Java-Based Remote Viewing and Processing of Nuclear Medicine Images: Toward "the Imaging Department Without Walls"

Piotr J. Slomka, Edward Elliott, and Albert A. Driedger

Departments of Diagnostic Radiology and Nuclear Medicine and Electrical and Computer Engineering, University of Western Ontario, London, Ontario, Canada

In nuclear medicine practice, images often need to be reviewed and reports prepared from locations outside the department, usually in the form of hard copy. Although hard-copy images are simple and portable, they do not offer electronic data search and image manipulation capabilities. On the other hand, picture archiving and communication systems or dedicated workstations cannot be easily deployed at numerous locations. To solve this problem, we propose a Java-based remote viewing station (JaRViS) for the reading and reporting of nuclear medicine images using Internet browser technology. **Methods:** JaRViS interfaces to the clinical patient database of a nuclear medicine workstation. All JaRViS software resides on a nuclear medicine department server. The contents of the clinical database can be searched by a browser interface after providing a password. Compressed images with the Java applet and color lookup tables are downloaded on the client side. This paradigm does not require nuclear medicine software to reside on remote computers, which simplifies support and deployment of such a system. To enable versatile reporting of the images, color tables and thresholds can be interactively manipulated and images can be displayed in a variety of layouts. Image filtering, frame grouping (adding frames), and movie display are available. Tomographic mode displays are supported, including gated SPECT. **Results:** The time to display 14 lung perfusion images in 128 × 128 matrix together with the Java applet and color lookup tables over a V.90 modem is <1 min. SPECT and PET slice reorientation is interactive (<1 s). JaRViS could run on a Windows 95/98/NT or a Macintosh platform with Netscape Communicator or Microsoft Internet Explorer. The performance of Java code for bilinear interpolation, cine display, and filtering approaches that of a standard imaging workstation. **Conclusion:** It is feasible to set up a remote nuclear medicine viewing station using Java and an Internet or intranet browser. Images can be made easily and cost-effectively available to referring physicians and ambulatory clinics within and outside of the hospital, providing a convenient alternative to film media. We also find this system useful in home reporting of emergency procedures such as lung ventilation–perfusion scans or dynamic studies.

Key Words: telemedicine; picture archiving and communication systems; Java

J Nucl Med 2000; 41:111–118

Received Sep. 29, 1998; revision accepted May 4, 1999.

For correspondence or reprints contact: Piotr Slomka, PhD, Nuclear Medicine Department, London Health Sciences Center, 375 South St., London, Ontario N5A4G5, Canada.

In nuclear medicine practice, situations often arise in which images have to be reviewed and reported on from locations outside the department. Typically, a hard copy of the image is used for that purpose, and, although simple and portable, its delivery is time consuming and it does not offer electronic data search and image manipulation capabilities offered by picture archiving and communication systems (PACS) stations (*1*) or dedicated imaging workstations. Moreover, films cannot be shared simultaneously among multiple reviewers and may be lost between imaging departments and referring centers. In addition, certain diagnostic parameters cannot be represented on film and hard copy. For example, gated SPECT scans must usually be viewed in an interactive fashion, adjusting particular tomographic planes and providing a cinematic mode of display.

On the other hand, PACS workstations cannot be easily deployed at numerous locations, because they are fairly expensive and require dedicated hardware for image display, mainly to satisfy radiologic requirements. Dedicated nuclear medicine workstations are typically too complex for the casual reviewer (e.g., the referring physician), and although they use a standard computer platform, they often require the UNIX operating system or have other advanced requirements (*2*). Even if the nuclear medicine workstation operated on a standard office personal computer, this machine would require considerable software setup and maintenance. In addition, licensing costs for nuclear medicine workstations can be prohibitive.

Recently, a new software paradigm has been developed by Sun Microsystems (Mountain View, CA), who introduced the Java programming language (*3*), which, among other features, allows remote execution of applets (small programs or applications) by World Wide Web browsers. Such applets can be combined with hypertext markup language (HTML) documents to provide interactive content in web pages. Java applets are downloaded on the fly from the server in an executable form called "bytecode." Current web browsers, running on ubiquitous office computers such as a personal computer with Windows95/NT (Microsoft,

Redmond, WA) or Macintosh (Apple Computer, Cupertino, CA), support the execution of such bytecode. Conceivably, Java applets residing on the main imaging server could be used for on-line distribution of medical images and contain the code to perform interactive image analysis (*4–8*). A related concept has been introduced for the development of teaching files (*9*).

We propose to use the web-based imaging system to provide on-line capabilities for the daily remote review of nuclear medicine studies, including typical image processing functions and viewing modes. We evaluate the feasibility of such an approach and contrast it with the standard PACS architecture design for nuclear medicine. We also describe our software prototyping experience in using Java for developing a Java-based remote viewing station (JaRViS) and discuss practicality and potential application of such a system.

The rationale for the development of such a system is to create the "department without walls," where diagnostic images would be made routinely available to the referring physicians involved in patient care without equipping them with dedicated imaging workstations or installed software. This is important if we want to combine the medical experiences of physicians involved in patient care and effectively communicate nuclear medicine findings (*10*). We aim at providing such a solution at a fraction of the cost and complexity associated with current PACS and extending it outside of the hospital to physicians' homes or other locations where PACS access is not practical.

METHODS

The JaRViS system has been implemented in Java as a set of applets and Common Gateway Interface (CGI) scripts. Applets have been implemented in Java version 1.1 (*11*).

The overall design of the system is presented in Figure 1. We implemented a direct interface to the clinical patient database on a Hermes Nuclear Diagnostics (Stockholm, Sweden) workstation. This station uses the Interfile (*12*) and Digital Imaging and Communication in Medicine (DICOM) formats for image storage. The patient search module is written in JavaScript and HTML. The search of the patient database on the server is implemented using the Perl 5.04 language (*13*). The user name and password are provided using standard HTTP security mechanisms. The patient database can be searched by several key words such as patient name, study type, and acquisition dates (Fig. 2). In addition, the login name can be matched with the "doctor" header field to provide selective access to patient studies for referring physicians. Access to the database is direct; there is no need for any operator steps on the workstation side.

A selected patient study is prepared for downloading by compressing the raw image data. This is done using a lossless compression algorithm available in Java 1.1 implementation (*11*) Subsequently, the compressed nuclear medicine image series together with the server color lookup tables (31 lookup tables each with 256 entries) and the Java applet bytecode for interactive image viewing and processing are bundled into a Java archive (JAR) file and downloaded to the web browser. The JAR file format is similar to the popular ZIP data format (*11*). It allows for combining of data files, which include Java applets, color lookup tables, and compressed image data. Using this format, the web browser makes only one connection to the server and obtains all of the relevant files needed to interact with the chosen study. JaRViS applets include common nuclear medicine operations such as frame grouping (adding images), movie display (up to 512 × 512 size), interactive color table manipulation, threshold manipulation, spatial (3 × 3 kernel) and temporal filtering, and tomographic operations including gated tomography display. We have created several applets for different study types (static, dynamic, whole body, tomographic PET/SPECT display, and gated tomography) that are customized to provide the display format suited for each type of study. Examples of the display in JaRViS are provided for planar (Fig. 3), PET (Fig. 4), and gated SPECT studies (Fig. 5). After the selection of the

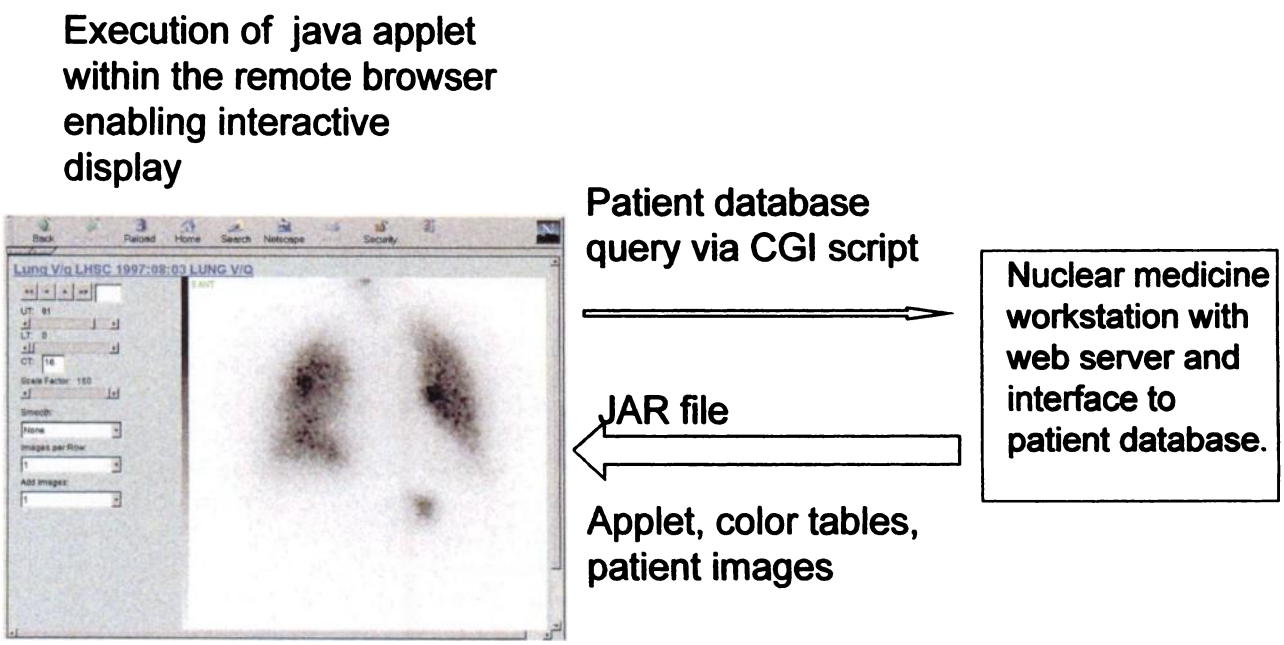

FIGURE 1. Design of JaRViS.

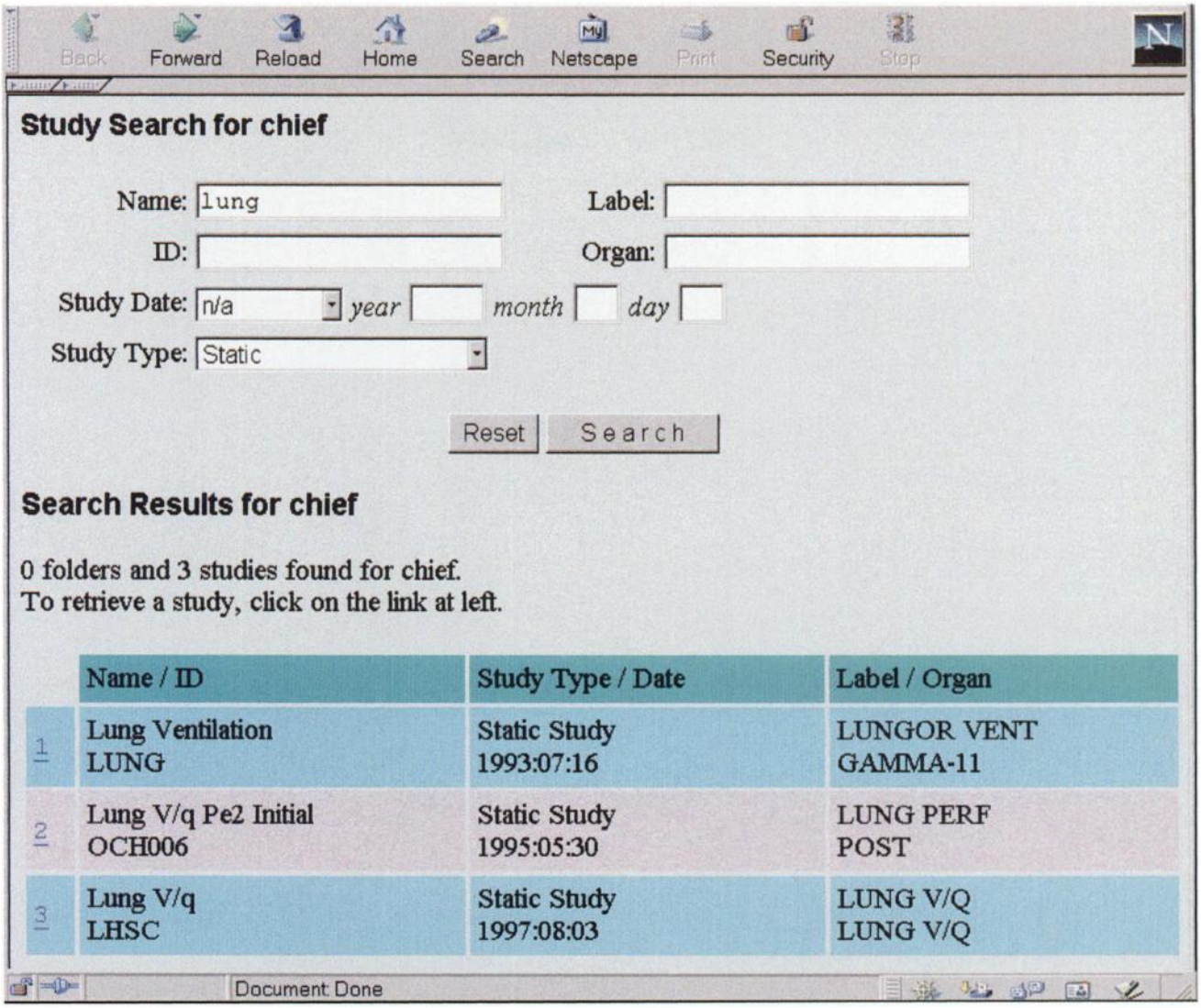

FIGURE 2. HTML user interface for patient study search.

study type, the appropriate applet is downloaded from the server. Applet sizes ranged from 14 KB (planar display) to 50 KB (gated tomography), reflecting the complexity of the required operations.

Because the Java programming language supports object-oriented software design and its syntax is similar to the C/C++ programming language, we were able to port efficiently existing image manipulation routines written in C/C++ to Java.

We tested the remote applet execution on a personal computer with a Pentium 233 MMX with Windows 98 running Netscape Communicator 4.5 (Netscape Communications Corporation, Mountain View, CA) and Internet Explorer 4.01 and on a Macintosh iMac running Internet Explorer 4.01. For better performance, we used just-in-time (JIT) compiler options in these browsers. JIT compiler significantly speeds up the execution of applets by converting Java bytecodes to the native machine code on the fly as the applets are downloaded, but it is not available on all platforms (*14*).

RESULTS

The average time to display 14 lung ventilation and perfusion images in 128 × 128 matrix (448-KB raw data plus 8-KB header, 110 KB after lossless compression), with Java bytecode (14 KB) and color lookup tables (18 KB), over a V.90 (56-kbps) US Robotics modem (3Com, Santa Clara, CA) was 55 s. The lossless compression ratio varied from 2:1 to 5:1 depending on the type of study. This ratio reflects the gain in transfer speed as a result of compression. The average download speed of compressed data was 4.5 KB/s. The total time included the 11 s of Java startup time on the viewing computer. After the study is downloaded from the server, the modem can be disconnected and the analysis and review can be performed without further communication with the server. The transfer time over a local area network is <1 s.

We tested the download time with and without using JAR file format (but with images compressed in both cases). Without using the JAR file format for transferring study data from the server, the transfer and startup time over the V.90 (56-kbps) modem was 59 s, and 3 connections were made to the server (to separately download compressed images, color tables, and the applet). In contrast, using the JAR file

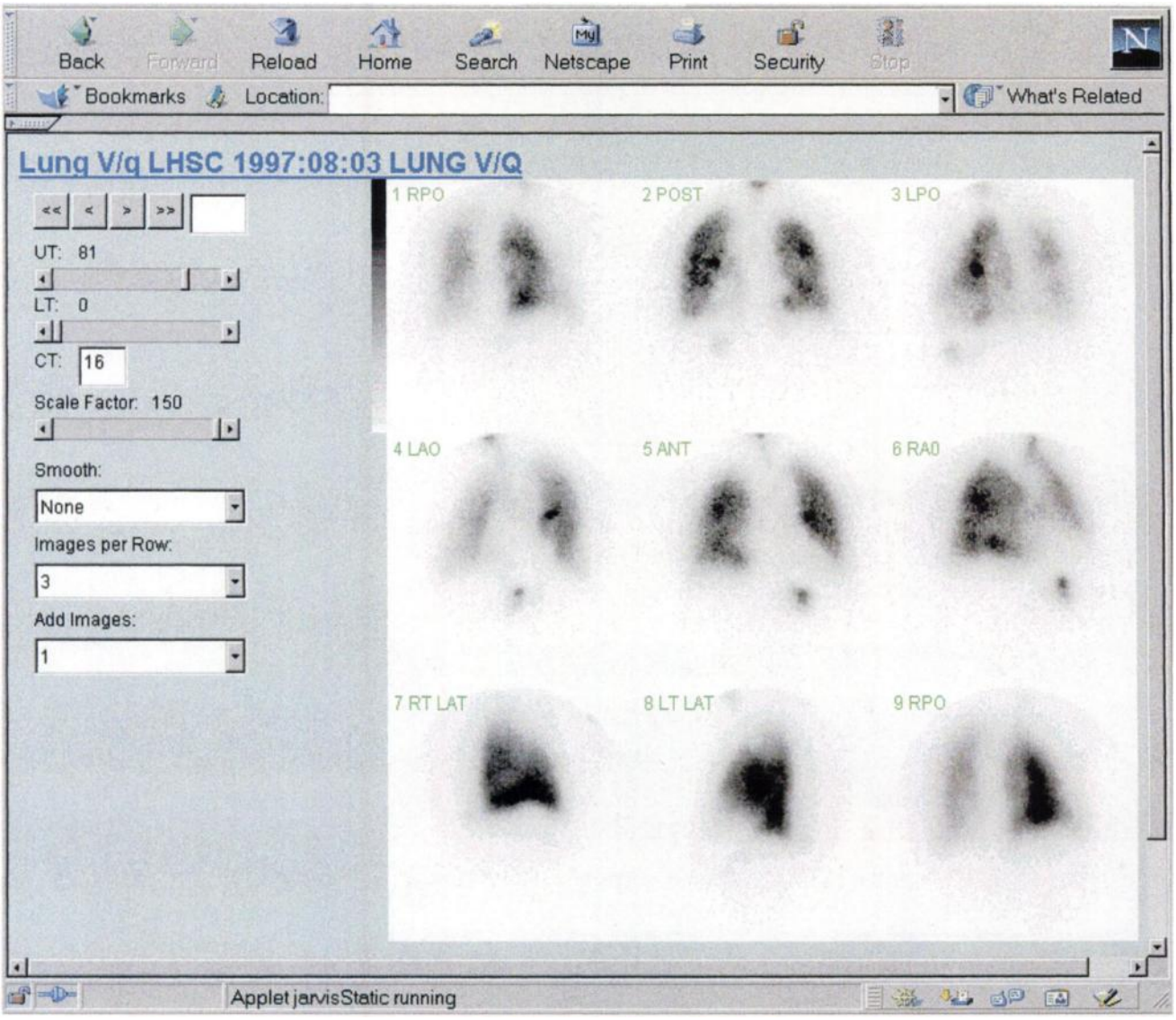

FIGURE 3. User interface and image display in applet for viewing static images. Elements of user interface include 3 × 3 smoothing filter, image size, adding individual images (add images), image rescaling, and color table thresholds.

format, the transfer time was reduced by 4 s because only 1 connection to the server was made, downloading the data, applet, and color tables as 1 file. On a local area network the difference in transfer times was negligible.

The time to spatially filter the 128 × 128 × 63 PET dataset using the 3 × 3 kernel takes <1 s. Bilinear interpolation, threshold of lookup tables for visual presentation of images, and cine display could be performed interactively in a 512 × 512 image window using the JIT compiler available with Netscape and Internet Explorer.

In tomographic applications, the display of 3 orthogonal PET or SPECT 128 × 128 slices including trilinear interpolation and slice angle adjustment (Fig. 4) could be performed interactively (<0.5 s). This feature enables on-the-fly angle adjustment on the client side. MR images (256 × 256 matrix; 20 slices) were converted into Interfile from DICOM format (*15*) using the dicom2if program (Hermes, Nuclear Diagnostics) and could also be viewed in JaRViS; the time for volume reslicing in 3 orthogonal planes was <2 s. In gated SPECT display, tomographic views could be adjusted interactively and a simultaneous movie of 5 different tomographic slices could be played with a rate > 20 frames/s. To allow the fast interactive adjustment of the color thresholds, the multithreading mechanism of Java was used (*11*).

In principle, Java bytecode should be able to run on all computer platforms supporting Java Virtual Machine (*3*). We have tested JaRViS on the following platforms: Windows 95/98 and NT (Netscape, Internet Explorer) and Macintosh (Internet Explorer). We experienced some small changes in program behavior, such as different than specified font sizes. Hopefully in the future full Java implementation will be guaranteed to perform correctly and identically on all the platforms. The faithful display of at least 200

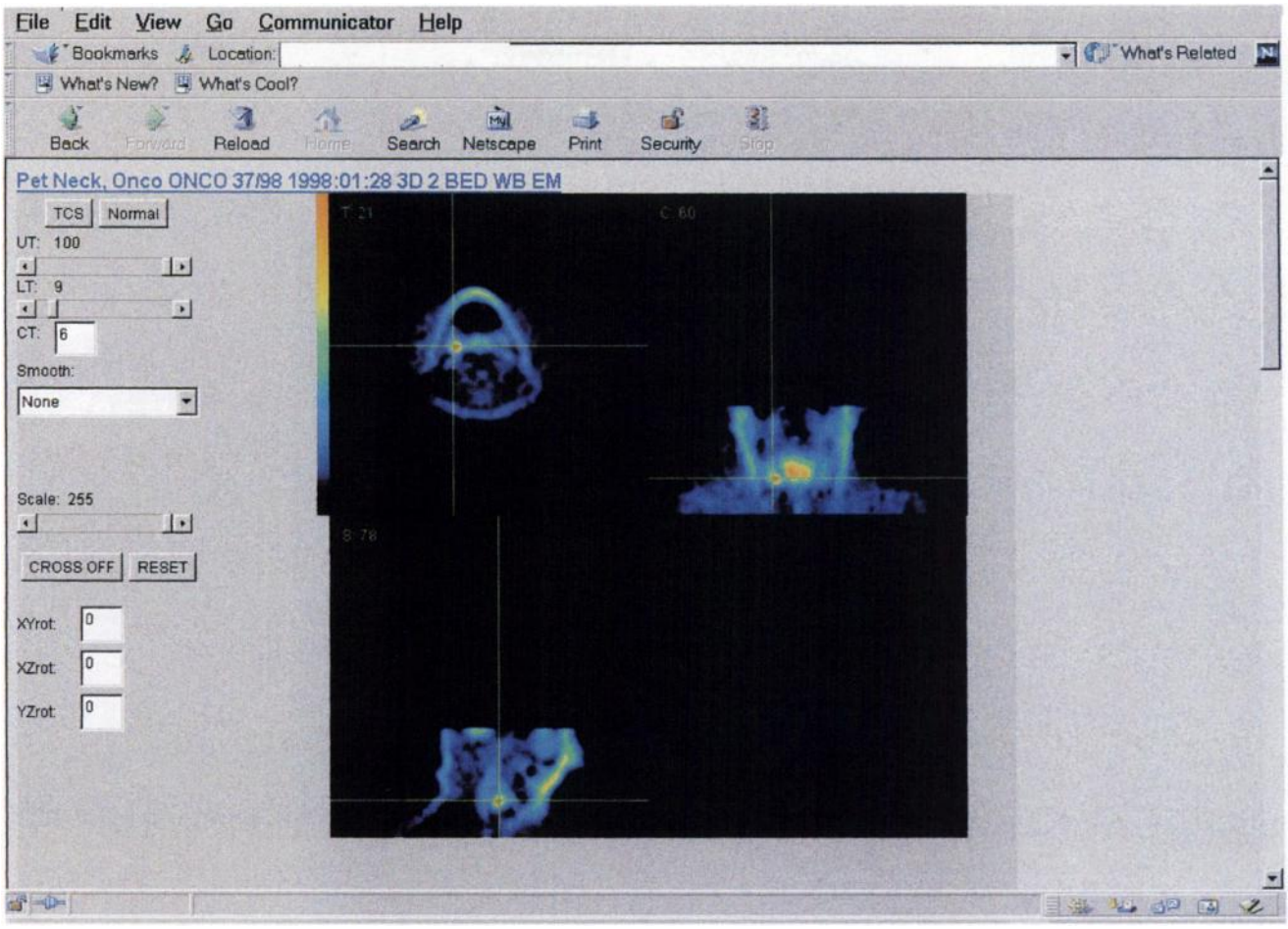

FIGURE 4. Interactive tomographic PET/SPECT viewing applet in TCS (3 orthogonal views) mode. Both cursor position and 3-dimensional alignment can be adjusted interactively. Only transverse dataset is used by applet and other views are derived on the fly. Original tomographic slices are displayed in "normal" mode.

colors in the lookup table required the use of at least 24-bit display depth on these computers; this is because of the color allocation mechanism by the web browsers. For applet demonstration, connect to http://www.irus.rri.on.ca/~pslomka/jarvis.

DISCUSSION

The described solution to access nuclear medicine images has several advantages. No software is required on the viewing site, thus images can be made easily and cost-effectively available to referring physicians and ambulatory clinics within and outside of the hospital, providing a viable alternative to film media and to dedicated review stations. In our limited experience, this has already proven to be a very important feature. We were able to use the described system on a variety of remote computers without any installation procedures. This may prove of critical importance when a large number of referring clinicians would like to reliably access the images from various locations. The described setup ensures that everyone uses the same version of the software and that differences in configurations and type of remote computers are of no consequence.

The advantage of the proposed image distribution system compared with the film media traditionally used is especially significant when presenting tomographic or dynamic data. Routinely all slices must be shown in all orientations, resulting in hard-copy images that are difficult to read because of the large number of slices. Also, it is often helpful to view images in coordinated 3 orthogonal planes (*16*) and to readjust the slice angles in all orientations while viewing the images. This kind of display mechanism is not possible with traditional hard-copy media. Furthermore, high-quality color hard copy, which is often needed in nuclear medicine, is currently unreliable and expensive. Such limitations are not present in computer-based displays, and, therefore, these are recommended, especially for the presentation of tomographic images (*17*). Cine display is needed to view dynamic data, SPECT projections, or gated tomography. Furthermore, most types of images require interactive contrast or color table adjustment.

The main reason for selection of the Java language for implementation of remote viewing and processing was the possibility of execution of applets on remote computers without any specific software configuration or other require-

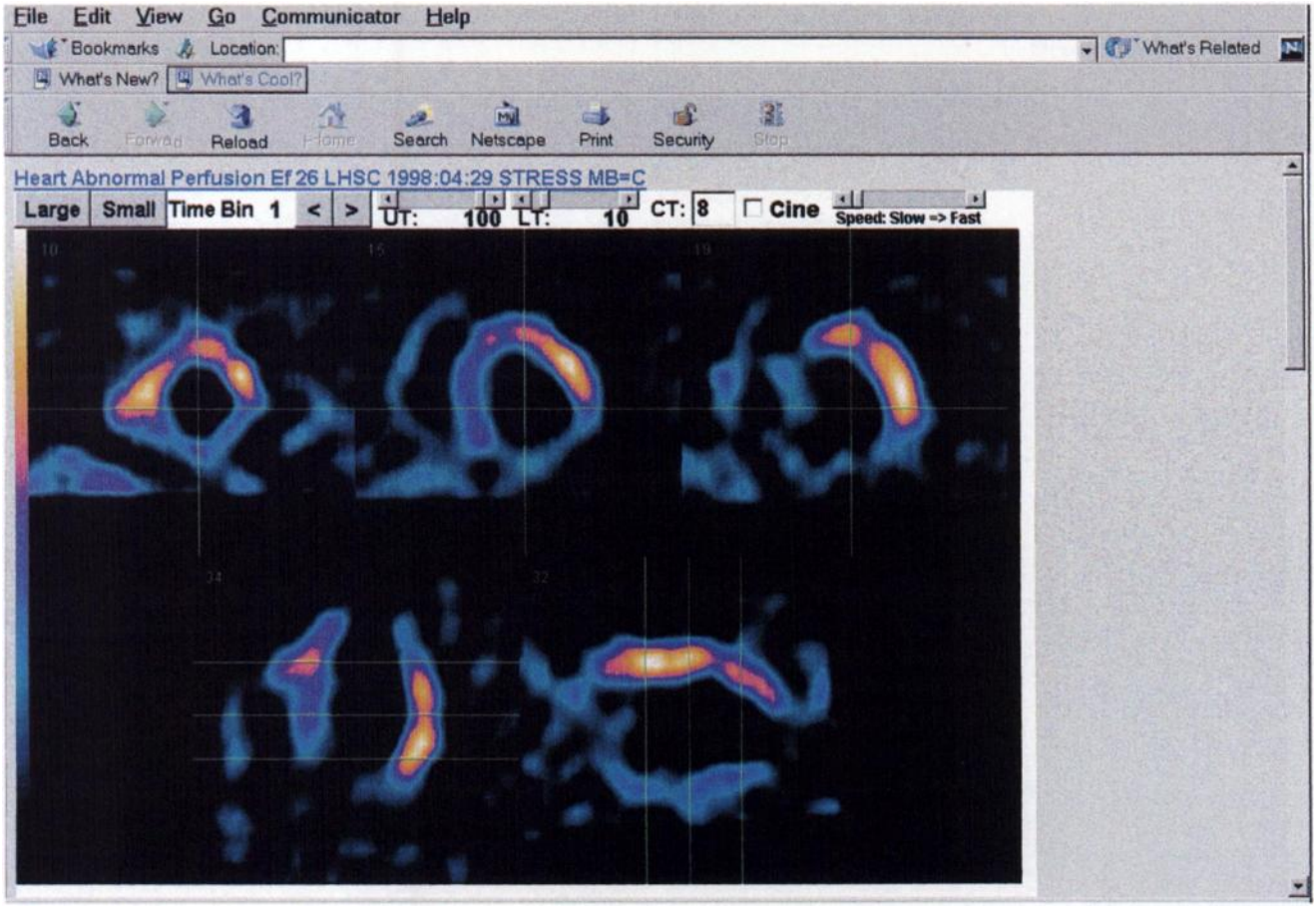

FIGURE 5. Gated tomography applet. These views can be shown in cinematic mode, and reference lines can be adjusted interactively. CT designates color table number. Color table numbers are identical to those on a nuclear medicine workstation.

ments, using standard web browsers. In addition, Java is available on all major computer platforms and is nonproprietary. There are other software mechanisms for remote display, such as X-windows (*18*), that provide remote display in the UNIX environment but the execution of such programs is performed on the server and images are constantly updated with intermediate display results computed on the server. Therefore, an X-window display cannot be used over slow modem lines. Another possible platform, ActiveX, is a proprietary system developed by Microsoft and therefore is available only on Microsoft operating systems, which is its primary disadvantage compared with Java.

Several security issues are also well addressed in Java. It is designed with the industrial-strength security features needed, for example, in web-based banking systems. The secure version of the HTTP protocol (HTTPS) that is based on secure socket layer architecture (*19*) can be used with all standard browsers. This provides encryption of the transferred images and patient information, which will be perhaps mandatory in a clinical environment. The multiple login features facilitate granting selective access to patient images on a per-study basis. In addition, other security issues such as spread of computer viruses are well addressed in Java because the executing applet cannot interact with the viewing computer files and operating system (*3*).

Such distributed image viewing functionality overlaps with that offered by a fully fledged PACS (*1*), in which medical images can be viewed on display stations throughout the institution. Currently, PACS require dedicated viewing hardware, software licenses, and maintenance support for each display station. A Java-based system like JaRViS, however, could potentially provide similar functions using standard computers already present in many locations in a hospital setting, thereby drastically reducing the cost of such a solution and simplifying the training for the use of such a system (*7*). This solution appears to be practical for nuclear medicine remote viewing. Indeed, a more controversial view could be that this type of system could be used for general radiology PACS purpose—perhaps with higher requirements for the display graphics board and monitors on the viewing side. The results obtained in this implementation suggest that interactive viewing of data such as CT or MRI could also be accomplished. This type of approach has been already reported for the presentation of radiologic images (*8*).

The cost-savings aspect of the proposed solution is especially important, taking into consideration the rapid depreciation of computer hardware and the dramatically increasing capabilities of standard computer equipment. Thus, a PACS designed and implemented on a high-end workstation may in fact be inferior to the latest off-the-shelf

office computer systems. In this situation it becomes important to use existing computer infrastructure rather than investing in hardware for the review stations, which need frequent upgrades. The paradigm of sending the viewing application together with the image data also has the benefit of being less likely to be outdated, if new functionality in the data display was required. A new type of data presentation would require a costly software upgrade of the whole network of PACS, whereas, in the Java-based approach, only an upgrade at the source of the data would be required.

In addition to the use of such a system as a component in the hospital PACS, an immediate practical application for JaRViS is home reporting of emergency procedures such as lung ventilation–perfusion scans or dynamic studies. We found the speed of download over a telephone line modem sufficient for such use. These speeds could be dramatically improved with cable modems (*20*) or other similar technologies.

A natural extension of such a system is the integration of the viewing software with image reporting software, in which patient reports could be stored in HTML format, including captured images and animation. This function would be somewhat equivalent to the DICOM screen capture concept (*15*), but more powerful because animations and text reports could be also included. The benefit of using the HTML format (perhaps encapsulating the DICOM definitions of medical terms) would be the simplification of the software needed to review such reports. A similar concept has been pursued by other investigators (*21*).

An often discussed issue is the performance of the code implemented in the Java programming language (*14*), especially when it involves computationally intensive tasks, such as image filtering or interpolation. In this project, we have found that the performance of Java code for bilinear interpolation, cine display, tomographic reorientation, and filtering approaches that of a standard imaging workstation if a JIT compiler is used. Therefore, we do not perceive the performance of Java to be an obstacle for this application, even if more sophisticated image algorithms are required. Furthermore, dramatic performance improvement will likely occur with the introduction of Java hardware on the viewing computers (*14*).

One issue in the design of a system such as JaRViS is consideration of the standards in the user interface and display lookup color tables. To the best of our knowledge, there is currently no standard specific enough to provide detailed guidelines for "look and feel" and functionality of applications such as gated SPECT display or tomographic image manipulations. However, as soon as such standards emerge, Java-based remote display should conform to these specifications. Our current design is to some extent arbitrary and can serve as an example of the implementation. In the design process, we were trying to elicit the most important functions needed to review and process the study and provide those functions with maximum simplicity. The design evolved from a series of discussions with the users of the system.

The available color maps (lookup tables) also raise the issue of standards. In the current design, we copied the color table set from the main server (Hermes, Nuclear Diagnostics) to provide the identical display appearance to that on the main nuclear medicine workstation. This approach could be adapted to other nuclear medicine workstations. However, there is a lack of standardization among vendors in this area and perhaps in the future a standard set of color tables may be available across all vendors. The National Electrical Manufacturers Association recently provided a detailed proposal for the gray scale and display definitions for radiologic images to facilitate reproducible image display (*22*). Such guidelines for nuclear medicine have not been developed. It would be desirable to provide a vendor-independent set of color tables in a system such as JaRViS.

Potential pitfalls in the implementation of the Java-based review system include a certain lack of quality control of the whole display pipeline. Because images are displayed on standard office equipment that are not covered by quality control procedures, there could be a possibility of a faulty or inadequate monitor or graphics card. This could produce subtle differences in comparison with the images displayed on the standard nuclear medicine workstation, which are delivered and tested by the nuclear medicine vendor. Thus, one could have difficulty in guaranteeing the exact image appearance on a remote station. This perhaps would not be a frequent problem, but it could raise questions regarding the responsibility for image quality control and the liability in case the images were not read correctly. One solution to this problem could be the use of certain predefined test patterns, for example, those designed by the Society of Motion Picture and Television Engineers (SMPTE) (*23*) to verify the quality of the image on the remote site. Such images could be placed directly on the JaRViS web page, and the remote user could acknowledge recognizing the test pattern before commencing the clinical review.

CONCLUSION

We propose a cost-effective, simple alternative for reviewing nuclear medicine images using Internet browser technology. The clinical imaging database can be searched by a browser interface, and compressed patient images with the Java applet and color lookup tables are downloaded to the client computer from the departmental server. This paradigm does not require nuclear medicine software to be set up on remote viewing computers, simplifying support and deployment of such stations in comparison with PACS. Images can be interactively manipulated and displayed in a variety of layouts, including tomographic, gated tomographic displays, and dynamic movies, offering significant advantages compared with hard-copy presentation. We conclude that it is feasible and beneficial to set up a nuclear medicine reviewing station using Java and an Internet or intranet browser.

ACKNOWLEDGMENTS

This study was supported by a grant from the Research and Development Fund of Nuclear Diagnostics (Stockholm, Sweden). The authors thank Drs. Michael Chamberlain and Barry McKee from the Ottawa Civic Hospital, Canada, for useful discussions regarding image viewing requirements and Prof. Klaus Tatsch, Department of Nuclear Medicine, University of Munich, Germany, for PET datasets.

REFERENCES

1. Meyer-Ebrecht D. Picture archiving and communication systems (PACS) for medical application. *Int J Biomed Comput.* 1994;35:91–124.
2. Slomka PJ, Dey D. Review of the instrumentation developments at the 1996 annual meeting. *J Nucl Med.* 1996;37(9):25N–33N.
3. Arnold K, Gosling J. *The Java Programming Language.* Reading, MA: Addison-Wesley; 1996.
4. Phung NX, Wallis JW. An Internet-based, interactive nuclear medicine image display system implemented in the Java programming language [abstract]. *J Nucl Med.* 1997;38(suppl):210P.
5. Wittry MD, Farris JS, Fletcher JW. TIFFnet: a web-based PACS for nuclear medicine [abstract]. *J Nucl Med.* 1997;38:307P–308P.
6. Lowe HJ, Lomax EC, Polonkey SE. The World Wide Web: a review of an emerging internet-based technology for the distribution of biomedical information. *J Am Med Inform Assoc.* 1996;3:1–14.
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. 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/xraylmg.html. Accessed November 6, 1999.
9. Parker JA, Wallis JW, Halama JA, et al. Collaboration using internet for development of case-based teaching files. *J Nucl Med.* 1996;37:178–184.
10. Wagner HN. Nuclear medicine: the road to smart medicine and surgery. *J Nucl Med.* 1998;39(8):13N–34N.
11. Zukowski J. *Java AWT Reference (Java 1.1).* Sebastopol, CA: O'Reilly & Associates; 1997.
12. Todd-Pokropek A, Cradduck TD, Deconinck F. A file format for the exchange of nuclear medicine data: a specification of INTERFILE version 3.3. *Nucl Med Commun.* 1992;13:673–699.
13. Wall L, Christiansen T, Schwartz RL. *Programming Perl.* Sebastopol, CA: O'Reilly & Associates; 1996.
14. Gosling J. The feel of Java. *Computer.* 1997;30:53–57.
15. American College of Radiology and National Electrical Manufacturers Association. Digital Imaging and Communications in Medicine (DICOM): Version 3.0, draft standard. Washington, DC: ACR–NEMA Committee, Working Group VI; 1993.
16. Wagner H, Szabo Z, Buchanan J. *Principles of Nuclear Medicine.* Philadelphia, PA: WB Saunders; 1995:338.
17. Keyes JW, Kline RC, Resinger WW, et al. Comparison of observer performance reading from a video CRT versus reading from film. *Invest Radiol.* 1983;18:298–300.
18. Nye A. *X Protocol Reference Manual: Volume Zero for X11, Release 6 (Definitive Guide to X Windows, Vol 0).* Sebastopol, CA: O'Reilly & Associates; 1995.
19. Freier AO, Karlton P, Kocher PC. The SSL protocol version 3.0. internet draft. Available at: http://home.netscape.com/eng/ssl3/ssl-toc.html. Accessed November 6, 1999.
20. Parker JA, Donohoe KJ, Kolodny GM. Cable modem–assisted tele-nuclear medicine. *RSNA ej* [serial online]. 1996;1. Available at: http://ej.rsna.org/EJ_0_96/0019–96.fin/cable_modem.html. Accessed November 6, 1999.
21. 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.
22. American College of Radiology and National Electrical Manufacturers Association. Digital Imaging and Communications in Medicine (DICOM): Version 3.0, draft standard part 14, supplement 28—grayscale standard display function. Washington, DC: ACR–NEMA Committee, Working Group VI; 1998.
23. Gray JE. Use of the SMPTE test pattern in picture archiving and communication systems. *J Digit Imaging.* 1992;5:54–58.