eHealth International

BioMed Central

Research

Experience in the application of Java Technologies in telemedicine

Fedyukin IV[1], Reviakin YG[1], Orlov OI[1], Doarn CR*[2], Harnett BM[2] and Merrell RC[2]

Address: [1]Space Biomedical Center for Training and Research, Moscow, Russia and [2]Virginia Commonwealth University's Medical College of Virginia, Richmond, Virginia USA

E-mail: Fedyukin IV - crdoarn@hsc.vcu.edu; Reviakin YG - crdoarn@hsc.vcu.edu; Orlov OI - crdoarn@hsc.vcu.edu; Doarn CR* - crdoarn@hsc.vcu.edu; Harnett BM - crdoarn@hsc.vcu.edu; Merrell RC - crdoarn@hsc.vcu.edu

*Corresponding author

Published: 17 September 2002

eHealth International 2002, **1**:3

Received: 23 June 2002
Accepted: 17 September 2002

This article is available from: http://www.ehealthinternational.org/content/1/1/3

Abstract

Java language has been demonstrated to be an effective tool in supporting medical image viewing in Russia. This evaluation was completed by obtaining a maximum of 20 images, depending on the client's computer workstation from one patient using a commercially available computer tomography (CT) scanner. The images were compared against standard CT images that were viewed at the site of capture. There was no appreciable difference. The client side is a lightweight component that provides an intuitive interface for end users. Each image is loaded in its own thread and the user can begin work after the first image has been loaded. This feature is especially useful on slow connection speed, 9.6 Kbps for example. The server side, which is implemented by the Java Servlet Engine works more effective than common gateway interface (CGI) programs do. Advantages of the Java Technology place this program on the next level of application development. This paper presents a unique application of Java in telemedicine.

Introduction

Since 1995, when Sun Microsystems announced introduction of new software technology, many industries have taken advantage of Java.[1] Medicine was no exception. Java technologies and capabilities, which includes development and use of client-server applications, offers tremendous benefit for applications in public health and the health care sector.[2] The simple client's work place is provided with the help of a "thin-client" or a network computer. The way information is arranged within the network does not matter to a physician; what matters is how to find patient's records. Functions such as 'Search' will be the job of an application, which has been controlled, debugged and dynamically loaded from the server. In the 1970's, dumb terminals tethered to mainframe computers relied completely on the host to deliver its textual display to the user. Over the next ten years computing power at the individual workstation surged, ushering a new era of network computing. Today, the user's workstation (clients) bears the brunt of the processing responsibility, allowing the host (server) to act primarily as an information repository and system-wide network administrator. As a result of this paradigm shift, these tools enhance remote health care. Work by Padeken, et al [3] and Viirre, et al [4] has further demonstrated this. Integration of these tools permits the exchange of great amounts of information in real-time. Emerging network infrastructures like Internet 2 will provide the ability for large amounts of data to be transmitted.[5]

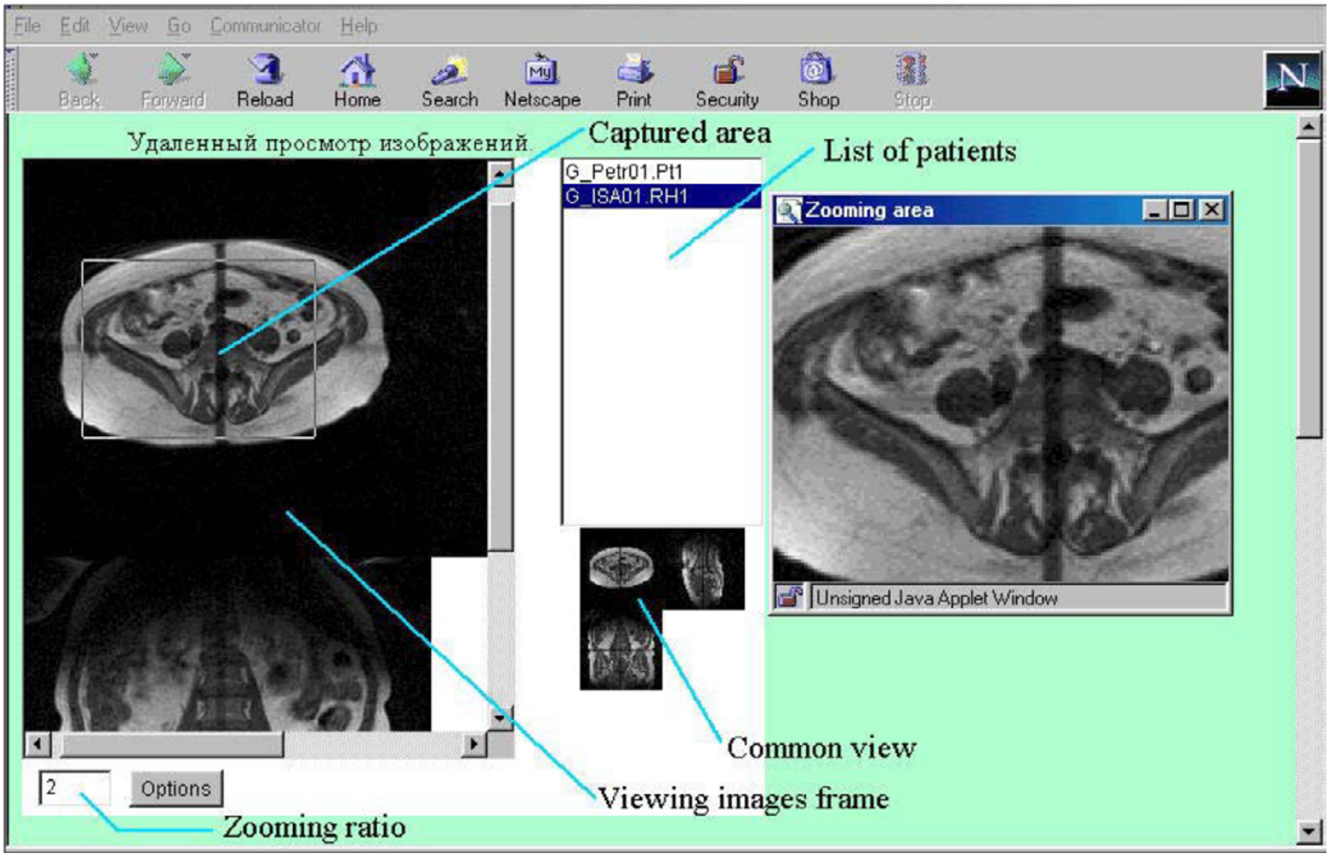


Figure 1
Remote Image View Program Interface

Background

Born out of legacy systems, the Java programming language was designed to build various applications types, which run seamlessly across heterogeneous platforms and take advantage of more capacity at the desktop.[6]

Java applications provide access to information from any networked terminal within or outside a medical institution. In addition, Java offers scalable, distributed access to data and, therefore, permits real time consultations from multiple locations. This approach to the formidable problem of acquiring distributed medical information facilitates more rapid diagnosis and treatment. As a rule, Java software does not require large amount of system resources and is stable within contemporary operating systems.[7] The use of Java technology is attractive because it accommodates program modules of existing system without interference with operation.[8] Thus, medical personnel may pull specific information from various sources with minimal of any computer-based knowledge because Java applications are intuitively designed. Use of this technology does not require profound knowledge of computer design and there is often no need to install the Java program at all points of data retrieval.[9]

SystemSource, Inc.[10] was one of the first to apply the Java technology in medicine by developing a product for getting access to various hospital systems through a single and simple user's interface.[11]

Java technology utilizes a standard browser such as Netscape Navigator or Microsoft's Internet Explorer as the client's software support in the Internet/intranet environment; in the long run this gives the user one-entry point for various applications. Operational simplicity is attained through the use of a single interface for different purposes. Additional capabilities can be provided by Java and usually implemented with minimal effort.[12]

Methods

Java can be of particular use in the analysis of images in medicine. Java can facilitate the delivery of images to the user in network environment; it has intuitive controls for implementation of downloaded functions; and the

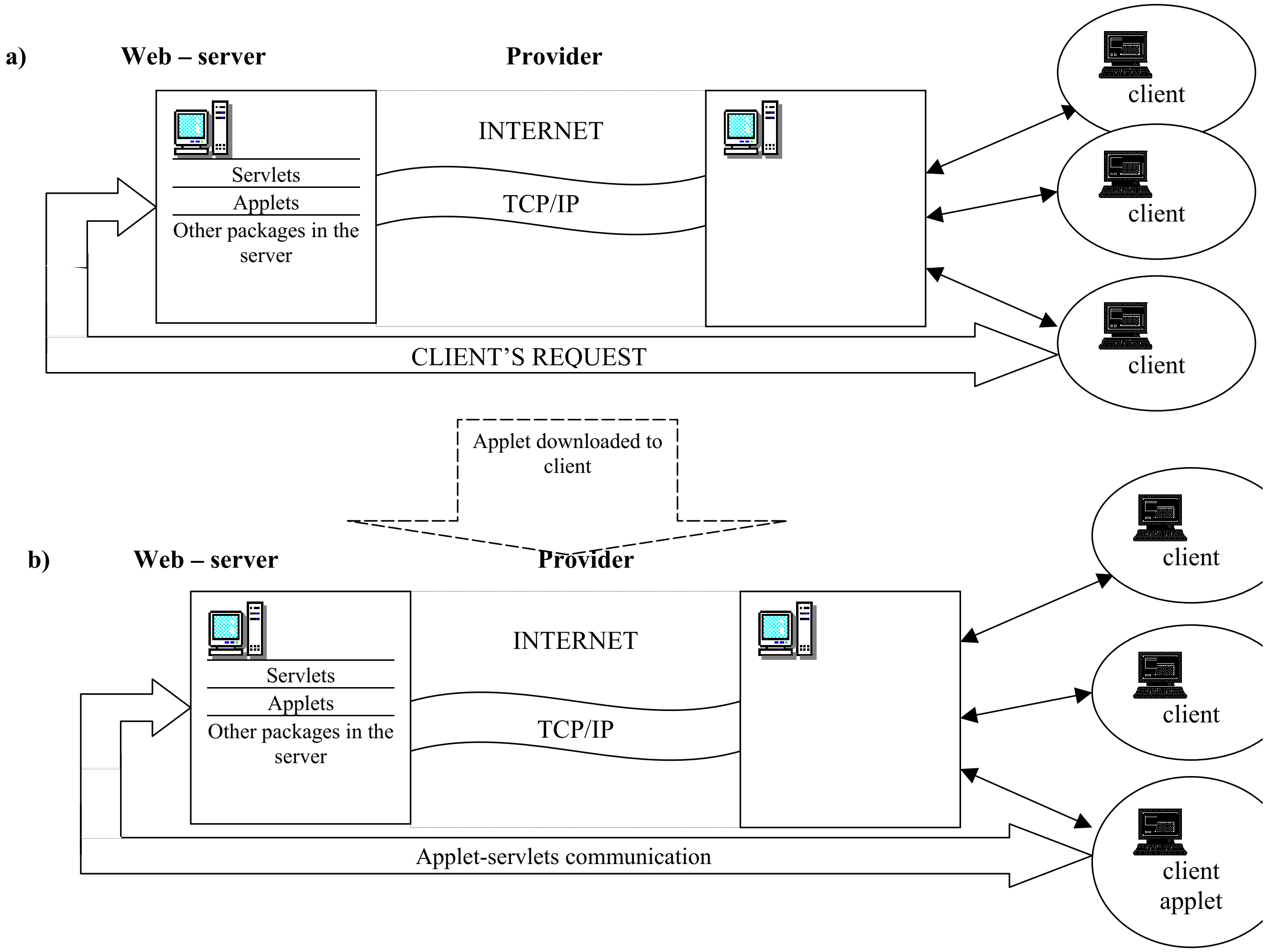
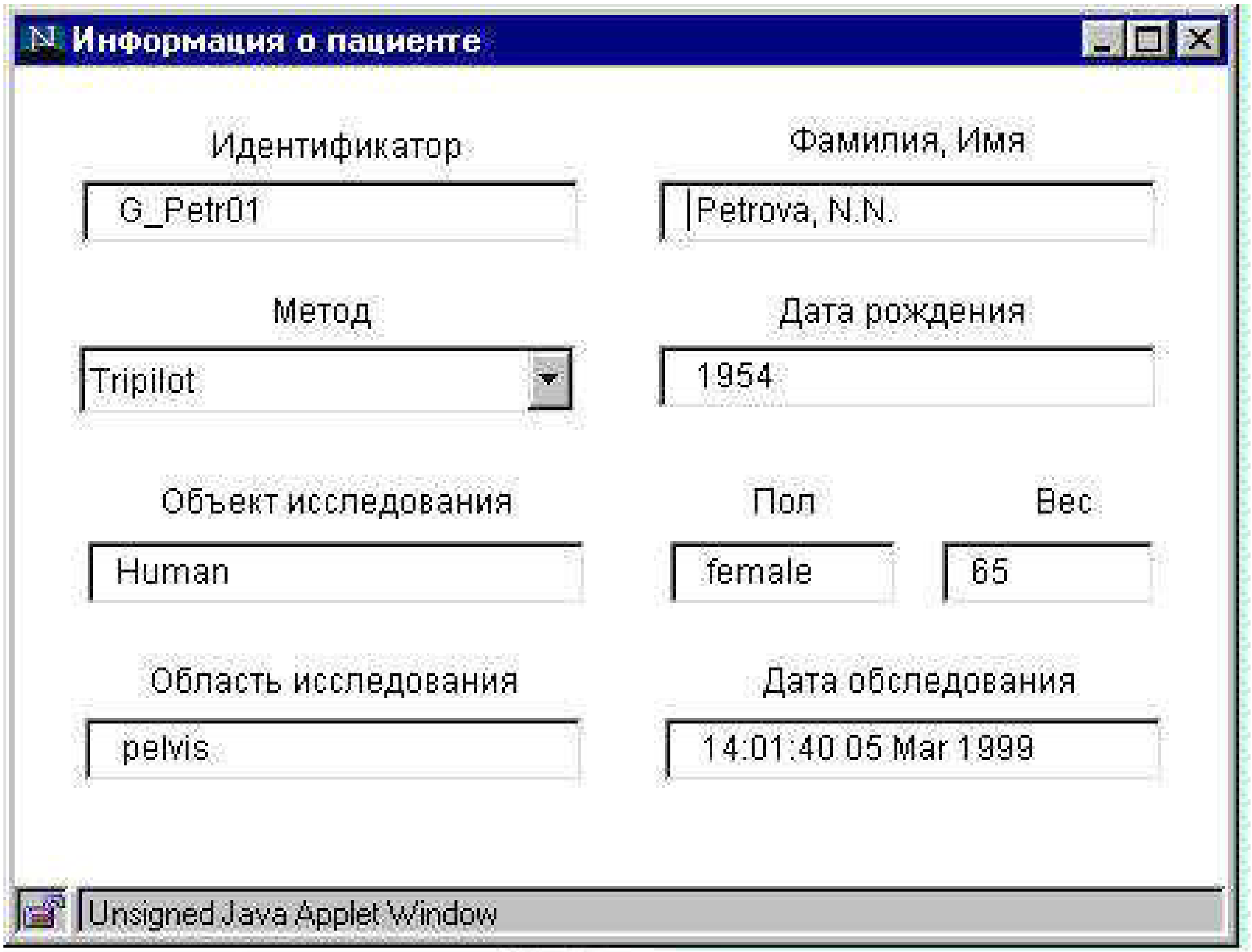


Figure 2
Layout of the Remote Image Viewing Program

browser interface is used as an environment for running a Java program (an applet).

The aforementioned features illustrate that new capabilities could be accomplished without the need for physical configuration changes on a client's workstation. Familiarization with the application interface is all that is required to proceed to work. The Visible Human Project can serve as an example.[13] One of the Java-based Digital Imaging Communication in Medicine (DICOM) [14] and HL7 supportive programs has been developed by SOFTMEDICAL.[15]

In the course of implementation of telemedicine consultation services, the Space Biomedical Center (SBC) for Training and Research in Moscow, Russia was challenged with the need to arrange access of the consultant or the consulting service coordinator to archives of medical images dispersed over various separated resources within the telemedicine network in Russia. The SBC was established through a partnership between the National Aeronautics and Space Administration (NASA), the Russian Space Agency, the Russian Ministry of Science and Technology Policy and Moscow State University in the spring of 1995.



Информация о пациенте

Идентификатор

G_Petr01

Фамилия, Имя

Petrova, N.N.

Метод

Tripilot

Дата рождения

1954

Объект исследования

Human

Пол

female

Вес

65

Область исследования

pelvis

Дата обследования

14:01:40 05 Mar 1999

Unsigned Java Applet Window

Figure 3
Patient's data interface of the Remote Image Viewing program

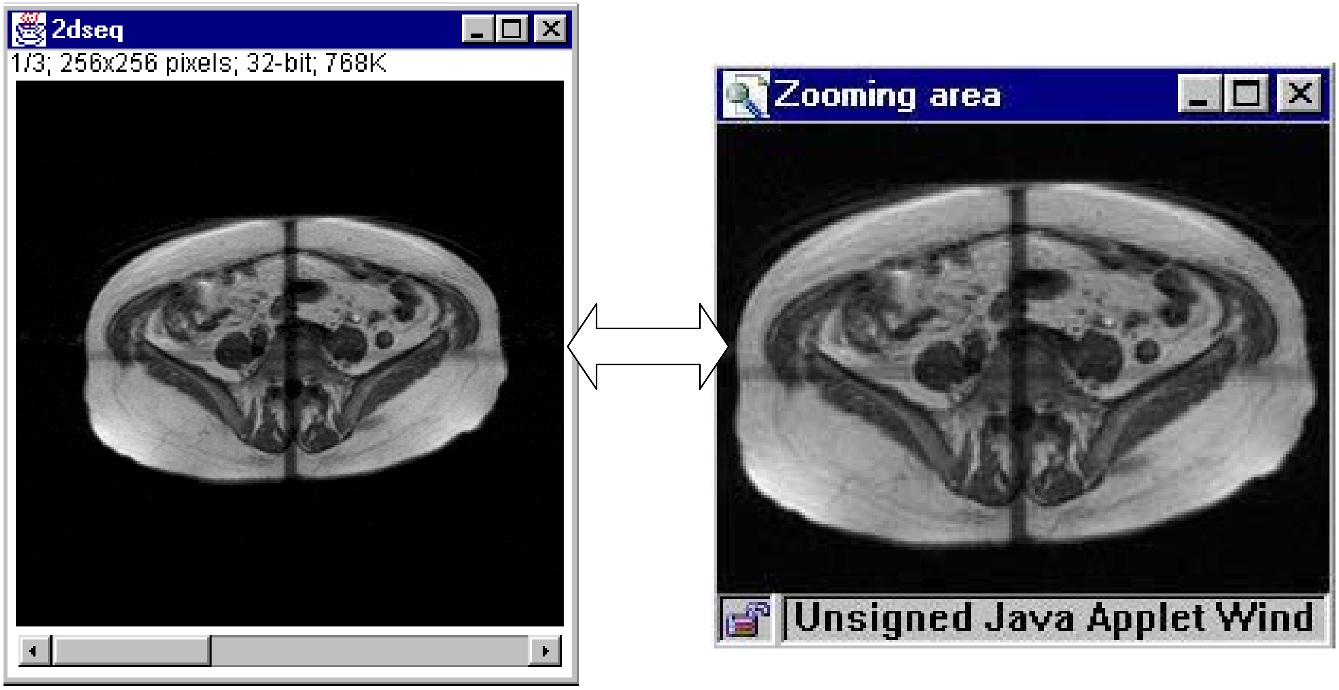
Remote image view program

The Remote Image View program provides access to the image database. The application allows browsing of Graphics Interchange Format (GIF) and Joint Photographers Experts Group (JPEG) images, and reception and visualization of results of investigation obtained using the Bruker Computer Tomography (CT).

The Remote Image View program was written in Java as a client-server application and, therefore, can be divided into two components: (1) the server component (servlet) and (2) the client component (applet).

The server component or servlet is a Java application on the server side that forwards to the applet the content of the directory with images including names of image files either in the GIF or JPEG format. Another servlet supplies data to the applet about the size of image files.

Advantages of the servlets mainly repose on the following concepts: as any Java application, the servlets are platform-independent and can be activated both from a local disk and from the network. Characteristics of servlets include 1) clients interacting with the servlet can be written in any language; 2) servlets can handle a larger numbers of requests. Unlike CGI, servlets functioning in the Webserver are not separate processes but threads that manage multiple requests only by rapid thread switching; and 3) servlets can handle requests without knowledge of the data exchange protocol.



Raw CT Image

Processed CT Image

Figure 4
Raw (left side) and processed (right side) images for evaluation.

The client's component is a Java applet enabling viewing of images. At first, the applet resides in the server. On arrival of a request, it will be downloaded to the client's side and implemented with the restriction and security attributes inherent in the applet exported via the network.

The applet interface consists of a viewing image frame, a files panel (list of patients) on the server, a common view browser, a zooming ratio field and an "Options" button (Fig. 1).

The Remote Image View program is the main part of the program-hardware complex "Telemedicine". The program has many applications. It can be used to access the image archives both individually and jointly with other applications. Future applications will make it possible to have shared access to the images that allows several users to work with them and use the application in the training process.

Layout of the program

The program operates in the following manner. First the client addresses the Hypertext Markup Language (HTML) file through the browser interface, which exports the applet (class-file). This activates a Java Virtual Machine, which starts its implementation (Fig. 2a). After initiation, the applet requests the list of images on the server (Fig. 2b. In its turn, the servlet responds to the applet request by scanning the images directory and sends file names. The number of images visualized simultaneously in the viewport depends on the image size. In the current version of the application, the viewport is limited to 300 × 350 pixels. All images that exceed these limits are viewed with the help of the scrollbars. The efficiency of the image manipulation is dependant on the client's computer work power. A special attention should be paid to the RAM and video memory capacity (each CT image is about 50 Kb). A computer working with Windows OS and 32 MB of RAM can simultaneously handle up to 20 images. A modified Remote Image Viewing application has been specially adapted in order to run a structure of data obtained during investigations of patients on the Bruker CT. The applet has been enhanced to allow viewing of patient's images (Fig 3). The server is able to process patient information data and graphics stored on the server in raw format received directly from the Bruker CT. Information is converted for representation in the browser. To optimize image download, data processing occurs at the first request. Later, the application handles the transformed data array. This sequence significantly accelerates data transfer through the network.

Another key element of this program is comprised of two independent applications: server and client. This resolves a number of problems arising due to the use of various implementations of computing environments of the Java-applications; this is particularly the case with the Microsoft implementation of the Java language (Java Virtual Machine) for Internet Explorer. Additional advantage gained by using an independent server rather than a servlet is implementation of the application in the absence of a Webserver.

The decision of whether to use server or servlet solutions depends on system (hardware and software) capabilities.

Minimum system requirements for the Remote Image Viewing: 1) preinstalled JRE 1.1 on the server side and 2) on the client side – Java 1.1-compatible browser, e.g. Netscape Navigator 4.07 or earlier versions or Internet Explorer, v.5. In absence of a browser, a special Java application must be installed on the client's side with the function of a task-oriented browser.

Results

CT images locally and remotely were evaluated and as a result there were no formidable differences between two copies of images, in spite of 32-bit to 8-bit conversion (Fig. 4). During the evaluation of this application, a problem of running the client side part in different browsers was encountered. This problem could be solved by a Java Plug-in or by running a stand-alone client application. The efficiency of a multithreaded model was proven on a large number of images. Background image loading was shown to be the best way to reduce waiting time. An original zooming tool solution is more intuitive then common zoom glasses. All zooming operations could be made in a simple way with the help of mouse manipulator.

Conclusions

Java technology opens up vast opportunities for viewing and analysis of radiological images. Platform independence facilitates access to medical resources in heterogeneous environments. The built-in security mechanism of program validation during the coding phase, affords high reliability of the product. The network technologies that are currently actively used in health service provide the fundamentals for promotion of the Java technology and transition to a qualitatively new level of medical services.

The evolution of Java Technology brings more features to the Java development tools and facilitates the creation process of applications and reduces the time of developing programs. The advantages, highlighted above, improve the quality of Java applications.

Limitations of Java such as the speed of program execution are being reduced as computer technology evolves.

Methods of getting simple, intuitive access to distributed medical information as well as considerable economic effect offered by these tools make the Java applications an integral part of telemedicine systems.

Competing interests

None declared

Acknowledgments

The authors express their gratitude to the SUN Microsystems AB (Moscow, Russia) and to the personnel of the Center of Magnetic Tomography & Spectroscopy of the Moscow State University. This project was funded under a cooperative agreement with the National Aeronautics and Space Administration.

References

1. Sun Microsystems Inc: **The Source for Java-Technology.** [http://java.sun.com]
2. Sun Microsystems Inc: **Sun Solutions Catalog.** [http://solutions.sun.com/catalogs/all/Medical_Systems_Healthcare/]
3. Padeken D, Sotiriou D, Boddy K, Gerzer R: **Health care in remote areas.** *J Med Sys* 1995, **19(1):**69-76
4. Viirre E, Warner D, Balch D, Nelson JR: **Remote Medical Consultations for Vestibular Disorders: Technological Solutions and Case Report.** *Telemed J* 1997, **3(1):**53-58
5. Brakeman L: **Universities launch new Internet with high-speed backbone.** *Telemedicine and Telehealth Networks* 1996, **2(11):**43-44
6. **Java World** *Russia* January 1997
7. Naughton P, Schildt H: **Java. The Complete Reference.** *McGraw Hill, Inc., USA* 1997
8. Emanuel S: **The outlook of Java technology.** *ComputerWorld/Russia* 1999, **28–29:**189-190
9. **Java World** *Russia* June 1998
10. Sun Microsystems Inc: **Sun Solutions Catalog** [http://solutions.sun.com/catalogs/all/Medical_Systems_Healthcare/Clinical_Information_System/30578.html]
11. Venture SystemSource Inc., DataSource Direct Inc: **MediWeb Features** [http://www.sysource.com/]
12. Sun Microsystems Inc: **Press releases about Jini and Java.** [http://www.sun.ru/java/index.html]
13. U.S. National Library of Medicine: **Visible Human Project.** [http://www.nlm.nih.gov/research/visible/visible_human.html]
14. Emelin IV: **The standard of electronic exchange of medical images.** *Computer Technologies in Medicine/Russia* 1996, **3:**56-58
15. Softmedical, Inc: **About UBIMED.** [http://www.softmedical.com/sections.php]