were written and compiled with CodeWarrior (Metrowerks, Austin, Tex). The Web pages were optimized for the Navigator 3.0 browsing program. More specifically, the Web pages contain routines written in Netscape's scripting language, JavaScript, which enables parts of a Web page to control an embedded Java applet. This interaction through JavaScript allowed, for example, a Web page's buttons, selection lists, and other screen controls to change the image, graphics, or text being displayed within the Java applet. These screen controls were later incorporated into the Java applets themselves, resulting in standalone applets that do not require JavaScript for control and therefore are functional within other Web browsers.

■ INTERACTIVE IMAGE ANNOTATION

The Java programming library includes methods for displaying digital images and graphics in the same screen area. With the help of this library, a Java applet was written to solve the quiz display problem for Web-based radiology teaching cases (Figs 1-3). The overall design goals for this page were (*a*) to give the user an opportunity to detect the findings and make the diagnosis from the image alone; (*b*) to allow the user to request additional information

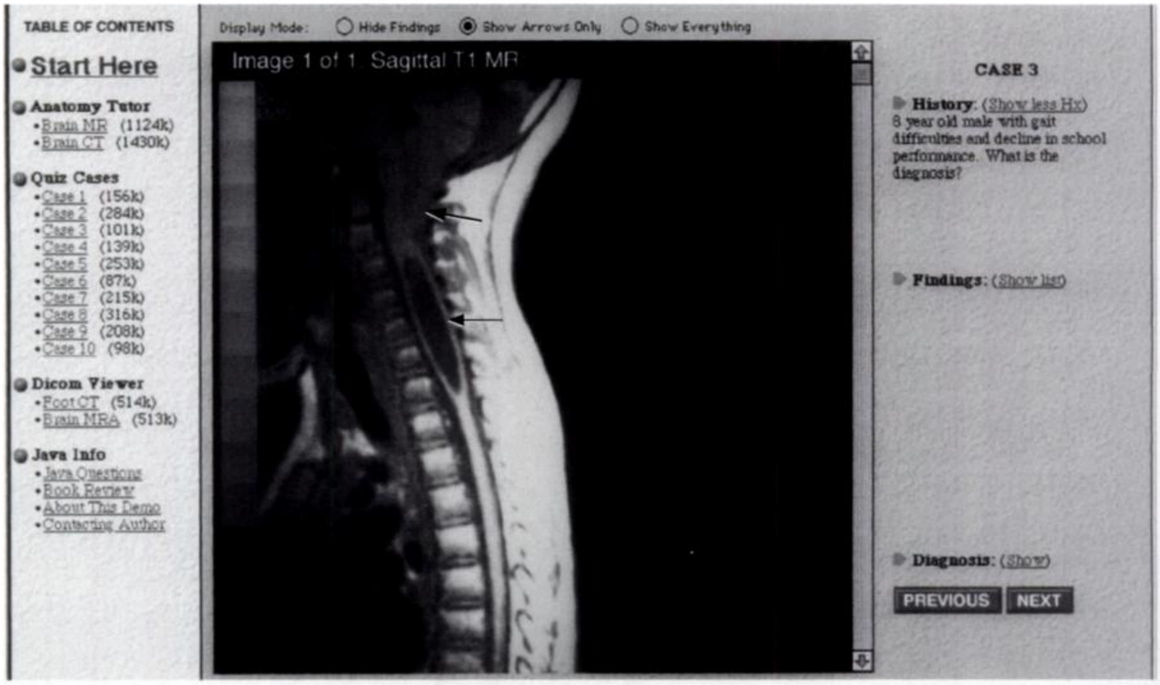


Figure 2. Interactive image annotation for unknown cases. Screen of the same document displayed in the "Show Arrows Only" mode. By selecting this mode, the user can choose to have arrows drawn on the image to point out the findings. This feature is particularly helpful for cases in which the findings are not as obvious, as in this case. Clicking on a hypertext link under "History" reveals a more detailed clinical history.

about the case as "hints" to the diagnosis, including partial and complete image annotation with arrows and labels; and (*c*) to allow the user to see the diagnosis or other information about the image without having to scroll the image off the screen or download a new Web page. The resulting page better simulates a case presentation in a traditional radiology teaching conference. In addition to the Java applet, HTML frames were instrumental in allowing these design goals to be met. The image, clinical history, list of findings, and diagnosis are presented in separate borderless frames on the same page so that the user can make independent choices in each area.

A list of the findings on each image and data for the position and labeling of each finding are contained in a case description file that the image display applet reads from the Web server. The case description file is a plain text file that follows a simple format consisting of five columns: x coordinate of finding, y coordinate of finding, length of arrow, orientation of arrow, and annotation text. To generate a new case, the screen coordinates are obtained from a graphics program such as Photoshop and, along with the other columns, are entered into a new case description file with any word processor or plain text editor. The applet itself requires no reprogramming.

The concept of superimposing graphics on top of images can be similarly applied to other forms of electronic radiology teaching material,

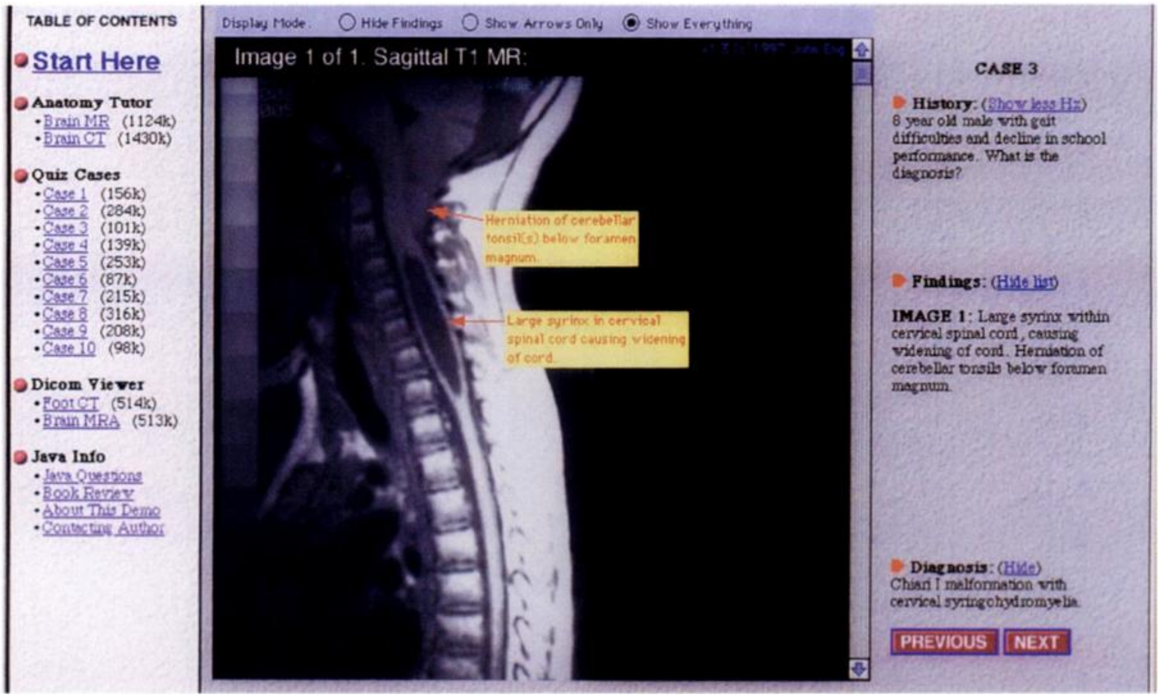
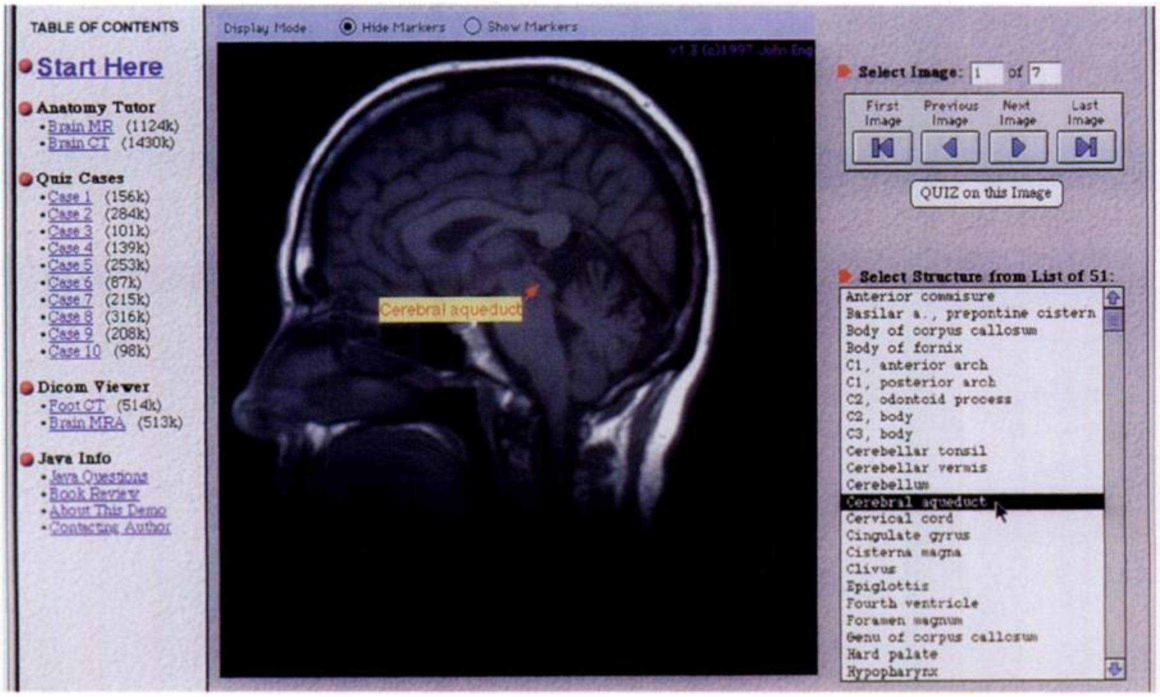


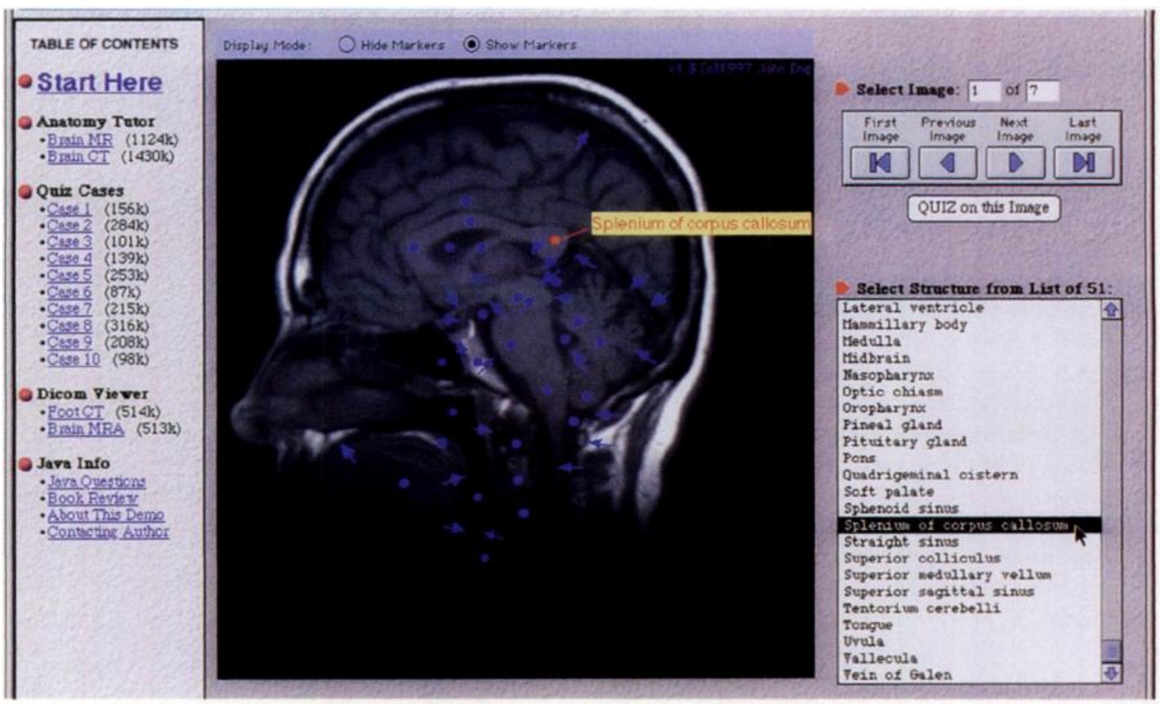
Figure 3. Interactive image annotation for unknown cases. Screen of the same document displayed in the "Show Everything" mode. By selecting this mode, the user can choose to have a label drawn next to each finding in the image. This is a quicker, less tedious method of describing the findings than navigating back and forth to a different portion of the screen or a different page altogether. Annotations are superimposed on a solid rectangular background for easy reading against the image. Hypertext links are available to display a list of findings or the diagnosis in separate HTML frames.

creating functionality beyond what is possible with standard HTML Web pages or even printed material. For example, the quiz display problem is also apparent in the presentation of normal radiologic anatomy. Creators of this material can choose from one of two formats: (*a*) label each anatomic structure with a number that corresponds to the numbered name of the structure in the figure caption, or (*b*) label each anatomic structure directly on the image. Each option has its disadvantages: With the first option, it is tedious to refer back and forth between the numbering on the image and the numbered list in the caption; with the second option, the opportunity to quiz oneself is lost.

To solve the "anatomy display problem," another Java applet was developed (Figs 4, 5). With this applet, the user can choose either of the two possible anatomy display formats. As with the quiz applet, a list of the anatomic structures present in each image and data for the position and labeling of each structure are contained in an editable text file separate from the applet itself. This applet also forms the basis for a searchable electronic imaging atlas, although the atlas has not been implemented yet.



4.



5.

Figures 4, 5. Interactive image annotation for radiologic anatomy. Screens show one of seven sagittal sections in the image display area; the user proceeds through consecutive sections by using the controls shown at the upper right. By clicking on a mode selector above the image, the user can choose to have anatomic structures demonstrated in one of two ways. **(4)** In the "Hide Markers" mode, each anatomic structure is labeled by clicking on a selection from the list of all structures present on the image. **(5)** In the "Show Markers" mode, a dark blue arrow or dot is displayed for each structure on the image. The user can choose to have a structure labeled by positioning the cursor on top of the corresponding image marker or by clicking on an item from the list of structures. (The structure selection list and image navigation control are HTML elements that control the Java applet through JavaScript.)

Figure 6. DICOM image processing. Screen shows an HTML page whose scrolling panel in the center is a Java applet that reads and displays DICOM image files. Information from the DICOM image header is displayed in a borderless frame to the right of the image. The user can interactively manipulate the window and level of the image. In this demonstration applet, only one image can be displayed at a time. (The horizontal scroll bars ["Brightness Control" and "Contrast Control"] are themselves small Java applets. Standard HTML does not provide these scroll bar controls. The scroll bar applets "communicate" with the main image display applet by way of JavaScript.)

■ NEW IMAGE FORMATS

It is usually best to convert electronic case material into optimally windowed and leveled 8-bit gray-scale images to reduce data transmission time and save each user the effort of manual contrast manipulation. However, as network speeds increase and digital radiology archives become more commonplace, it may become necessary to access case material directly from DICOM-compatible image servers and acquisition devices. As a first step toward supporting the additional image format, a Java applet was developed to display 16-bit DICOM image files (Fig 6). DICOM images can be windowed and leveled automatically by the Java software or manually by the user (Fig 6) without loss of gray-scale detail, which otherwise can occur in the standard 8-bit gray-scale formats normally used on the Web. In the future, it will be possible to integrate this DICOM image display engine with the interactive annotation functions discussed earlier to form an interactive educational viewer for DICOM images.

■ LIMITATIONS, RECOMMENDATIONS, AND THE FUTURE OF JAVA

Java programming for the Web is not a cure-all. Because Java is a true programming language (albeit easier to program with than either C or C++), development of Web pages containing Java requires a computer programmer. Compared with other programming languages, Java is very young, and its specifications are still evolving. As of this writing, many implementations of the JVM are sloppy in their adherence to the Java standard, and the standard itself may be vague in some parts. As a result, rigorous attention is presently required to write Java programs that are truly cross-platform-compatible. The latest version of the Java standard, version 1.1, promises to address these compatibility difficulties.

Because Java programs are always run within JVM emulators, they are 2–10 times slower than programs written and compiled in

traditional languages such as C. However, execution speed will become less of an issue as in-line compilers and faster computers are developed. In theory, Java programs can slow the loading of the Web pages in which they are embedded, as can any other object in the page. Embedded Java applets are typically not large, however. Each of the applets described in this article contains 20–25 kbytes, compared with hundreds of thousands of bytes in each image.

Despite these concerns, the Java programming language can be recommended for the following applications: (*a*) adding new interactive features to Web pages, especially those involving the display of graphics or images or special graphical user interface features not available with standard HTML; (*b*) writing programs for fetching and manipulating files, images, or other data directly from the Internet; (*c*) writing programs in which the same compiled program code file can be run without modification on multiple platforms; and (*d*) writing programs that can be distributed over the Internet.

The inability to overlay elements on a Web page is an important recognized limitation of HTML. To address this and other limitations, Netscape Communications and Microsoft have built a new mechanism into their Web-browsing programs called "dynamic HTML," which promises the abilities to overlay elements on a Web page and to manipulate their positions and other properties under the control of embedded scripts (eg, JavaScript). As of this writing, the two implementations of dynamic HTML are not completely compatible, and there are unresolved specification issues between the two companies and the World Wide Web Consortium, the HTML standards organization. In the future, it will probably be possible to use dynamic HTML to duplicate the functionality of Web pages containing Java; however, it is unlikely that less programming will be required. In the author's experience, HTML scripting languages like JavaScript are less efficient than Java for controlling the interactive behavior of a Web page.

Future work on this project will focus on reducing the effort required to create new case material. A stand-alone Java case editing program is being developed that will enable authors to specify the location and appearance of an image's annotation arrows and text with a graphical user interface, eliminating the need for manual entry of numeric data into the case description file. Under consideration is expansion of the case description file format to handle the positioning of multiple images per case, which would eliminate the need for manual composition of individual images in Photoshop.

Although the Web may be a compelling medium with which to distribute radiology teaching cases, Web technology alone does not eliminate the substantial amount of time, effort, and attention to detail required to collect and develop a high-quality teaching file. Java is an exciting new tool in Web-based software development, but it should not overshadow the importance of sound design principles for Web pages. Nevertheless, by enhancing the presentation of radiology teaching material with emerging technologies such as Java, it is possible to provide additional functionality beyond that which is available with standard Web pages and printed material without increasing the incremental effort required for each case. This additional functionality should provide greater incentive for using a Web-based approach in the development of radiology teaching files.

On-line Supplements: The Web site presented in this article may be accessed at the following Internet location: *http://www.rad.jhu.edu/~jeng/javarad/*. The Web site includes instructions regarding system hardware and software requirements, as well as information for Web developers regarding the availability of Java applets for the development of new cases.

■ REFERENCES

1. Richardson ML. A World-Wide Web radiology teaching file server on the Internet. AJR 1995; 164:479–483.
2. Galvin JR, D'Alessandro MP, Kurihara Y, Erkonen WE, Knutson TA, Lacey DL. Distributing an electronic thoracic imaging teaching file using the Internet, Mosaic, and personal computers. AJR 1995; 164:475–478.
3. McEnery KW. The Internet, World-Wide Web, and Mosaic: an overview. AJR 1995; 164:469–473.
4. Mammone GL, Holman BL, Greenes RA, Parker JA, Khorasani R. Inside BrighamRAD: providing radiology teaching cases on the Internet. RadioGraphics 1995; 15:1489–1498.
5. Elmer-Dewitt P. Why Java is hot. Time, January 22, 1996, 58–60.
6. Croal N, Rogers A. Now for some HotJava. Newsweek, November 13, 1995, 87–89.