INVITED COMMENTARY

Java and Teleradiology

Nuclear medicine has been at the forefront of digital radiology for many years, in large part because of the digital nature of γ cameras and the relatively small image file sizes. Teleradiology and picture archiving and communication systems (PACS) are maturing, being carried along with advances in Internet computing, data transmission, image compression, and image file standards. The article by Slomka et al. (*1*) in this issue of *The Journal of Nuclear Medicine* uses many of these technologies, and a brief discussion of the state of the art in these areas may be beneficial.

JAVA: A SMOOTH BLEND OR A BITTER BREW?

Java (*2–4*), developed by Sun Microsystems (Mountain View, CA), is a computer programming language designed to be machine independent. For ease of programming, most computer programs are written in a "high-level" language, such as C or FORTRAN. Before they can be run, these programs need to be compiled to a more basic form that is specific to a type of computer processor and operating system (OS). Thus, programs written for the Intel (Intel Corporation, Santa Clara, CA) processor (Windows OS; Microsoft, Redmond, WA) will not run on a PowerPC processor (Mac OS; Apple Computer, Cupertino, CA), unless the manufacturer has compiled two versions of the same program and provides both versions on the disk.

Java attempts to surmount this machine barrier, producing a single version of a program that can run on any computer. To accomplish this, something must be interposed between the Java program and the computer processor, translating the Java instructions on the fly into a form that can be understood by that machine. This task is performed by a Java virtual machine (JVM). Once you have the JVM written for your specific type of computer, you can then run any Java program.

Initial versions of the JVM ran slowly, interpreting the Java code one line at a time. More recent versions translate large sections of Java code and hold the translated code for later reuse, using a just-in-time compiler, greatly speeding the running of Java programs.

When coupled with the Internet and the World Wide Web (WWW), a major advantage of Java is that programs can be downloaded as part of a web viewing session, adding capabilities to the web browser without the need for any program installation by the user. These capabilities can include gray-scale adjustment, cine viewing, and viewing of orthogonal tomographic planes, as demonstrated by Slomka et al. (*1*) and others. This transfer of processing power to the local machine allows the user to manipulate the images locally, avoiding the retransmission of data that would be necessary if the image adjustments were done by the sending computer (server).

Allowing remote computer programs to automatically execute on your computer could be dangerous; however, Java has built-in security features to prevent such programs from tampering with your machine (*3*). Version 1.0 of Java executed these small downloaded applications (applets) in a restricted environment termed a "sandbox," which allowed access to the computer processor and display, but prohibited access to your computer disk drive. Java 1.1 added the concept of trusted applets, enforced by encrypted digital signatures, that would be allowed free local computer access. Java 1.2 / 2.0 will add the capability of more specific permissions, such as allowing an applet to store data in an "incoming images" folder, but forbidding access elsewhere.

Unfortunately, Java in the real world falls somewhat short of the ideal. Java has been evolving rapidly, growing in built-in features and routines (APIs) from 200 in 1995 to 1600 today (*5*). Different users are running various versions of the JVM, some of which may lack the features needed by a specific Java program and some of which may be "buggy" and unstable. This can lead to unpredictable effects and computer crashes when running Java code and makes Java code difficult to debug, because each Java program needs to be verified to work on a wide variety of computer systems. Because of this, there has been more emphasis recently on using Java to provide a needed set of features to a specific set of users (who use JVMs testable by the programmer), rather than to add frills to web sites accessed by a wide and unpredictable set of users. On-call viewing of nuclear medicine images by a specific set of users is therefore a situation reasonably well adapted for Java use.

A VIEW FROM AFAR

Java is one of several techniques that can be used for teleradiology; other methods are listed in Table 1. As can be seen, there are limitations to each of these methods. A well-designed Java program for remote viewing has the potential to combine all of the advantages of these methods, while avoiding most of the disadvantages. Java has the additional advantage of needing no installation, other than requiring a recent web browser to be present on the remote machine.

The potential of Java in radiology has stimulated recent work in this field (*1,6–18*), and at least one new commer-

Received Jun. 14, 1999; revision accepted Aug. 6, 1999.

For correspondence or reprints contact: Jerold W. Wallis, MD, Mallinckrodt Institute of Radiology, Washington University School of Medicine, 510 South Kingshighway, St. Louis, MO 63110.

TABLE 1
Other Teleradiology Methods

Method	Advantages	Disadvantages
Manufacturer supplied remote viewing station	Robust, well-integrated solution	Expensive, leading to limited number of viewing stations
Display of static images on web pages	Inexpensive, multiplatform	No gray-scale adjustment or cine capability*
Use of X-windows or other screen-mirroring programs to view the workstation display at a remote site	Inexpensive	May not work with all vendors' software; slow because minor changes in display may require retransmission of entire screen
Manual download of files, with remote viewing in general image-viewing programs	Inexpensive	Multiple steps required of user; viewing programs may work on only a single computer platform*
Semiautomatic web-based download of preformatted files with automatic loading into local image-viewing programs	Inexpensive	Viewing programs may work on only a single computer platform*

*These methods may require custom programming (e.g., for file format conversion).

cial teleradiology system is Java based (*19*). The remote-viewer Java applet developed at Washington University (*10*) is in many ways similar to that of Slomka et al. but lacks the tight integration with a clinical database. Other work has focused on Java enhancement of viewing of images in on-line publications (*9*) and teaching (*6,14,20*) and in creating remote viewers capable of receiving images and data directly from Digital Imaging and Communication in Medicine (DICOM) imaging and storage devices (*12–18*).

Our experience parallels that of Slomka et al., in that we found Java implementations on personal computers to be sufficiently fast to be useful for home viewing of nuclear medicine images, especially on machines using a just-in-time compiler. A major limitation for on-call use is that correlative images are rarely available in digital form—the majority of our on-call studies are ventilation–perfusion studies, requiring comparison with a chest radiograph. Slomka et al. report that interpretation based on a verbal description of the radiograph from an in-house radiologist has sometimes been performed, but is not always satisfactory, and methods of converting the radiograph for digital transmission are being investigated (personal communication, May 1999).

Standardization of Java implementations will help substantially in creating a robust programming environment. As the language has evolved, Sun Microsystems has both added new features and made other features obsolete, making it difficult to maintain working programs. Slomka et al. have appropriately insisted that their users install a specific version of the JVM (Java 1.2, as contained in Netscape 4.5 or greater; Netscape Communications Corporation, Mountain View, CA) to simplify program development (personal communication, May 1999). Sun Microsystems has recognized this problem and has stated that it will work on refining the current version of Java 1.2 into a version 2.0, rather than continually adding new features (*5*). Unfortunately, specification of Java as a national or international standard has been hindered by disputes between computer companies as to who will control the new standard (*21,22*).

Slomka et al. appropriately insist that a 24-bit monitor be used for viewing images. Such a monitor allows millions of colors (and a full range of grays) to be displayed. Some users may have a more limited 8-bit display, which allows only 256 colors to be displayed. On 8-bit systems, web browsers (such as Netscape) allocate a fixed color table that includes only 6 levels of gray (including pure black and pure white), with the rest being used for other colors (*23*). Intermediate gray levels will be displayed using dithering, alternating patterns of these limited values to simulate a greater range and significantly degrading the image in the process.

One of the best alternatives to Java for a flexible, inexpensive PACS solution would be to make use of a web browser's capability of passing downloaded files to user-configurable "helper" applications, shown as the last option in Table 1. The web interface could be used for patient file selection and downloading, with automatic handoff of the downloaded file to a radiology image viewing program. The medical image viewing program Osiris (*24*), designed for viewing of DICOM 3 format images, is available for UNIX, Macintosh, and Windows platforms free of charge. Another freely available nuclear medicine image viewing program for the Macintosh is NucMed_Image (*25*). Although this approach would involve manual installation of the viewing program on each user's computer, it otherwise has many of the advantages of a Java-based solution and has been successfully used at a major medical center (*26*).

With regard to file formats, the Interfile standard for nuclear medicine image file format has many advantages: It is easy to understand, human readable (and editable) in a text editor, and widely available among nuclear medicine vendors (*27*). Slomka et al. take advantage of this well-designed format to transmit images to their Java viewer. For better or worse, however, radiology PACS are moving toward the vastly more complex DICOM standard (*28,29*). DICOM includes the ability for patient file selection and requesting transmission of image data. It also includes provision for image compression (discussed below) using the Joint Photographic Experts Group (JPEG) standard, and work is underway to add capability for wavelet compression. Java viewers for DICOM images have already been developed by other investigators (*12–18*), and it is likely that this combination will see widespread use in teleradiology and PACS during the next decade.

BLAZING BITS

Slomka et al. used a V.90 modem for sending data to a remote site. The V.90 standard has evolved from K56-flex and x2, two earlier competing 56K communication protocols. A V.90 modem is the fastest telephone modem available, translating digital signals into analog (tone) form using frequency modulation (MOdulator DEModulator). Speeds using a V.90 modem are asymmetric, with faster downloads from a server than uploads, reaching download speeds of about 50 kbps (1000 bits/s). Thus, a 1-MB file (8182 KB) would require 164 s, or about 3 min, to download (compression may significantly shorten this time). To achieve this rate, the user must have a good-quality phone line, the user must be located within a few miles of the local telephone company switch box, and one must connect to an Internet service provider (ISP) who has a direct digital telephone connection. Frequently, these optimal conditions are not met, and the rate achieved is slightly slower; the download rate seen by Slomka et al. was about half the theoretic maximum for this type of modem.

Faster transmission technologies are now available (*30*). More reliable connections that are 2–4 times faster can be achieved using an Integrated Services Digital Network (ISDN), an all-digital phone service that can be purchased from the telephone company. Even faster is the newly introduced asymmetric digital subscriber line (ADSL), which can reach speeds 10–50 times that of modems and allows concurrent use of voice telephones over the same line using a splitter device. Again, location within a few mile proximity of the telephone switch box is necessary for maximal data transmission rates. Cable modems, using lines used for cable television, have also become available in selected regions. Cable modem speeds may be faster than that of ADSL, provided there is not much demand from other users on the shared cable segment. Approximate speeds of these modalities are shown in Table 2, although different companies claim various data rates. Individual experience will also be a function of other links in the chain, such as the speed of the ISP's Internet connection and the rate at which data are being served by the sending computer. Competition between ADSL and cable modems is driving the price of a rapid Internet connection to an affordable level, while providing service comparable with a dedicated business Internet connection (T1 line).

THE BIG SQUEEZE

Data compression can significantly reduce image file sizes, thereby shortening transmission times. Image compression can either be lossless, in which the uncompressed image is identical to the original image, or lossy, in which (hopefully minor) image degradation occurs as part of the compression process. Slomka et al. used gzip (*31*) compression (personal communication, May 1999), which is a lossless general-purpose compression algorithm that can be applied to both image and nonimage data. Use of such a general file compression technique allowed them to send the original 16-bit image file, complete with patient header information.

Image-specific compression techniques are also available (*32*). Graphics Interchange Format (GIF) images achieve about 2- to 4-fold compression of 8-bit image data using a lossless technique. JPEG format allows variable degrees of compression of image

TABLE 2
Relative Speeds and Features of Internet Connections

Method	Speed* (download)	Speed (upload)	Connection type	Connection to	Simultaneous phone use?
V.34 modem (33.6 modem)	28.8–33.6 kbps	Same	Need to dial	Any ISP or computer	No
V.90 modem (56K modem)	40–50 kbps	28.8–33.6 kbps	Need to dial	Any ISP (connections to other computers at V.34 rate)	No
ISDN	64–128 kbps	Same	Need to dial	Any ISP or computer	Yes
ADSL	384 kbps–1.5 Mbps†	128 kbps†	Always on	One ISP	Yes
Cable modem	500 kbps–10 Mbps‡	64–768 kbps§	Variable§	One ISP	Variable§
T1 (DS1)	1.5 Mbps	Same	Always on	Internet	N/A

*Speeds are listed in kbps (1000 bits/s) and Mbps (1,000,000 bits/s); if upload speed to server is different, a separate upload value is listed.
†Faster ADSL configurations, 1.5–6 Mbps download and 384 kbps upload may be available in certain regions, at higher cost.
‡Actual cable modem speed may be slower, depending on number of other local cable users as a result of shared bandwidth. Upper limit of 30 Mbps is sometimes claimed for cable modems, but connection to user's computer is typically by 10-Mbps ethernet card.
§Some cable modems have no upload capability and require simultaneous use of slower V.90 or V.34 modem over regular phone dial-up connection to provide upload link.

data, trading off accuracy for file size using a lossy technique (dividing the image into 8 × 8 pixel blocks and applying the discrete cosine transform with subsequent partial elimination of high-frequency components). JPEG compression is typically applied to 8-bit data, but a 12-bit medical JPEG variation is also available.

It is difficult to know what level of lossy compression is acceptable for diagnostic purposes. Preliminary studies suggest that 10- to 20-fold JPEG compression is acceptable in medical imaging, varying somewhat with imaging modality (*33,34*). Only limited information is available with regard to compression of nuclear medicine studies, but it appears that similar compression will be achievable (*10,35*). Another technique using wavelets also divides the image into low- and high-frequency components but operates on the image as a whole rather than in 8 × 8 pixel blocks. For this reason, it does not yield the blocky pattern seen in highly compressed JPEG images. Wavelet compression appears to yield slightly higher compression ratios at equivalent image quality, possibly reaching 40:1 or 80:1 for certain radiographic applications (*36,37*).

THE FUTURE

The availability of ready access to the Internet, the relatively urgent need for digital transmission of images through large health-care systems, and the maturing of file format and compression standards will all spur the growth of PACS in radiology. Java provides an inexpensive and easy-to-use method of providing images to users at a wide variety of sites and will likely be used both for on-call viewing and for providing images to referring clinicians.

Jerold W. Wallis
Mallinckrodt Institute of Radiology
St. Louis, Missouri

REFERENCES

1. Slomka PJ, Elliott E, Driedger AA. Java-based remote viewing and processing of nuclear medicine images: toward "the imaging department without walls." *J Nucl Med.* 2000;41:111–118.
2. Gosling J, McGilton H. The Java language environment: a white paper. Available at: http://java.sun.com/docs/white/index.html. Accessed November 6, 1999.
3. Campione M, Walrath K, Huml A. The Java tutorial: a practical guide for programmers. Available at: http://java.sun.com/docs/books/tutorial/. Accessed November 6, 1999.
4. Rodgers RP. Java and its future in biomedical computing. *J Am Med Inform Assoc.* 1996;3:303–307.
5. Levin R. Java technology enters the next dimension. Available at: http://java.sun.com/features/1998/11/jdk.html. Accessed November 6, 1999.
6. Anderson JE, Umans C, Halle M, et al. Anatomy browser: Java-based interactive teaching tool for learning human neuroanatomy. *RSNA ej* [serial online]. 1997;2. Available at: http://ej.rsna.org/ej2/0050–97.fin/index.html. Accessed November 6, 1999.
7. Bellon E, Wauters J, Fernandez-Bayo J, et al. Using WWW and Java for image access and interactive viewing in an integrated PACS. *Med Inform (Lond).* 1997;22:291–300.
8. Hooper P, Fulham M. NM TeleWEB: a picture archiving system for nuclear medicine using internet technologies [abstract]. *J Nucl Med.* 1997; 38(suppl):311P.
9. Lynn L. Radiographic image processing with Java. *RSNA ej* [serial online]. 1997;1. Available at: http://ej.rsna.org/EJ_0_96/0036–97.fin/xrayImg.html. Accessed November 6, 1999.
10. Phung NX, Wallis JW. An internet-based, interactive nuclear medicine image display system implemented in the Java programming language. *RSNA ej* [serial online]. 1998;2. Available at: http://ej.rsna.org/ej2/0072–98.fin/scinternet-e.html. Accessed November 6, 1999.
11. Truong D, Huang S. Image analysis through the World Wide Web [abstract]. *J Nucl Med.* 1997; 38(suppl):309P.
12. Wendt RE III, Hazle JD, Schomer DF, Podoloff DA. Demonstration of a world-wide-web-based viewer for DICOM-formatted medical images [abstract]. *J Nucl Med.* 1997;38(suppl):310P.
13. Barbero O, Fernandez J, Rubies C, Sentis M, Valls R. Practical issues of integrating multimodality and multivendor DICOM equipment: independent tools and RAIM, a self-developed C++ and Java DICOM viewer [abstract]. *Radiology.* 1998;209(P):676.
14. Eng J. Improving the interactivity and functionality of web-based radiology teaching files with the Java programming language. *RadioGraphics.* 1997;17:1567–1574.
15. Fernandez-Bayo J, Barbero O, Rubies C, Sentis M. Distributed medical images using internet technologies: a DICOM web server and a DICOM Java viewer [abstract]. *Radiology.* 1998;209(P):689.
16. Henri CJ, Rubin RK, Cox RD, Bret PM. Design and implementation of World Wide Web-based tools for image management in computed tomography, magnetic resonance imaging, and ultrasonography. *J Digit Imaging.* 1997;10:77–79.
17. Lee SC, Wu LC, Liu RS. DICOM-based web PACS system [abstract]. *J Nucl Med.* 1999;40(suppl):323P.
18. Mikolajczyk K, Szabatin M, Rudnicki P, Grodzki M, Burger C. A JAVA environment for medical image data analysis: initial application for brain PET quantitation. *Med Inform (Lond).* 1998;23:207–214.
19. Web-Link-Medical. n-vision: a web Java-based image viewer. Available at: http://www.weblinkmedical.com/web.htm. Accessed November 6, 1999.
20. Dagher AP, Fitzpatrick M, Flanders AE, Eng J. Enhancing Web applications in radiology with Java: estimating MR imaging relaxation times. *RadioGraphics.* 1998;18:1287–1293.
21. Babcock C. Sun to bear Java standard? Available at: http://www.zdnet.com/zdnn/content/zdnn/1114/241776.html. Accessed November 6, 1999.
22. Associated Press. Judge rules for Sun over Microsoft. *New York Times* [serial online]. Available at: http://www.nytimes.com/library/tech/99/05/biztech/articles/26sun.html. Accessed November 6, 1999.
23. Engel V. No Dither Netscape Color Palette. Available at: http://the-light.com/netcol.html. Accessed November 6, 1999.
24. Ligier Y, Ratib O, Logean M, Girard C. OSIRIS imaging software. Available at: http://www.expasy.ch/UIN/html1/projects/osiris/osiris.html. Accessed November 6, 1999.
25. Wittry MD, Farris JS, Lowe VJ, Fletcher JW. NucMed_Image as a nuclear medicine work station. Available at: http://165.134.33.50/NucMed_Image/NucMed_Image.html. Accessed November 6, 1999.
26. Barbaras L, Parker JA, Donohoe KJ, Kolodny GM. The all-digital department moves to the Web. *RSNA ej* [serial online]. 1996;1. Available at: http://ej.rsna.org/EJ_0_96/0006–96/home.htm. Accessed November 6, 1999.
27. Todd-Pokropek A, Cradduck TD, Deconinck F. A file format for the exchange of nuclear medicine image data: a specification of Interfile Version 3.3. *Nucl Med Commun.* 1992;13:673–699. Available at: ftp://ftp.largnet.uwo.ca/pub/nucmed/Interfile/interfile33.complete. Accessed November 6, 1999.
28. Clunie DA. DICOM standard status. Available at: http://idt.net/~dclunie/dicom-status/status.html. Accessed November 6, 1999.
29. Horii SC. A nontechnical introduction to DICOM. *RadioGraphics.* 1997;17:1297–1309.
30. Beckman M. Modems' last stand. *MacWorld* 1999; May:86. Available at: http://macworld.zdnet.com/1999/05/features/modems.html. Accessed November 6, 1999.
31. Leininger KE. *UNIX Developers Tool Kit.* New York, NY: McGraw-Hill; 1994:305–309.
32. Murray J, VanRyper W. *Encyclopedia of Graphics File Formats.* Sebastopol, CA: O'Reilly & Associates; 1994.
33. Persons K, Palisson P, Manduca A, Erickson BJ, Savcenko V. An analytical look at the effects of compression on medical images. *J Digit Imaging.* 1997;10:60–66.
34. Baker WA, Hearne SE, Spero LA, et al. Lossy (15:1) JPEG compression of digital coronary angiograms does not limit detection of subtle morphological features. *Circulation.* 1997;96:1157–1164.
35. Rebelo MS, Furuie SS, Munhoz AC, Moura L, Melo CP. Lossy compression in nuclear medicine images. *Proc Annu Symp Comput Appl Med Care.* 1993:824–828.
36. Erickson BJ, Manduca A, Palisson P, et al. Wavelet compression of medical images. *Radiology.* 1998; 206:599–607.
37. Erickson BJ, Manduca A, Persons KR, et al. Evaluation of irreversible compression of digitized posterior-anterior chest radiographs. *J Digit Imaging.* 1997;10:97–102.