Use of a Rich Internet Application Solution to Present Medical Images

Chia-Hung Hsiao · Cheng-Ying Shiao · Yu-Ming Liu ·
Max Min Chao · Chung-Yueh Lien · Chi-Hsien Chen ·
Sang-Hue Yen · Shih-Tsang Tang

Published online: 19 March 2011
© Society for Imaging Informatics in Medicine 2011

Abstract Browser with Rich Internet Application (RIA) Web pages could be a powerful user interface for handling sophisticated data and applications. Then the RIA solutions would be a potential method for viewing and manipulating the most data generated in clinical processes, which can accomplish the main functionalities as general picture archiving and communication system (PACS) viewing systems. The aim of this study is to apply the RIA technology to present medical images. Both Digital

Imaging and Communications in Medicine (DICOM) and non-DICOM data can be handled by our RIA solutions. Some clinical data that are especially difficult to present using PACS viewing systems, such as ECG waveform, pathology virtual slide microscopic image, and radiotherapy plan, are as well demonstrated. Consequently, clinicians can use browser as a unique interface for acquiring all the clinical data located in different departments and information systems. And the data could be presented appropriately and processed freely by adopting the RIA technologies.

Electronic supplementary material The online version of this article (doi:10.1007/s10278-011-9374-1) contains supplementary material, which is available to authorized users.

C.-H. Hsiao
Department of Medical Informatics, Tzu Chi University,
Hualien City, Taiwan

C.-Y. Shiao · Y.-M. Liu · S.-H. Yen
Cancer Center, Taipei Veterans General Hospital,
Taipei, Taiwan

M. M. Chao
Institute of Radiological Sciences,
Central Taiwan University of Sciences and Technology,
Taichung, Taiwan

C.-Y. Lien
Institute of Biomedical Engineering,
National Yang Ming University,
Taipei, Taiwan

C.-H. Chen
Graduate Institute of Biomedical Informatics,
Taipei Medical University,
Taipei, Taiwan

S.-T. Tang (✉)
Department of Biomedical Engineering, Ming Chuan University,
5 De-Ming Rd.,
Gui-Shan, Taoyuan 33348, Taiwan
e-mail: sttang@mail.mcu.edu.tw

Keywords PACS · Clinical application · Clinical image viewing · DICOM · XML · RIA

Introduction

Medical imaging plays an important role in clinical diagnosis and therapy. Many modern hospitals have constructed Picture Archiving and Communication System (PACS) for performing digitized and integrated image-examination processes [1]. Conventionally, the structure of the PACS can be categorized into three main portions: modalities, repository, and a viewing system.

An example framework for a PACS is shown in Fig. 1. Current PACS usually deploys the Digital Imaging and Communications in Medicine (DICOM) standard for integration. DICOM is a successful and widely accepted standard in the domain of radiography and in other departments such as cardiology, pathology, and radiation oncology. All of the DICOM objects are stored in a DICOM repository and are accessed through DICOM query and retrieve. The physician can then inspect the images via a DICOM viewer.

Although PACS with DICOM standard is successful in radiology department, current PACS architecture has the

Fig. 1 Example of a PACS framework. *US* ultrasound, *NM* nuclear medicine, *CR* computed radiography

limitation when we expand the usage for handling more sophisticated image-viewing requirements. As that in Fig. 1, physicians usually wish to have a single and unique interface for viewing and manipulating all clinical data generated in all healthcare processes. However, current DICOM image-viewing systems cannot fulfill this requirement. A general-purpose DICOM viewer can handle most 2D clinical images, such as those obtained from computed tomography (CT), magnetic resonance imaging, computed radiography, ultrasound, nuclear medicine, and endoscopy. Few DICOM viewers can handle overlaps and waveforms such as radiotherapy (RT) planning data and electrocardiogram (ECG) waveforms. Furthermore, many types of examinations, such as cardiology, dental, ophthalmology, and mammography, have domain-specified image-viewing and measurement requirements. Currently, there are 35 image objects, 26 non-image objects (waveforms, presentation requirements, and structure reports), and eight RT objects defined in DICOM part 3 annex A composite information object definitions. It will thus require tremendous efforts to develop a universal DICOM viewer that can handle all DICOM objects properly. With the rapid growth of DICOM objects in the DICOM standard, increasingly advanced medical images and display specifications will emerge in the near future, and this will further impede the development of a universal DICOM viewer.

Under current PACS architecture, the physicians' dream for creating a unique viewing system to deal all the medical images and related data seems to be impossible [2, 3]. However, the need for viewing all the healthcare information is the foundation for physicians to perform proper judgment for better healthcare. Integrating and viewing all medical exams data would be the first step of patient oriented healthcare [4–9]. For viewing all clinical exam reports in a single user interface, the IHE proposed a simple Retrieve Information for Display (RID) profile for access-

ing clinical reports generated from various departments [10, 11], which can utilize a single universal interface (a Web browser) to access and navigate all medical information. The IHE RID profile contains the Information Source, which provides the clinical examination results, and the Display, which supports the presentation of the results. The RID utilizes HTTP to access clinical documents that are stored in Information Sources. Under RID specification, the clinical documents could be in Portable Document Format, or Hypertext Markup Language (HTML) or HL7 Clinical Document Architecture (CDA) formats. An example of an IHE RID profile for accessing clinical documents is shown in Fig. 2. An RID-based Web solution would be a proper architecture for next generation of medical information systems. Currently, many hospitals have following similar Web architecture for accessing clinical documents. The only shortage is that Web solution usually has limited functionalities comparing with conventional window applications. However, the shortage has gradually overcome by more powerful Web technologies (Rich Internet Application, RIA). Browser with RIA Web pages could be a powerful and unique user interface for handling sophisticated data and applications. As such, it is worth investigating RIA applications for use in the field of healthcare.

In this paper, we will use an RIA technology (Microsoft Silverlight) for handling multimedia clinical data. We will demo how to use the RIA solution to present 2D medical images, ECG waveform, pathology microscope image, and RT planning result on browser. We will follow the framework demonstrated in Fig. 2 for integrating clinical data generated from different departments. In healthcare, each different type of clinical data might require domain-specified systems for appropriately processing. As in Fig. 2, RID information sources would be located in different departments and systems for handling the domain-specified

Fig. 2 An example of an IHE RID profile for accessing clinical documents

clinical contents. The RID information sources could be combined with conventional clinical systems, for example, combining with PACS in radiology department, for accessing clinical exam results. Consequently, clinicians can use browser as a RID Display for acquiring all the clinical data located in different departments. And the data could be presented appropriately and processed freely by adopting the RIA technologies in the Web-based framework. We will show the power of the RIA solution. The limitation and drawback of the RIA solution will also be addressed in the paper. Finally, some considerations for bringing the RIA solution into real usage will also be discussed in this paper.

Materials and Methods

The Microsoft Silverlight RIA is first introduced, followed by a discussion of how to apply the RIA to draw medical waveforms in a Web browser. The integration of the RIA solution with PACS is then discussed, and finally, the handling of DICOM tag information in the Web and the handling of domain-specific DICOM objects is described.

Web Graphics Functionality of Microsoft Silverlight

There are several RIA technologies for drawing graphics in browsers, such as Microsoft Silverlight, Adobe Flash, and the World Wide Web Consortium Scalable Vector Graphics recommendation [12]. Microsoft Silverlight is an RIA Web browser plug-in for handling animation, vector graphics, and audio/video that is compatible with various Web browsers (Microsoft Internet Explorer, Firefox, and Safari). Our research adopted Silverlight vector graphics for two reasons:

1. Testing revealed Silverlight to be the most stable vector graphics solution, especially for IE.
2. Silverlight supports JavaScript events, which supports the Document Object Model for handling HTML and XML data. Consequently, we can design a set of JavaScript objects to handle waveforms and images generated from information systems in different clinical domains.

Microsoft Silverlight uses XML-based tags and Extensible Application Markup Language (XAML; named by Microsoft) data to define graphics drawn in a browser. Geometric shapes, animations, user interfaces, and trigger events can be defined in an XAML file. An example of an XAML program is in Appendix 1. The corresponding result is shown in Fig. 3.

Each XAML program has a root canvas, which may contain geometric shapes, images, videos, and a subcanvas. Silverlight uses JavaScript and HTML <div> for creating and rendering the XAML graphics in the browser (as in Fig. 3). We may define JavaScript events in the canvas in the form of XAML data subobjects (as in the example, MouseLeftButtonDown,

Fig. 3 Result corresponding to the example codes

MouseLeftButtonUp, and MouseMove in the subcanvas tag and Loaded in the image tag). In addition, appropriate JavaScript codes corresponding to the events could be added, and then it could dynamically draw a line in the subcanvas using the three events (onMouseDown, onMouseUp, and onMouseMove). When the mouse left button is pressed, a line starting point (x_1 , y_1) will be set by the JavaScript code defined for the onMouseDown function. The movement of the mouse will then continuously change the line ending point by the event function onMouseMove. Finally, the line ending point (x_2 , y_2) is set by the event function onMouseUp. Eventually, a line will be fixed on the canvas. A line object formatted similar as `<Line X1="*" Y1="*" X2="*" Y2="*" Stroke="Red" StrokeThickness="3"/>` will be dynamically added to the subcanvas. Silverlight refreshes the presentation of XAML graphics for each JavaScript event, and hence, lines can be drawn dynamically on the canvas using the drawing functionalities, as shown in Fig. 4.

In Fig. 4, when a medical image has been loaded by the event "LoadImage" defined in the image object, the properties of the image are also extracted from the DICOM image object. The image size and pixel spacing are loaded as variables in the JavaScript codes. In the case presented in Figs. 3 and 4, the image size is 512×512 pixels, and the pixel spacing is 0.703 mm in both the horizontal and vertical directions. As a result, in addition to drawing lines on the image, the real length corresponding to each line can be measured.

Drawing Waveforms in the Browser

Conventionally, it is difficult to draw a curve or waveform in a Web browser, and there is no such tag for irregular curves or waveforms defined in HTML specifications. A

Fig. 4 **a** A line was added on the canvas: `<Line X1="328" Y1="165" X2="323" Y2="204" Stroke="Red" StrokeThickness="3"/>`. **b** Added another line: `<Line X1="302" Y1="201" X2="339" Y2="205" Stroke="Red" StrokeThickness="3"/>`

curve may be represented in a browser by linking a series of small HTML tags, such as the small `<div>` tags [13]. However, many small `<div>` tags would be required to draw complicated curves, and browsers are not designed to handle huge numbers of tags, resulting in poor efficiency. Silverlight defines an Ink object for drawing curves. A waveform or

curve can be represented by a single Ink object, thus significantly improving the efficiency. The Silverlight Ink object also is XML-formatted and can be defined in XAML data. An example of an Ink object is in Appendix 2.

In this example, Silverlight uses an InkPresenter object to define curves; both the background image and the strokes are also assigned. There are two strokes, representing two curves on the canvas. Each stroke contains the DrawingAttributes and StylusPoints attributes for defining drawing properties and the points of the curve. The drawing and refreshing of Silverlight objects may soon be the same as for conventional Windows applications. Consequently, the curves or waveforms can be updated continuously and smoothly, a function that can also be triggered by JavaScript events.

Figure 5 presents a dynamic ECG waveform generated by a Silverlight Ink object. The ECG waveform is shifted and displayed smoothly, triggered by a JavaScript time event. As in Fig. 4, we can also add measurements to the waveform using JavaScript mouse events. The physiologic signal waveform can then be presented in the browser, which would be very useful for the real-time surveillance of a patient's condition.

Integrating the RIA Solution with the PACS

As described in the "Introduction," most medical images are stored on a PACS server, which supports DICOM protocols for clinical applications that allow query and retrieval of the stored images. However, DICOM protocols are not compatible with Web-based protocols. And medical images that are stored as DICOM objects cannot be handled or displayed directly in the browser. As described in our previous study [13], an interface system must be constructed for integrating PACS with Web. The interface supports standard DICOM network communication services (DICOM Storage, Query, and Retrieve) for accessing medical images that stored in PACS. And the interface system also is a Web server that supports HTTP protocols and provides Web pages for users using browser to search and observe medical images. The Web-accessing specification Web Access to DICOM Persistent Objects (WADO) has also been defined in the current DICOM standard [14], which includes two Multipurpose Internet Mail Extension (MIME) types: application/dicom and image/jpeg. There are currently many commercial and freeware PACS systems that support Web solutions so that users can access medical images in a browser. Most Web-based PACS systems use the application/dicom MIME type to

```
http://140.129.64.226/WADORet/MR2S1IM4_R358_C358_WC500_WW1000.jpg?requestType=WADO&StudyUID=1.2.840.113674.1118.54.200&SeriesUID=1.2.840.113674.1118.54.179.300&ObjectUID=1.2.840.113674.950809132638066.100&contentType=image/jpeg&Rows=307&Columns=307&Region=0.2,0.2,0.8,0.8&WindowCenter=500&WindowWidth=1000
```


Fig. 5 Dynamic presentation of an ECG waveform using a Silverlight Ink object

provide powerful image-viewing functionalities, which utilize plug-in components for handling DICOM objects. Although these plug-in components may have powerful viewing functionalities, they are also associated with compatibility problems with different types and versions of the browser.

Compatibility with different browsers may be better achieved using the pure Web solution that supports the image/jpeg MIME type. The pure Web solution implemented with RIA technologies can also provide image-viewing functionalities that are as powerful as those of Windows applications or plug-in Web solutions. With our RIA solution, the DICOM objects are retrieved from the PACS server and then sent to the Web interface system through DICOM query/retrieve protocols. The DICOM objects are dynamically converted to Joint Photographic Experts Group (JPEG) images according to the parameter settings in the HTTP request. An example of an HTTP request with parameters for identifying the size and window level of a specific image object is

The format and parameters in the example follow DICOM WADO specifications. An RIA user interface dynamically generates the HTTP request, and the JPEG image corresponding to a specified DICOM image object will be generated according to the image-presentation parameters (Rows, Columns, Region, and WindowCenter/WindowWidth) in the request. When the image-presentation parameters are adjustable by using the RIA interface in the browser, a corresponding JavaScript event function will dynamically initiate an HTTP request. The Web server then reconstructs the corresponding new JPEG image. Consequently, the user can adjust the presentation of images in the browser in the same way as in a conventional Windows application.

Handling DICOM Tag Information on the Web

According to DICOM specifications, a considerable amount of clinical information also should be added into

DICOM image objects as tag data, which is useful and important for diagnosis. The XML method was adopted to present the DICOM tag attributes on the Web [15]. An example demonstrating the XML data contain DICOM tag information is in Appendix 3 (some tags are not shown in the example for simplicity). According to the DICOM standard, each DICOM tag element has four parts: (1) Tag: Group and Element value, (2) VR: Value Representation data type, (3) Data length, and (4) Data. As with the XML data in the example, Group and Element value are added to each tag. VR and Data length are converted to VRName and Length attributes, respectively. Two other attributes, Fieldname and Offset (offset value of each data in the DICOM file), are also added. Most of the data in the DICOM object are converted to text format and added to the corresponding XML tags. The pixel data (in Tag7fe00010) are not converted into text and added to an XML tag. The converting program only adds the first ten values of the pixel data. Consequently, the file size of the converted XML data is small enough to be easily handled by the browser. When the browser requests an image, the associated XML data would also be sent to the browser. In addition, the XML data can be further handled by JavaScript in the RIA solution. For example, according to Tag00280030 (Pixel Spacing), the real size of each pixel is achieved. Consequently, it is able to perform measurements and calculate the real lengths of lines that are drawn on the medical images, as demonstrated in Fig. 4.

Handling Domain-Specific DICOM Objects

As stated in the “Introduction,” more than 70 clinical data objects are defined in the DICOM standard. Some of the objects (e.g., RT planning results, ECG waveforms, and structured reports) have a very sophisticated data structure. The DICOM standard uses nested data elements (Sequence

data elements) to deal with sophisticated data. However, some domain-specific DICOM objects cannot be handled by a general DICOM viewer. In the proposed RIA solution, the DICOM objects are first converted to XML data and then handled with specific Web pages, such as a specific ECG-viewing Web page to deal with ECG XML data converted from an ECG DICOM object and a specific treatment-planning view Web page to handle RT planning and CT images. An example of an XML file that was converted from a DICOM RT object is in Appendix 4. Using the XML data, we can present RT planning results on Web.

Results

A portal interface of the developed system is shown in Fig. 6. In this section, the RIA medical image-viewing system is presented, which as well supports the 3D cross-sectional image. The RIA solution for presenting DICOM RT planning results is then described, and finally digital microscopy images are exhibited.

RIA Medical Image-Viewing System

A Web-based PACS system supports DICOM query protocols for searching medical images stored on the

servers and for responding to search results in the browser. The functionalities provided by the developed RIA system are the same as those provided by Web-based PACS systems that need special MIME plug-in components, as shown in Fig. 7. Figure 7a presents the Web search page (in traditional Chinese characters) for assigning search conditions (patient name, identification number, gender, age, and modalities). Figure 7b shows the search results, which comprise a series of corresponding medical images.

In addition to being able to view all of the resulting images in the Web page, it is possible to select a specific image for further editing, by clicking the “Annotation” button near by each image, which initiates an HTTP request with image Service–Object Pair instance Unique Identifiers to the RIA Web server. The RIA server then responds another user interface, as shown in Fig. 8, which handles the JPEG images and XML files generated from DICOM image objects and provides the functionalities of viewing, editing, drawing, measuring, and rescaling, similar to a traditional DICOM image-viewing system. Annotations and DICOM tag data in the XML file can also be presented. As described in the “[Materials and Methods](#),” real measurements can also be made based on the DICOM pixel spacing information. Most of the functionalities demonstrated in Fig. 8 can be accom-

Fig. 6 An example screenshot of the developed system

(a)

(b)

Fig. 7 **a** Web search interface (in traditional Chinese characters). **b** Series of corresponding medical images that are assigned to the search conditions given in **a**

plished by the RIA solution running in a regular browser without special plug-in components. Only two functions should be accomplished with the cooperation of the Web server, the save function of graphic drawing and annotation results, and the adjustment function of the window level and window width of the DICOM image. As discussed in the “Materials and Methods,” each time a new window width/level is set, a WADO HTTP request with window width/level parameters is sent back to the Web server, which then generates a new JPEG image that is displayed in the browser. Consequently, the window width/level is also adjustable in the Web-based image-viewing solution.

Viewing 3D Cross-Sectional Images on the Web

The proposed solution also provides viewing 3D cross-sectional imaging on the Web. A series of CT or MR cross-sectional images can be built up as a 3D voxel array on the developed Web server. The user interface of the 3D cross-sectional view is shown in Fig. 9, which

displays transverse, sagittal, and coronal views. The locations of the cross sections are adjustable directly on the RIA Web page by the three red lines on the interface (e.g., dragging the vertical red line on the transverse image will change the corresponding sagittal view). A change in the cross-sectional location is accomplished by issuing an HTTP request (listed in the lower center of Fig. 9) to the Web server, which is similar to a standard WADO request with two extra parameters: ViewAxis and Index (meaning the location index). The Web server then generates the corresponding JPEG image that is dis-

(a)

(b)

Fig. 8 **a** The original image in RIA interface. **b** The image after size rescaling, window level adjusting, distance measuring, and annotating using the functionalities of the RIA interface

Fig. 9 User interface for three cross-sectional views of a 3D image voxel

played in the browser. The sizes of the sagittal and coronal images can be calculated according to the original transverse image size and the related resolution information contained in the DICOM tags (Pixel Spacing and Slice Thickness). Consequently, three cross-sectional views for a 3D voxel array can be generated in the Web server, and 3D navigation of the human body structure can be achieved in the browser.

RIA Solution for Presenting DICOM RT Planning Results

RT objects contain contours for dose distributions and vital organs related to a series of DICOM CT or MR images of relational organs. As stated in the “[Materials and Methods](#),” RT planning results can be formatted as DICOM RT objects and stored in the PACS server. However, the DICOM RT objects and RT planning contours cannot usually be processed in a general-purpose DICOM viewer. In fact, there is no general-purpose system that can handle all of the DICOM objects and present all of the DICOM data appropriately.

Most of the information defined in the DICOM standard can be transformed into XML files, as explained in the “[Materials and Methods](#).” The XML data can thus be easily processed via RIA Web solutions. Using Web architecture, we can construct many special-purpose Web servers for various special types of clinical data and then integrate the special-purpose Web servers using HTTP linkages. Conse-

quently, all of the clinical data can be presented appropriately using a single browser.

Figure 10 presents a simple architecture of an integration Web solution designed to handle medical images and RT planning results, in which the DICOM image and RT data can be processed in separate Web servers. Eventually, both data can be presented in the same browser. The processing and presentation of RT data have many domain-specific requirements. Figure 11 shows the preliminary results using an RIA solution to present two series of CT images and their corresponding RT planning results. The upper frames

Fig. 10 Integrated Web framework for handling DICOM images and RT data

Fig. 11 RIA solution for presenting RT images and contours. *ROI* region of interest, *CTV* clinical target volume

of Fig. 11 show two series of CT image icons that were captured from a patient; the left frame of icons was captured at the first CT session, before RT, and the right frame shows icons that were captured after 1 month of treatment. When an icon is selected, the corresponding true size of the images and the RT planning contours are shown in the large lower images. The contours are constructed from two RT planning results, one from before therapy and the other from after 1 month of treatment. The RT planning system generates two DICOM RT objects corresponding to the two planning results, which are then converted to XML files. Finally, JavaScript is applied to extract the information from the XML files and draw contours in the Web browser. The contours in the browser are also erasable and editable. For example, in Fig. 11, a specific contour is selected in the right bottom image by the small transparent rectangle with a deep blue contour in the rectangle. And the description of the contour would be displayed in a message box as that shows in Fig. 11.

Viewing Digital Microscopy Images on the Web

Pathologists usually prefer to examine microscope slides freely, first taking a gross view at low resolution and then examining a portion in detail at a high resolution. There are currently commercial products that can scan and digitize microscope specimens at resolutions that are equivalent to those achievable using an optical microscope. This therefore constitutes a virtual slide-viewing system that is similar to conventional optical microscopy in that it allows

adjustment of resolution and free navigation of any portion of the specimen, similar to the well-established map navigation available on the Web [16–18].

Adapting the Silverlight RIA Web solution, we can also construct a Web-based virtual slide system. The user interface of the virtual slide system, as shown in Fig. 12, deploys the functionalities similar to those described for radiological viewing in Fig. 8. In addition to image annotation and measurement, a navigating icon is located on the upper-left corner, which represents the whole slide. A small rectangular area on the icon shows the current scope, and this can be changed by clicking the icon image or dragging the rectangular area. The resolution can be changed by selecting the required magnification number (40, 20, 10, 5, or 2.5) on menu bar. The developed virtual slide system is more powerful than conventional microscopy; for example, it provides dynamically added annotations on the viewing images. We refer the reader to our previous paper for details regarding handling and presenting huge microscopy images on the Web [18].

Discussion

We first discuss the pros and cons of the RIA Web solution for handling clinical data and then the Web solution for handling non-DICOM multimedia contents. Finally, a proposal for standardization of healthcare RIA requirements is presented.

Fig. 12 Silverlight RIA solution for presenting digital microscopy images

Pros and Cons of the Web Solution for Handling Clinical Data

There are many benefits associated with adopting Web architecture for handling clinical data. Web solutions provide a well-known and unique user interface: the browser. Furthermore, integration is much simpler by using HTTP in the heterogeneous and distributed healthcare environment. There is thus a huge potential for providing more powerful and convenient healthcare services. Web-based solutions are conventionally more difficult to implement and have more limited functionalities compared to Windows-based solution. However, the limitations have been gradually (or, in some applications, rapidly) overcome with the advent of RIA technologies. As demonstrated in this paper, RIA solutions can implement most of the image-viewing functionalities as using conventional DICOM viewing systems. In addition, Web technologies are more appropriate than the conventional DICOM viewing systems for handling very large microscopy images [18]. The only drawback of Web architecture for handling medical images is that the image processing and reconstruction must be accomplished on Web servers. Image processing requires considerable computation, and DICOM images are traditionally handled and processed in standalone DICOM viewing systems. In the RIA Web solution, as shown in Fig. 10, DICOM images should be retrieved and handled

on a Web server, and the process for generating JPEG images on the Web server is very time-consuming. Unlike conventional PACS viewing systems, it is the Web server that performs most of the image processing when using the RIA solution, and this may constitute a loading overhead when all of the images are loaded onto a single Web server. However, it may be possible to deploy many Web servers to share the loading. If the clinical workflow requires the processing of many images, such as for the image-examination workflow from a radiology department, it may even be possible to allocate a Web server to each image-viewing workstation. DICOM images would thus be processed at the workstation as a standalone DICOM viewing system; the only difference is that images would be viewed using the browser on the workstation. Apart from radiology departments, for most clinical workflows physicians usually need to share image-processing results (the same image presentation) with each other. Therefore, in most cases, the Web server only has to generate JPEG images at the first viewing of the images and can then save the generated JPEG images for subsequent viewing. Since JPEG images are much smaller than the original DICOM objects, the Web-based solution will have a much quicker response time than the conventional DICOM viewing system in internet. Therefore, the pure Web-based solution would be more suitable than standalone DICOM viewing applications or DICOM plug-in browsers for sharing

image-examination results in a low-bandwidth environment, such as when sharing images between hospitals.

Clinically, most image-processing tasks are performed in high-performance workstations within radiology departments, such as quality-control workstations, computer-assisted diagnosis systems, and 3D imaging workstations. Most current systems that perform the image processes are not Web-based solutions. However, the performance of the Web RIA systems may be as good as traditional viewing systems in a stable and fast intranet environment, and as we have described, the Web solution is suitable for sharing image-processing results across departments or hospitals. The adoption of a Web solution has brought a new architecture for sharing medical images in clinical workflows. It would be interesting and useful to investigate the most appropriate architecture of Web systems for clinical image-examination processes.

Web Solutions for Handling Non-DICOM Multimedia Data

Some clinical data are not suitable for formatting as DICOM objects or for transferring using DICOM protocols. For example, audio and video data need to be sent using real-time streaming protocols so that users can observe the audio and video simultaneously and be able to respond rapidly to them. However, DICOM protocols cannot fulfill this requirement; a viewing system first needs to receive all DICOM objects before processing and presenting them. The current DICOM standard does not have the capability of sending and processing parts of data objects. If the data object is large, it would take a lot of time to send the entire object. Therefore, a real-time response cannot be implemented in DICOM data packets and use DICOM network protocols. Some medical images, such as digital microscopy images, also need to be stored in very large files that cannot be processed following DICOM standards. As shown in the “**Results**,” the Web solution is capable of handling such large data files and is consequently superior to conventional PACS for handling various types of clinical data. The architecture of the Web solution for handling different kinds of clinical data is shown in Fig. 13. Medical data may be stored in separate servers and processed in separate Web systems; however, all of the data can be integrated and presented in a unique browser viewing interface.

Healthcare RIA Requirements Standardization

There are some issues other than system development that require further investigation and discussion before adopting RIA solutions in the healthcare domain. First, as shown in previous HTTP request with parameters, we use DICOM WADO protocols for the browser to access

Fig. 13 Web solutions for accessing various types of medical data

specific images and determine the presentation of the images. Figure 9 also shows that a WADO-like request with two extra parameters (ViewAxis and Index) can be used for navigating three cross-sectional views of a series of medical images. We also show that the Web protocol can be used to access large data files containing microscopy images and video data (Fig. 13). Consequently, carefully considered Web-accessing specifications could be used as a standard for sharing all clinical data. These specifications may be similar to current WADO standards with appropriate extensions and modifications.

Other than standardizing the Web-accessing protocol, RIA solutions would also require widely and stably supported browsers. This would be very important for adopting RIA solutions designed to share clinical documents across healthcare enterprises and to handle electronic medical records (EMRs) that require long-term preservation. Currently, there are several RIA solutions with different specifications and technologies. This study selected Microsoft Silverlight due to its ease of integration with Microsoft Visual Studio .net Web programming. Silverlight Web pages are supported by most of the popular browsers (Microsoft IE 6 and 7, Firefox 2 and 3, Safari 2, and Google Chrome 1). However, the RIA technologies, browsers, and platforms update and change very frequently, which would be greatly complicated for RIA with respect to preservation of EMRs. For example, Fig. 11 shows the use of Silverlight to present RT planning results, which we have shown that can function properly using most of the current browsers. However, because of the rapid evolution of computer platforms and browsers, we cannot guarantee that the RIA Web page will be presented consistently in, say, 5 years later. RIA solutions would be an appropriate tool with which to present sophisticated clinical data. However, in the case of EMR documents, RIA solutions should be well-defined and widely supported. Standardization is crucial for the effective use of RIA solutions in the healthcare domain.

Conclusions

This paper presents some demonstrations of the use of RIA Web solutions for the processing and presentation of clinical data in a Web browser. The developed Web RIA image-viewing system can accomplish the main functionalities as general PACS viewing systems. However, RIA Web interfaces can also readily present certain clinical data that are difficult to present using PACS viewing systems, such as ECG waveforms and virtual-slide microscopy images. We also have shown that domain-specific clinical data, such as RT planning results, can be handled by specially design Web servers. Web solutions are the current trend in clinical informatics. Although currently there are few clinical systems using RIA solutions, we believe that such Internet applications for use in the healthcare domain will experience explosive growth in the near future.

References

- Hastings S, Oster S, Langella S, et al: A grid-based image archival and analysis system. *J Am Med Inform Assoc* 12(3):286–295, 2005
- Kurc T, Janies DA, Johnson AD, Langella S, et al: An XML-based system for synthesis of data from disparate databases. *J Am Med Inform Assoc* 13(3):289–301, 2006
- Bessière K, Pressman S, Kiesler S, Kraut R: Effects of internet use on health and depression: a longitudinal study. *J Med Internet Res* 12(1):e6, 2010
- Tang ST, Huang YF, Hsiao ML, et al: Rapid prototyping strategy for a surgical data warehouse. *Methods Inf Med* 42(3):243–250, 2003
- Embi PJ, Payne PR: Clinical research informatics: challenges, opportunities and definition for an emerging domain. *J Am Med Inform Assoc* 16(3):316–327, 2009
- Kaelber DC, Jha AK, Johnston D, et al: A research agenda for personal health records (PHRs). *J Am Med Inform Assoc* 15(6):729–736, 2008
- Richesson RL, Krischer J: Data standards in clinical research: gaps, overlaps, challenges and future directions. *J Am Med Inform Assoc* 14(6):687–696, 2007
- Claudia Pagliari: Design and evaluation in eHealth: challenges and implications for an interdisciplinary field. *J Med Internet Res* 9(2):e15, 2007
- Peiris DP, Joshi R, Webster RJ, et al: An electronic clinical decision support tool to assist primary care providers in cardiovascular disease risk management: development and mixed methods evaluation. *J Med Internet Res* 11(4):e51, 2009
- HIMSS and RSNA IHE Integration Healthcare Enterprise. Available at <http://www.ihe.net>. Accessed 17 Mar 2010.
- Weber GM, Murphy SN, McMurry AJ, et al: The Shared Health Research Information Network (SHRINE): a prototype federated query tool for clinical data repositories. *J Am Med Inform Assoc* 16(5):624–630, 2009
- W3C Scalable Vector Graphics (SVG) recommendation. Available at <http://www.w3.org/Graphics/SVG/>. Accessed 17 Mar 2010.
- Hsiao CH, Hsu TC, Chang JN, et al: Developing a medical image content repository for e-learning. *J Digit Imaging* 19(3):207–215, 2006
- DICOM Standard Part 18: Web Access to DICOM Persistent Objects (WADO) Service Class Specifications, National Electrical Manufacture Association, USA, 2008. Available at ftp://medical.nema.org/medical/dicom/2008/08_18pu.pdf. Accessed 17 Mar 2010.
- Zhao L, Lee KP, Hu J: Generating XML schemas for DICOM structured reporting templates. *J Am Med Inform Assoc* 12(1):72–83, 2005
- Kumar RK, Velan GM, Korell SO, et al: Virtual microscopy for learning and assessment in pathology. *J Pathol* 204(5):613–618, 2004
- Kumar RK, Freeman B, Velan GM, De Permentier PJ: Integrating histology and histopathology teaching in practical classes using virtual slides. *Anat Rec B New Anat* 289(4):128–133, 2006
- Lien CY, Teng HC, Chen DJ, et al: A Web-based solution for viewing large-sized microscopic images. *J Digit Imaging* 22(3):275–285, 2009