

## Recording and Reenactment of Collaborative Diagnosis Sessions using DICOM

Aldo von Wangenheim,<sup>1</sup> Martin Prüsse,<sup>1</sup> Rafael Simon Maia,<sup>1</sup> Daniel Duarte Abdala,<sup>2</sup>  
André Germano Regert,<sup>1</sup> Luiz Felipe de Souza Nobre,<sup>3</sup> and Eros Comunello<sup>1</sup>

This paper presents a radiological collaborative tool capable of direct manipulation of Digital Imaging and Communications in Medicine (DICOM) images on both sides, and also recording and reenacting of a recorded session. A special collaborative application protocol formerly developed was extended and used as basis for the development of collaborative session recording and playback processes. The protocol is used today for real-time radiological meetings through the Internet. This new standard for collaborative sessions makes possible other uses for the protocol, such as asynchronous collaborative sessions, decision regulation, auditing, and educational applications. Experimental results are given which compare this protocol with other popular collaborative approaches. Comparison of these results shows that the proposed protocol performs much better than other approaches when run under controlled conditions.

**KEY WORDS:** Collaborative session, session playback, session recording, telemedicine, teleradiology, DICOM workstation

### INTRODUCTION

Collaborative telemedicine systems for the joint remote discussion of cases on the basis of shared digital medical images are still being considered a cutting edge technology, waiting to be introduced in most hospitals. From the technical point of view, however, they are not a novelty anymore. In this context, a common communication standard among those systems has yet to appear, but all existing approaches share the philosophy of using an application-level network communication protocol of some sort, that allows two or more instances of some medical imaging workstation to communicate and share images, markups, pointing devices, and presentation state

modifications of these images, helping a geographically distributed medical team to explore some examination in real-time and to understand the patient's situation.

Normally, when a collaborative teleradiological session finishes, the connection is broken and everything that was discussed, every structure pointed at, every windowing modification or magnification that helped to explain a detail is lost, regardless of how instructive or helpful it could be for further attempts to diagnose similar cases, for teaching purposes, or as a simple and practical multimedia presentation of findings reporting and even for auditing purposes. Thus, the capability to provide a quality-preserving recording of collaborative sessions for later playback would be a very useful feature for digital imaging workstations.

---

<sup>1</sup>From the The Cyclops Group, CS Department—Technology Centre, Federal University of Santa Catarina—UFSC, 88.049-900 Florianópolis, SC, Brazil.

<sup>2</sup>From the Fachbereich Mathematik und Informatik, Mathematisch-Naturwissenschaftliche Fakultät, Universität Münster, Münster, Germany.

<sup>3</sup>From the Telemedicine Service, University Hospital, Federal University of Santa Catarina—UFSC, Florianópolis, Brazil.

Correspondence to: Rafael Simon Maia, The Cyclops Group, CS Department—Technology Centre, Federal University of Santa Catarina—UFSC, 88.049-900 Florianópolis, SC, Brazil; tel: +55-48-3721-9516; Fax: +55-48-3721-9516; e-mail: [simon@inf.ufsc.br](mailto:simon@inf.ufsc.br)

Copyright © 2008 by Society for Imaging Informatics in Medicine

Online publication 30 May 2008

doi: 10.1007/s10278-008-9130-3

For such recordings to be useful, they would have to be stored in an easy-to-handle, easy-to-share fashion representing all actions made during the collaborative session, even audio recordings of the conversations and mouse movements, as a simple mouse movement could be the decisive pointing to a structure responsible for a given pathology. The original medical images, probably in DICOM format, shared during the session should also be provided, as the future audience of the session could be interested in investigating other aspects of the examinations not covered in the session.

Simple movie recordings of the workstation's screen during a collaborative session clearly do not satisfy the requirements posed above, even if such screen recordings could provide portable data that would be able to be "played" with any standard media player software. A different, but also simple solution would be to develop a software module for collaborative teleradiology workstations to allow the recording of the interaction commands that have been shared between the collaborative partner systems during a session.

In other words, this could be accomplished by recording the collaborative protocol communication stream and adding to it the audio stream and the original images and making a workstation capable of reenacting a session by feeding it with the commands that were issued during a given collaborative session that occurred in the past. This would be a much more compact, flexible and easy-to-share way of representing such a session and was the approach followed by our medical imaging research group and that will be presented here. In this paper, we will focus particularly on the technical aspects of the approach we have developed, explaining our collaboration protocol and how the collaboration recording is archived. A review of current collaboration tools and a comparison between them is also presented.

### Objectives

The main objective of the work described here was to develop and test a collaborative workstation capable of recording, sharing, and reenacting teleradiological collaborative discussions, enabling easy storage and sharing of diagnosis sessions for documentation and teaching purposes.

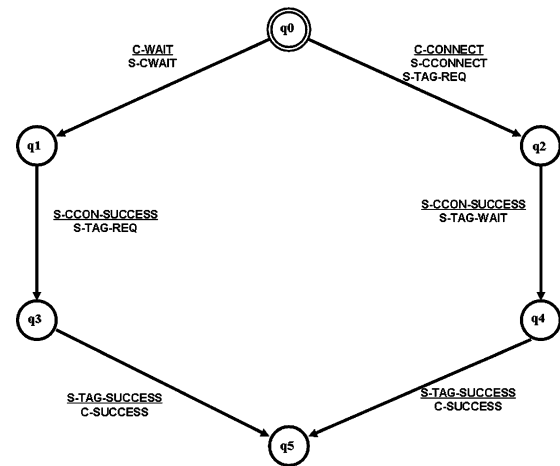


Fig 1. State-diagram for a direct connection between two clients.

For this purpose, we postulated the following requirements:

- R.1. Completeness. The resulting recordings should represent the entirety of the collaborative session, holding information on all interactions, measurements, and presentation state changes that happened (Figure 1), holding the audio stream of all partners that participated in the session and containing all the medical images and patient data handled during the session.
- R.2. Source-level data quality. Medical data such as images and structure measurements in a recorded session should be provided to a user reenacting a recorded session in the same quality as was available to the original session.
- R.3. Fine granularity. The session recording should not be an atomic video stream, allowing the user to pause a session, execute further analysis of the recorded data after the replay of a session, and to store these data locally in a PACS<sup>2</sup>.

To achieve these goals, the following specific objectives were accomplished:

- O.1. Development of a sequential collaborative teleradiological protocol suited for session recording. The Application Protocol<sup>3</sup> was developed as an interaction application protocol for the interconnection of DICOM-compliant<sup>1</sup> radiological workstations. It was

developed to enable a preexisting radiological workstation, the Cyclops Medical Station (CMS)<sup>4</sup>, to function as a collaborative telemedicine tool. This Teleradiology Workstation was called the Virtual Reading Room (VRR)<sup>3</sup>.

- O.2. *Development of a session recording strategy and a file format standard* to allow the easy recording, storage, sharing and reenactment of a collaborative session. The file format allows the storage of original image and patient data, session audio stream, and all interaction commands with their timestamps, allowing reenacting a session with pause and resume functionalities and posterior analysis of the original data by the user.

The remaining sessions of this paper will describe the technical aspects of the development of this approach and present and discuss the results achieved so far.

## METHODS

As a starting point, the teleradiological collaboration protocol developed previously<sup>3</sup> was adapted to allow its usage within a session recording/playback context. The CycAppDCM protocol was originally developed as a general purpose high-level collaboration protocol for contexts where handling of DICOM data plays a role, such as collaborative radiological sessions and distributed image analysis tasks. Its first implementation was in the Virtual Reading Room (VRR), a collaborative radiological workbench developed on top of the DICOM client called Cyclops Medical Workstation<sup>4</sup>. The CMS is a DICOM-compliant radiological workstation capable of manipulating a varied range of examination modalities, from magnetic resonance imaging (MRI) to US. It provides a series of standard image measurement and manipulation tools, such as zoom and pan, magnifying glass, windowing, MPR and DICOM printing. The VRR provides a communication layer that connects two CMS instances allowing collaborative case discussion sessions on DICOM data shared during the session. The documentation of the protocol described in<sup>3</sup> is available online at: <http://www.lapix.ufsc.br/vrr>.

For the recording and reenactment of sessions, the representation of all shared data and commands

had to receive a timestamp, providing isochronicity during the session's reenactment, allowing them to be reenacted with the same timing and speed they occurred when the session was recorded and also allowing the synchronization with the audio streams.

In the next subsection, we present briefly the CycAppDCM protocol which provides the basis for the interacting process. After, the extensions to the protocol developed to allow the recording and reenacting to be properly performed will be described. After the record and play processes have been explained, the portable session recording file format that was developed is described.

### The CycAppDCM Collaborative Application Protocol

A recorded collaborative session can actually be seen as a log of all shared interactions, described as commands of the collaboration protocol, which is stored together with the timestamps of each command, the actual audio streams produced by all participants and the DICOM data, such as images, presentation states and measurement overlays, shared through special commands of the protocol, that were issued during the session.

CycAppDCM was formally modeled as a state engine (a finite automaton) covering all states and events (state transitions) necessary to describe the situations that can arise starting with the opening of a session until it is terminated. To differentiate between control data, patient data, and communication/audio data, a special label for each data package was defined, allowing differentiation between the manipulations performed on each image set.

This label was called a *tag*, and a different one is issued for each DICOM image series that is shared during a session. Together with the protocol tag, commands not related to the manipulation of the images themselves, like voice channel control or new series being shared will be sent.

The recording process adds a temporal marker to each shared data, allowing synchronization of voice and manipulations during playback. Various videos showing the usage of the VRR and the session reenactment facility can be found at <http://www.telemedicina.ufsc.br/slv/index.php?lang=en&page=howitworks> and also at <http://www.telemedicina.ufsc.br/cms/>. Additional information can be found at the Virtual Reading Room official site<sup>9</sup>.

The CycAppDCM is protocol layered right on top of the Transmission Control Protocol/Internet Protocol (TCP/IP)<sup>10</sup>, being an Application Layer protocol. This guarantees the successful transmission and integrity of the data. It implements three well-defined steps:

1. connection opening,
2. information exchange, and
3. connection closing.

#### *Connection Opening and Session Initiation*

The figure bellow shows a simplified diagram for the connection between two clients, and definition of the first and obligatory tag, a tag for the first layer control.

#### *Information Exchange*

During the capabilities negotiation, the capabilities of each host's application are exchanged, allowing the definition of which functionalities are present on both hosts and will be available during the collaborative session. Each host is responsible to calculate the intersection of the exchanged capabilities descriptions. This intersection represents the valid operations that can be performed by both hosts, allowing different versions or configurations of collaborative systems to interact transparently. Only operations available on both hosts can be executed in a session. For example, if both hosts can adjust, the radiological density window of a computed tomography, this operation is going to be allowed in this session as a valid capability.

For information exchange (the second major state during a session), any tool function present in

the intersection of the capabilities of the involved hosts is available. This tool must have the proper *tag* assigned, declaring the series it belongs to, or another protocol *tag*, if the capability is not related to an image group. There are two basic package structures in the application protocol, one for each layer: *tagging* and *application* (Tables 1, 2, 3).

The tagging package is used for the identification of specific data shared during the session. It is used while creating new tags and mainly for the identification of data sent in a session. Within a session, there can be one or more contexts, where a context is defined by a tag.

The second-level wrapping is performed by the Application Package, which is used to trade operations in a given context. Its structure and functions of the fields are described below.

#### *Session Closing*

The closing process of a collaborative session does not differ much from the connection. It begins when a DISCONNECT message is received by the application protocol. The tags used during the session are released and made available for future use. Finally, the socket connection is closed.

#### *The Application Package*

The SUBC field has a different behavior dependent of the CMD field. When it defines textual communication (0x40), SUBC is ignored, as DATA will contain the message text. When it identifies a control field (0x20), SUBC represents one of the procedures defined in Table 4.

If CMD identifies a manipulation field (0x00), SUBC describes one of the tools on Table 5.

**Table 1. States and Actions Taken on a Direct Connection**

State	Description	Action	Description
Q0	No connection established	C-WAIT	The application protocol shall enter wait mode. (this side is the <i>server</i> on the client-server model)
Q1	Waiting for TCP/IP socket connection	C-CONNECT	The application protocol shall make the connection ( <i>client</i> on the client-server model)
Q2	Establishing TCP/IP socket connection	S-CWAIT	Listen on a TCP/IP port for a connection
		S-CCONNECT	Connects to a TCP/IP socket
		S-CCON-SUCCESS	The TCP/IP socket is open and connected
Q3	Requesting establishment of a new TAG, to use for application protocol control	S-TAG-WAIT	The application protocol waits for a new TAG request
		S-TAG-REQ	The application protocol sends a new TAG request
		S-TAG-SUCCESS	The new TAG has been successfully agreed
Q4	Waits for new TAG, to use for application protocol control establishment	C-SUCCESS	The application protocol is connected and shall enter in idle mode

**Table 2. Tagging Package Field Structure**

Field	Size	Description
CTRL	1 byte	This field identifies if the package is used for data transferring or tag establishing
TAG	1 byte	The TAG field contains a tag identifying the context of package data
SIZE	8 bytes	SIZE enumerates the number of bytes in the DATA field
DATA	SIZE bytes	Simply the data to be delivered; normally an Application Packet

### Recording a Session

When the application protocol packets are recorded while being sent to or received from a peer, all the users' actions are recorded. The packets, however, lack both absolute and relative temporal frames: It is not possible to know when they were sent, and it is also not possible to know the time intervals that occurred between packets, as the original VRR Protocol was designed to be used in a synchronous and real-time fashion. This poses the need to timestamp each packet, allowing at least identifying a temporal frame relative to the beginning of the session, when a given packet was issued. It allows each user action to be reenacted in the same temporal frame it was produced. The temporally tagged packets are stored in a XML<sup>7</sup> file with a specially developed syntax, called the Session Archive.

#### *The Session Archive*

To allow a comprehensive storage of collaborative sessions, a protocol sniffer was implemented. It captures each packet sent or received by a given client and stores it in a sequential "raw packet" file, adding a timestamp to each packet. After the recording session is ended, the analyzer parses this packet file and translates each packet into a semantic XML representation, storing it into a new file, which presents the structure shown in Figure 2.

The structure of this file, called XML Virtual Reading Room Log (XVRRL), is delimited by the <VRRFile> tag. It is divided into four sections, delimited by the tags <TOOLSET>, <COMMANDENTRIES>, <AUDIO>, and

<DICOMFILES>. The last section contains the DICOM files shared during the collaborative session, encoded in the MIME64 format to preserve binary data, each identified by a <FILE> tag, as described in Figure 3.

The <FILE> tag is responsible for holding the data of each DICOM file. The unique identifier is stored in the <NAME> tag. The content of the DICOM file is stored in the <CONTENT> tag as binary data.

The section delimited by the <TOOLSET> tag, shown in Figure 4, represents the CycAppDCM protocol capabilities negotiation. Each successfully negotiated tool functionality is represented by a <SUBC> section and references the named tool in the application package. The data shared during tool usage are described in the tools own syntax in the <DATADESCRIPTION> tag.

This structure reflects a difference between the CycAppDCM protocol capabilities negotiation procedure and the XVRRL file. Each of the tools successfully negotiated during initiation is stored by the CycAppDCM, while the XVRRL file only stores the tool commands actually used during the session recording. As a consequence, it is possible to reenact a recorded session in a client that lacks all of the tool capabilities of the original recording workstation, if it possesses at least all the tools actually employed during the recorded session.

In the section defined by the <AUDIO> tag, each conversation sequence is delimited by the <VOIPSEQ> tag, as shown in Figure 5. As in the <DICOMFILES> section, the VoIP stream is saved as MIME64 data in a <CONTENT> tag.

The <COMMANDENTRIES> section stores each command action performed during the ses-

**Table 3. Application Package Field Description**

Field	Size	Description
CMD	3 bits	This field identifies the kind of operation to be executed; these could be manipulation, communication or application control
SUBC	5 bits	The SUBC field identifies the operation to be executed, directly dependent of the CMD field
DATA	Variable	All data needed by the command and subcommand fields are placed here



Table 4. Control SUBC Field Operation

Operation	Description
SESSION_CONTROL (0x00)	Indicates one of the hosts is taking or releasing session control; the DATA field has 0 (zero) bytes
FILE_TRANSFER (0x01)	Indicates file receiving, DATA having information about file name, file size and the file itself
VOIP (0x02)	Indicates the audio stream [(VOIP)] should be started or stopped, The DATA field holds info about the format used by the audio stream
NUMBER_OF_FILES (0x03)	Indicates to the application to prepare for file transfer operation; the DATA field has 4 bytes, containing the number of files to be received or sent
DISCONNECT (0x04)	Indicates end of session, and all resources used during it must be released
APP_CAPABILITY (0x05)	Packets of this type are emitted by both sides on capability negotiation; the DATA field holds a string identifying a single capability

sion. Each action is described in an <ENTRY> section. There are two classes of actions: *examination manipulation/pointing actions* and *protocol control commands*. Each action is transparently recorded and saved during a session performed in recording mode.

Manipulation-related actions are represented as shown in Figure 6. The <ELAPSED TIME> tag sets the time when the action is to be performed during the reenactment. The <TAG> tag identifies the context of the action and point to the “TAG” field of the Tagging Packet described above. The <ACTION> tag identifies the actual action. It frames <TOOLCMD> actions, as defined during the capabilities negotiation or action provided by the protocol itself, such as <APPCMD> and <CHATCMD>.

### Application Protocol Validation Methodology

It was decided to test the protocol against current and well-known industrial standards for collaboration and the most common tools that implement at least one of these standards or that use a proprietary solution but are widely used enough to justify its inclusion in the validation. Recorded collaborative sessions, because during the replay they were reenacted and not replayed like a movie, provided the means to objectively compare different tools using exactly the same

collaborative session for each tool. This was possible because common present collaboration standards, even desktop sharing protocols, are all based upon the sharing of the representation of the user interface of a tool being run on one of the computers of a session. We explored this property of current collaborative protocols and applied the following strategy:

- A session between two instances of the Virtual Reading Room was performed under three different predefined network bandwidth settings, which were set using a bandwidth limiter. The session was recorded using the VRR session recording facility.
- The session recorded with the Virtual Reading Room for each bandwidth setting was reenacted in a collaborative session using the application sharing facility of each of the tools we have chosen for our validation comparison and under the same bandwidth limiter settings. The shared application was the VRR itself, while reenacting a recorded session in stand-alone mode. Because the VRR reenacts a session based upon the timestamps of each interaction command, this ensured that for each collaboration tool to be tested, the session was reenacted in exactly the same way. The collaborative facility of the VRR was turned off during this reenactment.

Table 5. Manipulation SUBC Field Description

Field	Description
COMMAND (0x00)	Indicates one of the operations allowed on the negotiation
COLOR_TOOL (0x01)	If the tool is a visual tool, a color can be selected
GENERIC_TOOL (0x02)	Indicates selection of one of the tools allowed on the negotiation stage
MOUSE_TOOL (0x03)	Provides a real time pointing utility, allowing a host to point an area of interest in the image, as well other real time interactions
KEY_TOOL (0x04)	Provides real time editing for text tools

<VRRLFile>
<TOOLSET>
Used Tools
</TOOLSET>
<COMMANDENTRIES>
Commands
</COMMANDENTRIES>
<AUDIO>
Audio Streams
</AUDIO>
<DICOMFILES>
Shared Files
</DICOMFILES>
</VRRLFile>

Fig 2. Structure of the XVRRL file.

c. To adequately register the latency time, both computers involved in each application sharing session were put side by side, and the collaborative session was recorded using a professional digital video camera in lossless, uncompressed DV format, at a 30 fps frame rate. This allowed us to replay the movie frame-by-frame and determine the response latency time with a precision of  $\pm 1/30$  s. This DV recording approach was also applied to the original sessions performed with the VRR.

We measured three different parameters in our experiment:

1. *Bandwidth usage* during collaboration.
2. *Latency time* until a change made on a medical image in one side is completely reflected on the other side or until the propagation of a change stops and the image representation is left incomplete.
3. *Errors* during propagation of a change.

We tested the VRR against the following collaboration applications:

<FILE>
<NAME>Unique File Name</NAME>
<CONTENT>
nZA719GGXTHyttqUo+wp44N*vF
8Qia68Cgd...
</CONTENT>
</FILE>

Fig 3. XML structure for shared DICOM files.

<TOOLSET>
<SUBC>
<NAME> Tool's Name </NAME>
<DATADESCRIPTION>
Readable Description
</DATADESCRIPTION>
</SUBC>
...
</TOOLSET>

Fig 4. Tools negotiated at a session initiation.

Applications based upon the ITU T.120 application sharing protocol:

- *NetMeeting*: implements ITU T.120 for application sharing without extensions.
- *Polycom PVX*: employs the NetMeeting ITU T.120 API for application sharing and some extra network configuration routines to be able to deal with virtual private networks (VPNs) and firewalls in a more user-friendly fashion.

Applications implementing proprietary solutions:

- *Remote Desktop*: implements Microsoft's Remote Desktop Protocol, originally based on ITU T.128 application sharing protocol, but extended to support a number of other features, being thus a proprietary protocol.
- *MSN*: also uses a T.120-related application sharing protocol that appears to be proprietary, but is different from the Remote Desktop Protocol.
- *lluminate*: web-conference tool that uses a proprietary protocol called the Collaborative Communication Framework.
- *Adobe Connect*: web-conference tool that uses a proprietary application sharing protocol called Real Time Messaging Protocol (RTMP), work over the Adobe Flash technology.

<VOIPSEQ>
<ELAPSEDTIME>
Time
</ELAPSEDTIME>
<CONTENT>
nZA719GGXTHyttqUo+wp44N*vF
8Qia68Cgd...
</CONTENT>
</VOIPSEQ>

Fig 5. Shared VoIP data representation.

<COMMANDENTRES>
<ENTRY>
<ELAPSEDTIME>
Time since the last action
</ELAPSEDTIME>
<TAG> Context </TAG>
<ACTION>
Action defined in "<TOOLSET></TOOLSET>"
</ACTION>
</ENTRY>
...
</COMMANDENTRES>

Fig 6. Body of the description of command actions performed during the session.

Many other collaborative software applications that implement the T.120 protocol, used by both NetMeeting and Polycom PVX exist. Because our objective was to compare a broad spectrum of collaborative approaches suited for radiological discussions and not the applications themselves,

it did not make much sense to compare a larger group of applications using a very similar approach. We have selected NetMeeting because it is present in almost all Microsoft Windows operating systems by default, and Polycom PVX, because Polycom has been on the market for a very long time, and the PVX is an established and well-known application. The other applications were selected to allow the comparison of our approach to a large group of different and widely used proprietary solutions.

## RESULTS

The totality of the protocols and tools which the CycAppDCM protocol was tested against are based on a pixel-transfer strategy using or inspired by the ITU T.120 collaborative protocols family, or on completely closed proprietary strategies. The

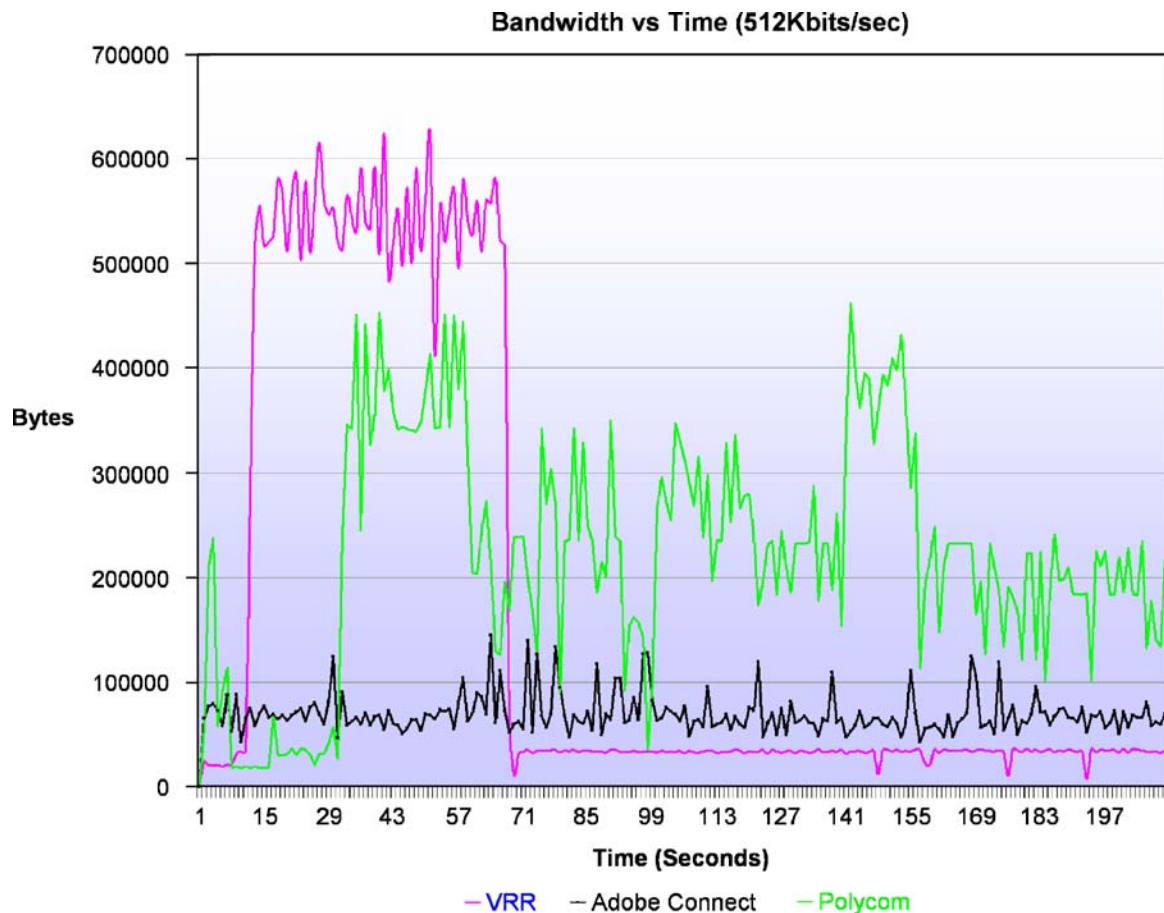


Fig 7. Bandwidth consumption during the collaborative work.



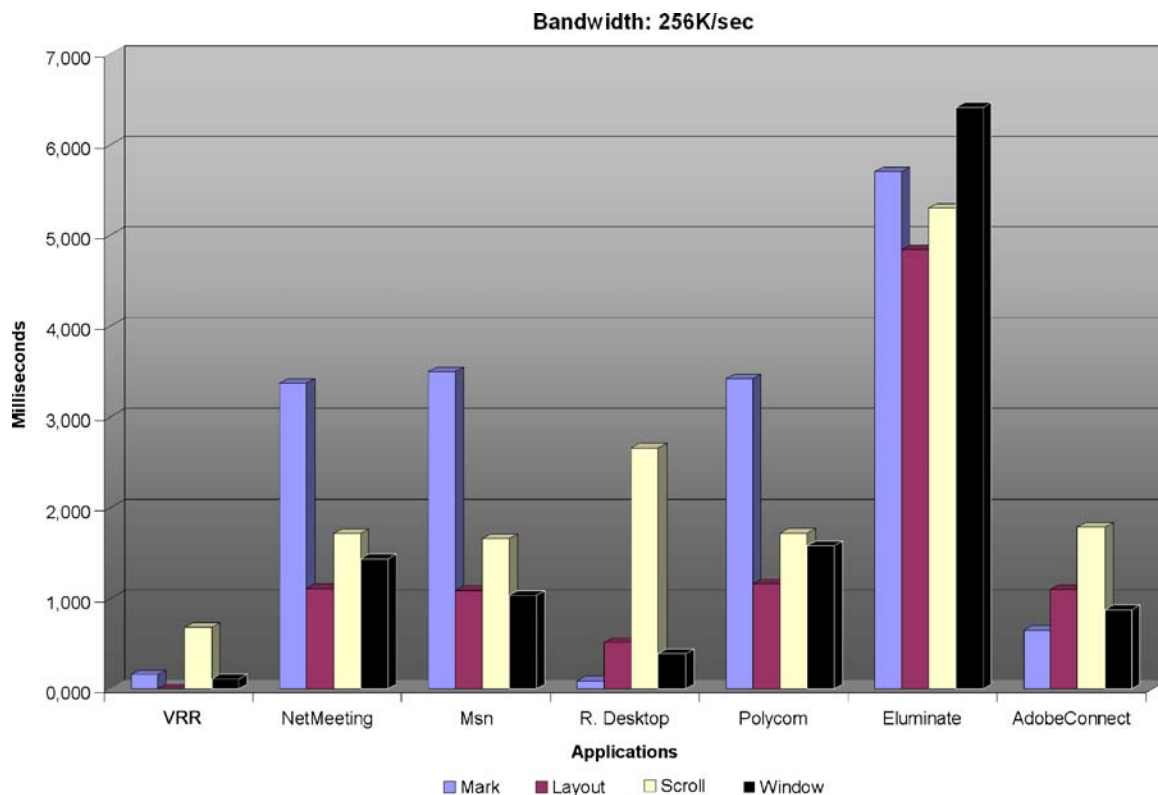
**Table 6. Total Transferred Data Exchanged During the Collaborative Work**

Bandwidth	Transferred data					
	VRR	NetMeeting	Msn	R. Desktop	Polycom	AdobeConnect
256	35,000	44,000	44,000	8,000	45,000	16,000
512	35,000	54,000	58,000	8,000	50,000	16,000
Unlimited	35,000	57,000	64,000	8,000	41,000	16,000

ITU T.120 protocol family follows a strategy that replicates in all collaborating partners the appearance of a tool being run on only one computer of these partners, which is a very simple and extremely bandwidth-consuming strategy. Our approach, on the other side, is a low bandwidth solution, where only the audio stream, the interaction commands, and the presentation-state changing commands are transferred during most of the collaboration, after an initial DICOM image sharing phase.

The quantitative analysis of the validation was carried throughout a careful study of the video and of the network packets captured during the collaborative session. The analysis shows that the first striking difference between the results

obtained with the other applications tested and our protocol is the fact that the CycAppDCM protocol is bandwidth-hungry only during DICOM image data synchronization events (Figure 7), when one or more DICOM volumes are transferred between sides to provide collaborating partners with all data necessary for the planned session, which normally occurs only once during session initialization. The data transferred during the session described in this experiment was a DICOM CT Study of 28 MB, and the collaborative session began only after the data were shared. Table 6 shows the total transferred data exchanged during the collaborative work. In the presence of a larger bandwidth, VRR, Adobe Connect and Remote Desktop have exchanged almost the same amount

**Fig 8. Average values of latency time collected over 256 Kbytes/s.**

of data bytes as when run with a lower bandwidth. NetMeeting and MSN Messenger, on the other side, have become larger bandwidth consumers with increasing bandwidth available. This seems to be mainly because there is an increase in quality performed automatically by these programs. Polycom increased its bandwidth usage when going from a limited 256 to 512 Kbits/s and decreased when using an unlimited Gigabit ethernet network.

Because of the fact that our approach is based on collaboration paradigm that is totally different from that implemented by the applications which it was tested against, a marked difference was found between the values measured of our protocol and those of the other applications. This difference has shown that our application has a great advantage because:

- a. Latency time was much shorter, mainly during the tests performed with limited bandwidth, because much less data is transferred by the CycAppDCM protocol than by ITU T.120-like protocols that transfer all pixels that have changed in the whole

application interface. This can be seen in Figures 8, 9, and 10 where data transfer volumes are represented;

- b. There were no interface representation inconsistencies between partners or un-updated interface patches, as under the CycAppDCM protocol all data is present on all partners and only interaction commands are shared, and each partner application recalculates its presentation locally, independently from network bandwidth availability.

This experiment can only be regarded as an approximate correlation between the tested applications because of the difficulty of evaluating distinct protocols and sharing paradigms, it will suffice, however, as a reasonable basis for comparing radiological collaboration paradigms. Because the ITU T.120–T.128 family and similar protocols are the de facto standards used today, the tools that implement them should be the ones which a new approach should be tested against.

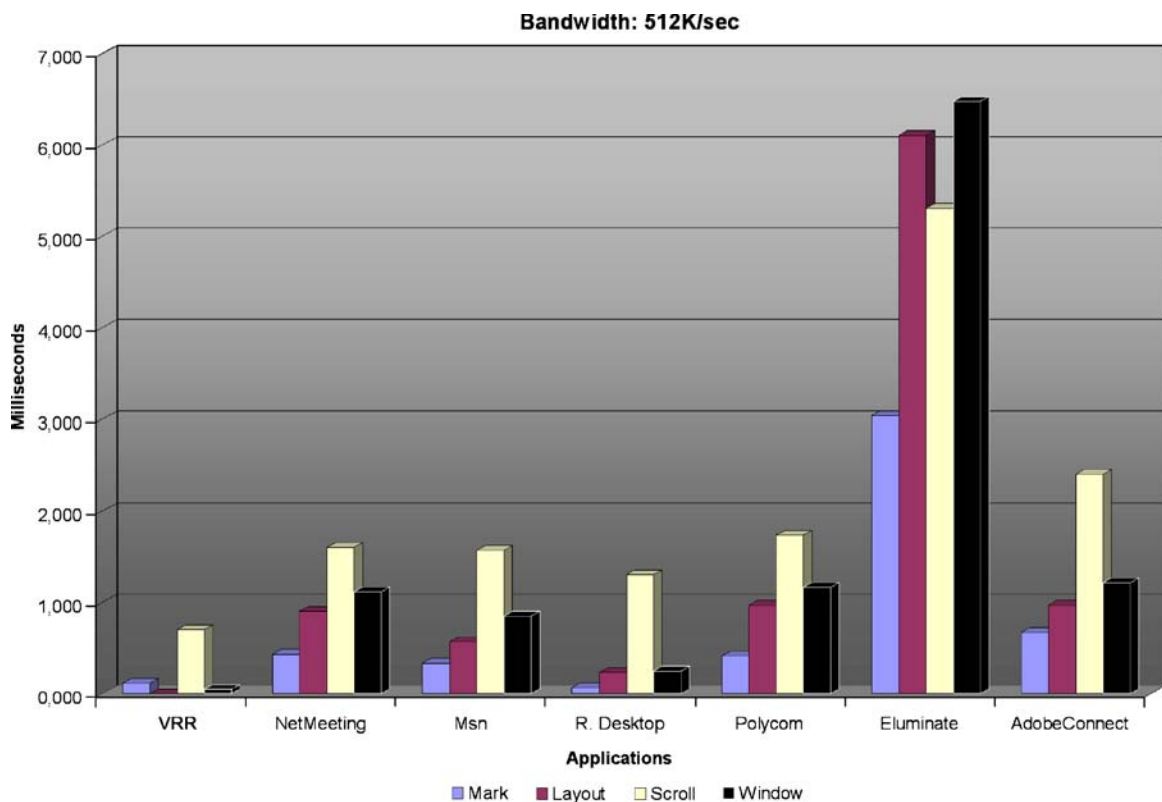


Fig 9. Average values of latency time collected over 512 Kbytes/s.

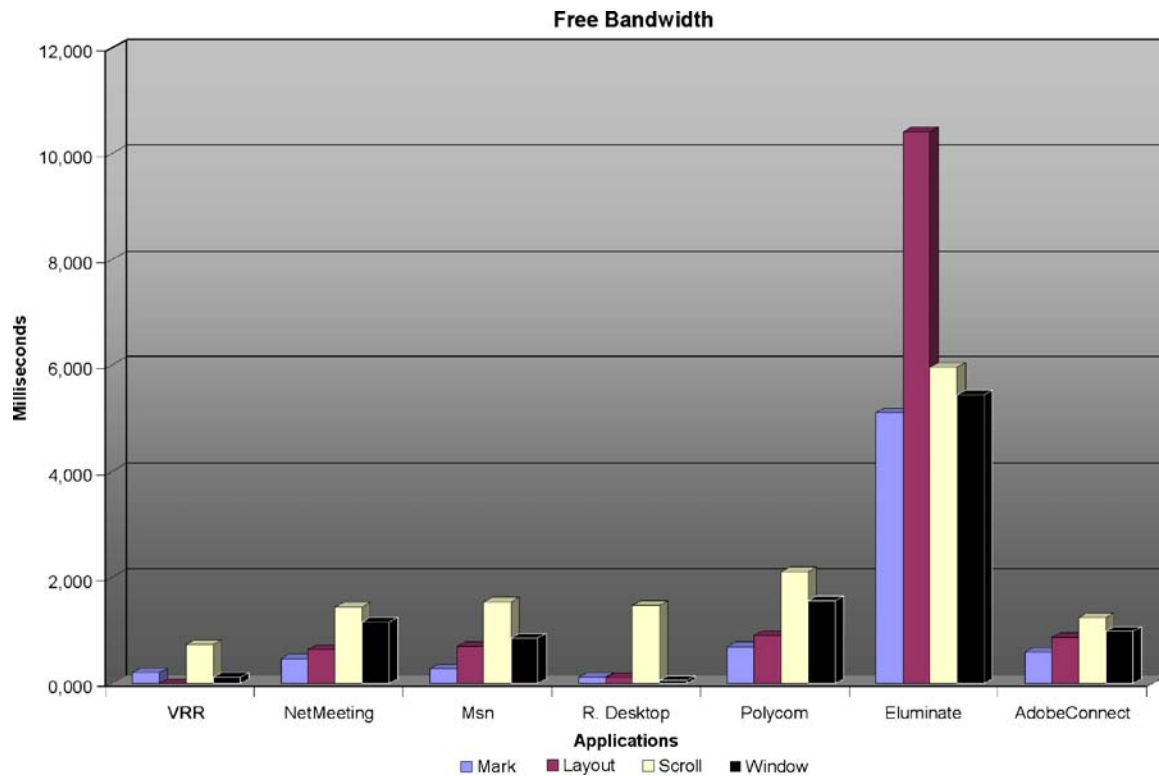


Fig 10. Average values of latency time collected over an unlimited bandwidth.

A second strategy of evaluating the software tools and protocols tested is presented in the Table 7: a feature comparison of the software tools tested. During the tests, there were noticeable image quality differences between the VRR and the other applications, mainly because VRR allows both applications to have the same DICOM images and could, therefore, present exactly the same quality, without any compression algorithms or serious presentation delays. Some applications, such as Elluminate and Adobe Connect, presented images that seemed to have been sent with high compres-

sion rates, improving performance but presenting the user with a picture that was visibly different from the original one. Elluminate has also presented a longer delay than other applications, mainly because their “No User Left Behind” technology guarantees the synchronization between all peers. Delays caused by synchronization were also noticed in the Remote Desktop, including having the outgoing side of the transmission completely disabling user interaction and even stopping the computer to allow the incoming side of the transmission to synchronize.

Table 7. Feature Comparison Between the Applications Tested

Feature comparison							
Feature	VRR	NetMeeting	MSN	R. Desktop	Polycom	AdobeConnect	Elluminate
VoIP	Yes	Yes	Yes	No	Yes	Yes	Half duplex
File transference	Yes	No	Yes	No	No	No	No
Video	No	Yes	Yes	No	Yes	Yes	No
Quality	ideal	Good	good	Good	good	tolerable	Tolerable
Web-Conference	No	No	No	No	No	Yes	Yes
Proprietary	No	No	Yes	Yes	No	Yes	Yes
Session recording	Yes	No	No	No	No	Yes	Yes
Rendering errors	No	Yes	Yes	No*	Yes	Yes	Yes

Equally important is the lack of session recording in almost all collaborative software studied on this validation. Only Adobe Connect and Elluminate presented a feature similar to our recorded session, although, to replay the session, the user must connect to the web server where the online session is stored, and it is not possible to recover the DICOM images if a new diagnosis is required, neither is it possible to another application to read the session, as protocols are proprietary.

What can be considered surprising was how unreliably the T.120-based applications (NetMeeting and Polycom) performed, even when compared to web conference applications, which were expected to be much slower and less reliable. When scrolling fast or changing repeatedly the presentation state of a set of images, whole parts of the application area on the partner side were often left without being updated, resulting in medical images that were presented

with errors. These errors were of great proportions and simple to identify, as shown in Figure 11, but it is easy to imagine situations where more discrete un-updated regions of a medical image could lead to wrong conclusions about the patient's situation (Figure 12).

## DISCUSSION

We presented a new approach to support collaborative teleradiological sessions that, besides addressing latency time and image data inconsistency problems shown by the user interface sharing paradigm implementations, like ITU T.120-related approaches, also presents a novel approach for the recording, storing, reenacting, and sharing of performed collaborative sessions.

The strategy for testing and comparing our approach to other collaborative tools was based



Fig 11. Rendering error using NetMeeting. Some screen areas remain without being updated after presentation state change (changing layout from three-column to two-column) is complete.



Fig 12. Rendering error using Polycom. Some screen areas remain without being updated after presentation state change (changing layout from three-column to two-column) is complete.

on the measurement of only three variables: latency time, bandwidth, usage and image data transmission errors. These parameters were considered to be most significant and user impacting and were possible to be measured adequately in an objective way. Other errors such as image quality, various degrees of jittering and software operation pausing episodes for delayed synchronization were observed for some of the applications. These errors varied broadly in nature, were difficult to measure objectively, and did not occur systematically. For this reason, we did not measure them. To preserve the objective nature of this paper, we decided to avoid describing them in a manner that could be considered subjective.

Our recording strategy has two advantages, when compared to the option of generating a movie of the collaboration session, a possibility presently offered by some tools:

- The recorded session file contains the actual DICOM files of the examinations, ensuring

that there will be no quality loss when a session is replayed;

- When replaying a session, the workstation is actually reenacting the session, as during replay, the radiological workstation is fed with the actual commands that were issued by both sides during the recorded session, and at the same rate they were performed originally. During replay, the workstation only reacts accordingly and provides the corresponding presentation-state changes, measurements, scrolling, and mouse movements on the original data. This allows the user to interrupt the reenacting at any time and to inspect the images by himself, performing any desired additional operations, and then to resume the session replay.

As an application protocol developed specifically for the collaborative analysis of radiological images, the VRR protocol in its present form has some limitations. It lacks the capability to share



other kinds of applications, a feature that is present in the T.120 and in the other application-sharing protocols. Nevertheless, as the VRR is an open high-level application protocol, it could be expanded to allow the sharing of other kinds of data, like text files, and other kind of images. The VRR protocol is an open protocol that has been published<sup>3</sup> and is also available online at <http://www.lapix.ufsc.br/vrr>, allowing other vendors or developers to implement it, free of royalties. A more detailed explanation of the protocol presented in this paper can be seen in<sup>3</sup>.

The MIME64 format for the Session File is not a compact storage format but it is needed, because we include the DICOM and audio files directly within the XML structure, and XML does not provide binary data support. This results in one XVRRL file that contains all the needed information to allow the reenactment of the whole collaborative session on any workstation that implements the protocol. Unfortunately, the binary conversion of the DICOM and audio files to the MIME64 format, increases the size of this files and the overall size of the resulting file. A new approach using file addresses as references to the binary files and the encapsulation of the XVRRL file into a zip file, would probably be better, and will probably be added as an extension to the current format in the future. On the other side, it is important to consider that packaging everything in one file simplifies the manipulation and sharing of the recorded sessions, which would not occur if the DICOM files and each audio stream would be stored in separate files in some directory structure that would have to be collected by hand before one could share it.

Finally, we believe that the possibilities offered by this approach to collaborative session recording have a potential impact on education, telemedicine, and auditing, as not only discussions between experts could be made available for others to watch and interact with. The system allows the recording of an interaction with an examination even if there is no any collaborative session active, allowing the recording of a lecture in radiology or of an auditable multimedia findings report for a telemedicine service.

## CONCLUSIONS AND FUTURE WORK

We developed a new approach for the execution and recording of collaborative teleradiology sessions, based upon an application layer network

service protocol, which has the potential to be integrated into any radiological workstation. From the technical point of view, this approach was validated against state-of-the-practice industrial standards and has shown several advantages in data integrity, performance, and bandwidth usage when compared to commonly used collaborative tools.

The CycAppDCM protocol was developed intending it to be an open protocol. This naturally leads to the question if a standardization of the protocol could be part of the future work, perhaps through the DICOM Standards Committee, as the protocol is based on the sharing of DICOM data, and to define it as an extension to the DICOM standard would be a natural development. In this context, our first step, after the publication of the core of the protocol in<sup>3</sup>, was to perform and publish a comparison study showing how it behaves when compared to state-of-the-practice standards and describe extensions that would allow the recording and reenactment of collaborative sessions, which was accomplished through this paper. Now, we expect to receive feedback of the community, which will be used in the planning of further actions.

From the usability and applicability point of view, this approach opens several new possibilities in telemedicine and medical education, broadening the applicability of teleradiological collaborative sessions. The professional, ethical, and cultural implications of this development have yet to be analyzed in depth. For this purpose, a multi-centric usability study involving several institutions of the RUTE—Brazilian University Telemedicine Network<sup>13</sup> is being structured. After incorporating the recommendations of this study will into the Virtual reading Room, our intention is to perform a broader international study.

## REFERENCES

1. NEMA, Digital Imaging and Communications in Medicine—DICOM, available at <<http://medical.nema.org>>.
2. Duerinckx AJ, Pisa EJ. Filmless Picture Archiving and Communication System (PACS) in Diagnostic Radiology. *Proc SPIE* 1982;318:9–18. Reprinted in *IEEE Computer Society Proceedings of PACS'82*, order No 388.
3. ABDALA, Daniel; PRUSSE, Martin; REGERT, André; v.WANGENHEIM, Aldo. Application Protocol for a DICOM Real Time Collaborative System. In: *Proceedings of the 19th IEEE Symposium on computer based medical sySTEMS — CBMS2006*. Los Alamitos: IEEE Computer Society, 2006. p. 490–494.

4. Cyclops Medical Station. *Official Site*, available at <<http://www.telemedicina.ufsc.br/cms>>.
5. M. G. Gouda. Elements of Network Protocol Design. John Wiley & Sons, Inc., 1998.
6. A. Y. Seydim, "Voice Over IP (VoIP)", Lecture notes, Digital Telephony Course, Southern Methodist University, December 1999. <http://citeseer.ist.psu.edu/seydim99voice.html>.
7. I. Graham, L. Quin, XML Specification Guide, Wiley Computer Publishers, New York, 1999.
8. Network Working Group (1996), *Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies*, available at <<http://tools.ietf.org/html/2045>>.
9. Virtual Reading Room. *Official Site*, available at <<http://www.telemedicina.ufsc.br/slv>>.
10. Postel, J. B., *Transmission Control Protocol*, available at <<http://www.faqs.org/rfcs/rfc793.html>>.
11. T.120, Data Protocols for Multimedia Conferencing, ITU-T Recommendation, 1996, available at: <http://www.itu.int/rec/T-REC-T.120-200701-I/en>.
12. T.128, Multipoint application sharing, ITU-T Recommendation, 1998, available at: <http://www.itu.int/rec/T-REC-T.128-199802-I/en>.
13. RUTE—Brazilian University Telemedicine Network, available at <<http://rute.rnp.br>>.