uOttawa

L'Université canadienne
Canada's university

FACULTÉ DES ÉTUDES SUPÉRIEURES ET POSTOCTORALES

uOttawa

L'Université canadienne
Canada's university

FACULTY OF GRADUATE AND POSDOCTORAL STUDIES

Gaofeng Liu
AUTEUR DE LA THÈSE / AUTHOR OF THESIS

M.Sc. (Systems Science)
GRADE / DEGREE

Department of Systems Science
FACULTÉ, ÉCOLE, DÉPARTEMENT / FACULTY, SCHOOL, DEPARTMENT

A Flash-based Multimedia Whiteboard for Clinical Motion Analysis

TITRE DE LA THÈSE / TITLE OF THESIS

Dr. Ed Lemaire
DIRECTEUR (DIRECTRICE) DE LA THÈSE / THESIS SUPERVISOR

CO-DIRECTEUR (CO-DIRECTRICE) DE LA THÈSE / THESIS CO-SUPERVISOR

EXAMINATEURS (EXAMINATRICES) DE LA THÈSE / THESIS EXAMINERS

Dr. Shervin Shirmohammadi

Dr. A. El Saddik

Gary W. Slater
Le Doyen de la Faculté des études supérieures et postdoctorales / Dean of the Faculty of Graduate and Postdoctoral Studies

A Flash-based Multimedia Whiteboard for Clinical Motion Analysis

Gaofeng Liu

Thesis submitted to the

Faculty of Graduate and Postdoctoral Studies

In partial fulfillment of the requirements

For the MSc degree in Systems Science

System Science Department

University of Ottawa

Your file Votre référence
ISBN: 978-0-494-49238-3
Our file Notre référence
ISBN: 978-0-494-49238-3

Abstract

Due to persistent shortages of rehabilitation resources and services, collaboration tools for motion analysis have received great attention. High quality Internet connectivity and rich-content Web application platforms - Adobe Media Server and Flash Player, have created opportunities for integrating motion analysis into Flash based whiteboard and video conference applications to enhance rehabilitation services. For this thesis, clinical motion analysis software was created using Flash. The Motion Analysis Tools runs directly from a URL with a Web browser. The video synchronization procedure was supported by shared objects on a Flash Media Server. Program evaluation was conducted from accessibility and quality perspectives. Video synchronization error evaluation, angle and distance measurements, whiteboard functional, network latency impact, and ease of access were discussed during the evaluation phase. The Motion Analysis program can be access via Real-time Media Protocol (RTMP). A single TCP port was used for both signaling and media to maximize the chance of establishing a connection. Synchronization error was maintained at an acceptable level (maximum 160ms, average 49.3 ms, standard deviation 0.21). Average root mean square errors were 2.7 degrees for angle measures and 1.95 cm for length measures. The test results were consistent with video motion analysis results and within an acceptable range for rehabilitation motion analysis. This Web-based motion analysis approach provided an enhanced tool for quantitative motion analysis.

Acknowledgements

I would like to express my sincere thanks and appreciation to my thesis advisor Dr. Edward Lemaire. Without him, this dissertation would not have been possible. I thank him for his patience and encouragement over the past four years that carried me on through difficult times and for his insights and suggestions that helped to shape my research skills. His valuable feedback contributed greatly to this thesis.

I thank Fang Wei Zeng, who helped me in various aspects of my programming, and other stuffs of the Institute for Rehabilitation Research and Development (IRRD) at the Rehabilitation Centre (Ottawa), their support was very helpful for my research.

I also thank my boss Bill Crago, Ted Gruszecki, and Elias Psarras in Bell Canada. Thanks for their encouragement and support over the past three years.

Last but certainly not least, I would like to thank my wife Yu Ting for her love and encouragement.

Table of Contents

Figure List

Table List

Abbreviations List

AMF	Action Message Format
CCITT	Comité Consultatif International Téléphonique et Télégraphique
FLV	Flash Video
HTML	Hypertext Markup Language
HTTP	Hyper Text Transfer Protocol
HTTPS	HTTP Over SSL
IP	Internet Protocol
IIS	Internet Information Server
ISAPI	Internet Server Application Programming Interface
ISDN	Integrated Services Digital Network
ITU	International Telecom Union
JRE	Java Runtime Environment
MAT-F	Motion Analysis Tools Flash Version
OOP	Object Oriented Programming
PC	Personal Computer
POTS	Plain Old Telephone Service
RMSE	Root mean square error
RTMP	Real-time Messaging Protocol
RTMPT	Real-time Messaging Protocol Tunnelling via HTTP
RTMPS	Real-time Messaging Protocol Tunnelling via HTTPS
SDK	Software Development Kit

SMTP	Simple Mail Transfer Protocol
SOAP	Simple Object Access Protocol
SSL	Secure Socket Layer
TCP	Transport Control Protocol
UDP	User Datagram Protocol
VC	Video Conferencing
XML	eXtensible Markup Language

Chapter 1. Introduction

More and more people are concerned about access to health care due to persistent shortages of resources and services distributed across geographical locations. To address gaps in the health system, collaborative care via telehealth has received great attention.

Collaborative care provides health dialog and medical decision making across distances via computer networks [48]. In order to avoid downgrading health service quality during collaboration care, an efficient collaborative care platform must meet the health service requirements. These requirements vary between healthcare fields.

In physical rehabilitation, patient motion analysis is an important activity during online collaborative care/consultations. Enhanced image/video sharing and quantitative analysis functions facilitate human motion collaboration in the physical rehabilitation sector [1]. These functions include angle calculation, distance measurement, timing, and velocity calculations.

A number of stand-alone data conferencing systems have Whiteboard applications. While these Whiteboard applications typically include basic graphical/data communication functions, these functions do not handle video. Video integration and motion-specific analysis tools are required to improve communication during physical rehabilitation on-line collaborations. [1]. No Web-based motion analysis tool with video sharing and motion analysis functions has been published for the physical rehabilitation field.

Most telehealth Whiteboard components are not web-based applications. For example, the Tele-Medicine Benchmark (TMB) application was a stand along application developed to share X-ray/CT image in a Whiteboard for joint analyses, image viewing from an optical microscope, and sharing patient records [3]. These stand-alone applications were typically written for a specific operating system and are individually install on each system, involving system and network configuration and storage space consumption. This may create difficulties for inexperienced computer users.

In addition, some data conferencing systems built on Internet technology have trouble passing through firewalls. For example, Microsoft NetMeeting was one of the most popular data conferencing tools on the Internet. Since NetMeeting uses TCP and UDP ports (522, 389, 1720 and 1731 plus two secondary dynamically negotiated UDP ports in the range 1024-65535) to communicate between Server and clients, NetMeeting rarely passes through modern corporate firewalls [4]. In Canada, almost all health organizations have an enhanced security policy on Internet management [5]. Making a successful connection between a conferencing server and a client workstation is another challenge for Web based motion analysis systems.

Web-based applications do not require extra installations on workstations and can work with a variety of system platforms. Control and Media messages are transmitted in HTTP/HTTPS protocol, which is allowed on most computer networks. Therefore, a Web-based quantitative analysis system is a good option to meet the requirements of easy access in the physical rehabilitation field.

For this thesis, clinical motion analysis software was created, using Flash, that can load and share Flash Video format video clips of a client's motion from different locations. Features included play, pause, forward, rewind and stop controls, angle and distance measurement on the whiteboard stage, and velocity calculation. With light-weight Flash Player plug-ins, the Motion Analysis Tools—Flash program runs directly from a URL with a Web browser. The video synchronization procedure was supported by shared objects on Flash Media Server. With the online quantitative analysis tools, clinicians can share a patient's movement video or image clips, do analysis by slowing down or performing step-by-step movements, and make decisions with support from other senior health professionals who are in different locations.

The first chapters of this thesis include a review of the literature in the data conferencing and telehealth fields and a statement of the objectives and rationale. Motion Analysis System architecture is discussed in the methodology chapter, as well as the detailed methods utilized during development. Methods and results chapters describe the evaluation of system performance and quality; including, network connection parameters, server performance, motion analysis tool functionality, and web browsers compatibility testing. Subsequent chapters discuss the results and recommend future work in this area. This thesis is a step towards a complete mobile, Web-based, online physical rehabilitation consulting system.

Chapter 2. Review of the Literature

2.1 Search Strategy

Computerized literature searches before Dec 2006 were performed using the MEDLINE, EMBASE, CINAHL, Ei Compendex*Plus, EBM Reviews, IEEE Explorer, Google Scholar and ProQuest databases. Search keywords included 'data conferencing', 'video conferencing', 'telemedicine', 'telehealth', 'e-health', 'web conference', 'telerehabilitation', 'data transfer', 'application sharing', 'whiteboard', 'motion analysis'. In addition to this search strategy, references cited in relevant publications were reviewed. Sections of the review are extracted from a review paper by the thesis author [6].

The following inclusion criteria were applied: the study involved data conferencing activities in health consulting; the study involved data conferencing activities in health education; review articles in health on data conferencing field; English language. The following were excluded: studies focused on video conferencing between patients and their family; studies not involving healthcare activities; articles that were similar to the same author's other published studies (only the most representative of the studies was included for further consideration).

From the publications identified in the literature search, 25 articles were retrieved for closer inspection. Ten articles provided detailed data conferencing applications. Of these, seven articles focused on Whiteboards, eight articles focused on Application Sharing, two articles focused on Chat, and one articles focused on screen sharing.

2.2 Data Conferencing Technologies

Data conferencing is among the fastest growing technologies and is a ubiquitous communication tool. Typical collaboration tools include shared Whiteboard, Chat, Application Sharing, Slide Presenting, Screen Sharing, Webpage Co-browsing, and synchronous work on documents. Data conferencing has been used for workgroup collaboration, meeting with work partners, technical support, and remote device control.

The ITU T.120 standard was developed to promote interoperability between data conferencing systems. This standard was ratified in 1996 and includes a set of standards for moving data in real time between computers [1]. These standards provide the audiographic portion of the H.320, H.323 and H.324 families, and define collaborative document sharing and Whiteboard activities. This standard provides a base for:

- **Multi-point data sharing**: data is automatically delivered to multiple parties in real-time.
- **Interoperability**: between vendors and between networks.
- **Reliable data transfer**: ensures all participants receive data in the correct sequence.
- **Scalable, transparent, and independent**: the application is separated from the data transmission method and is functional on most networking configurations (i.e., a Whiteboard application is not concerned if one user has a dial-up connection and another has a broadband connection).
- **Platform independence**: the standard can be applied to any computer system.

- **Application independence**: standard data communications can be applied to a wide range of data driven tasks (virtual reality, simulation, real-time event notification, etc.).

Several companies make T.120 data conferencing software for Microsoft Windows, MacOS, and various UNIX platforms. Microsoft NetMeeting is a free Windows implementation of this standard [13] (Figure 1). NetMeeting is well established but requires software installation on each local machine. NetMeeting 3 uses TCP port 522, 389, 1720 and 1731 plus two secondary dynamically negotiated UDP ports in the range 1024-65535 for the H.323 streaming protocol transmission of audio and video [4]. NetMeeting also uses TCP 1503 port for data conferencing [14]. Since these ports are often blocked by firewalls, NetMeeting has not been widely adopted in the hospital sector.

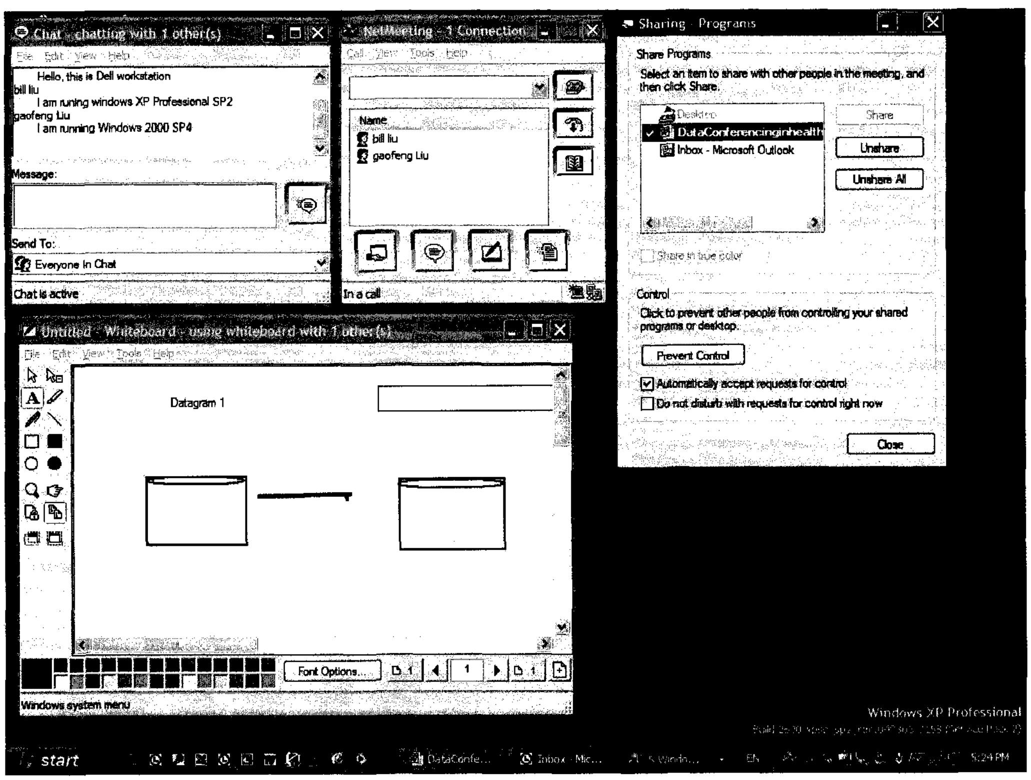


Figure 1: Microsoft NetMeeting

Web-based conferencing systems avoid many firewall issues by using typical web page communication ports. Most of these products use Java applets or Macromedia Flash in a web browser to provide functionality. Some products use HTTP Tunnelling to pass through firewalls [15]. Products such as ClicktoMeet / CUseeMe, WebEx, V-Stream, IMeet, and Lotus Web Conferencing support both web delivery and T.120 clients (Figure 2). Some web-conferencing approaches use proprietary communication technology and are not interoperable with other data conferencing systems.

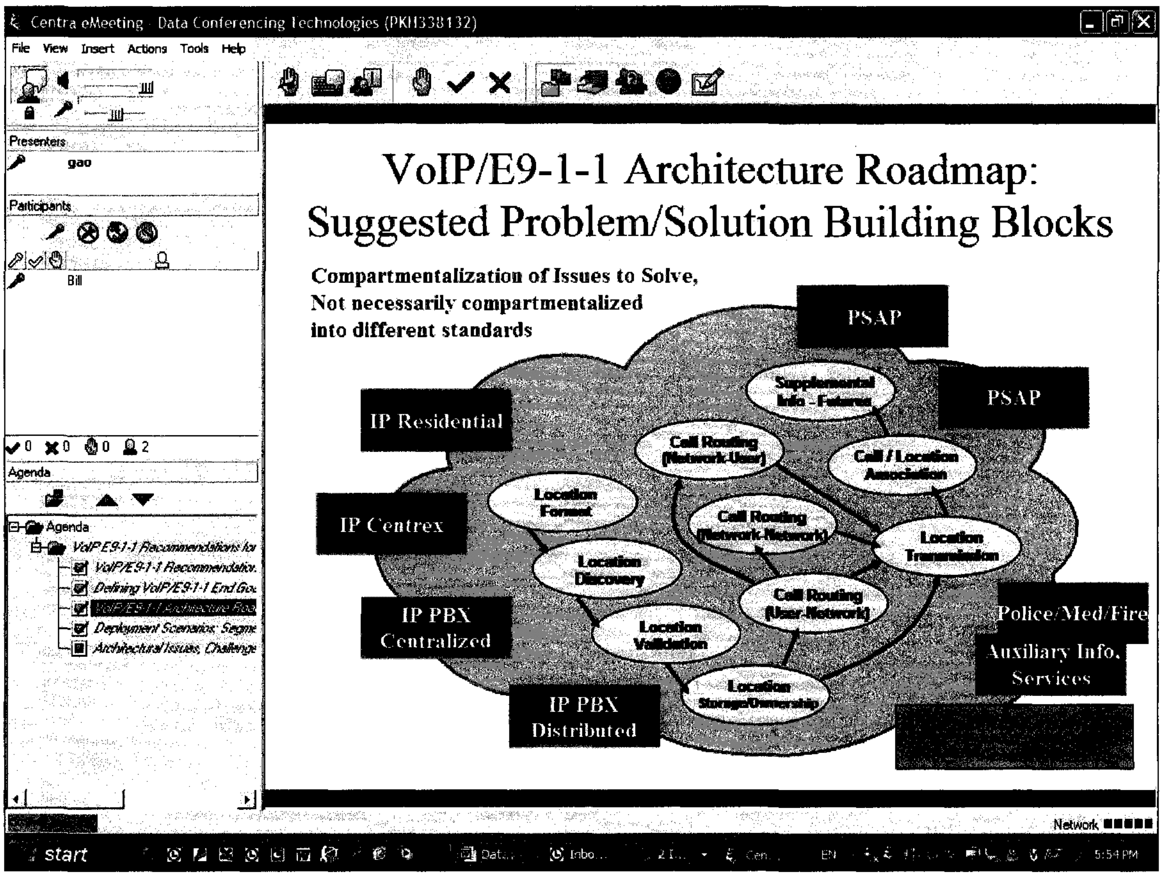


Figure 2: Centra eMeeting Data Conferencing

2.2.1 Text-format Communication

Text-format communication (i.e., Chat, Instant Messaging) is a simple data conferencing function where participants type their comments using a keyboard. The greatest advantages of Chat are low bandwidth requirements and the ability to save text discussions for review purposes. Information can also include hyperlinks to relevant health documents, text files, image/graph, or other multi-media files. Text lacks the rich dimensions of multimedia communication and can inhibit spontaneity due to the user's typing speed. Emoticons are special graphical symbols that are sometimes available to portray feelings or social behaviours quickly. Popular Chat or instant message tools provide emoticon support; including, Microsoft Messenger, Yahoo Messenger, AOL Instant Messenger, iChat, Trillian Messengers, and ICQ (Table 1). Users can also customize personal emoticons during text-format communication [16]. For health communications, standard graphical symbols could be used to increase communication efficiency in text-based applications.

Table 1: Chat Tools

Product	URL	Features
Microsoft MSN Messenger	http://messenger.msn.com/	Text, Emoticons, Display Picture, Customizing/Sharing Background, Mobile Access
Yahoo Messenger	http://messenger.yahoo.com/	Text, Emoticons, Customizing/Sharing Background, Avatars, Mobile Access
AOL Instant Messenger	http://www.aol.ca/aim/index_eng.adp	Text, Emoticons, Mobile Access
iChat	http://www.apple.com/macosx/tiger/ ichat.html	Text, Emoticons, Automator
Trillian Messengers	http://www.bigblueball.com/im/ trillian/ default.asp	Text, Emoticons, Customizing/Sharing Background
ICQ	http://www.icq.com/	Text, Emoticons, Customizing/Sharing Background

Since Chat applications are relatively easy to develop and require minimal bandwidth, almost all data conferencing systems have Chat functionality. A good example is the web-based Chat used with the CoMed system [30]. This Chat tool was built with Java and managed by a web server. Health professionals used Chat to exchange disease information between Seoul, Korea and Cheju Island. The program performance achieved the design target but needed to improve information security since the application ran on the public Internet.

2.2.2 Whiteboard

Whiteboard is a real-time, multi-point, graphical communication tool (Table 2). Participants at different locations simultaneously write, draw, or upload files onto an on-screen notepad viewed by all conference participants. The collective drawings then become available to store or print. Whiteboards provide a visible work environment for all connected sites. When a participant adds or modifies the Whiteboard contents, all conference displays are immediately synchronized to show the current view. The Whiteboards embedded in NetMeeting, WebEx, iMeet and Horizon also provide copy, paste, and file upload [15-17]. As well as viewing images/text/annotations, many Whiteboards include dynamic communication tools; such as, pointers and user tracking. After images have been loaded onto a Whiteboard, minimal bandwidth is needed for live graphical communication.

SMART Board / TeamBoard is a wall-size interactive Whiteboard that works with a computer and projection system (Figure 3). Touch-screen control is used to operate the computer and pen functions for drawing or writing on the SMART Board. SMART

Boards can also transfer handwriting to text and save the content as a word processor file. Using "Writing over Applications", users can add emphasis or additional information to Microsoft office applications using the built-in pen tools. The additional text or drawings can be saved in the existing document or be erased. SMART Board also can record onscreen activities as a movie file [19]. While typically applicable in a classroom or boardroom environment, SMART Board technology could be of use in healthcare situations where touch screen annotations can improve group communications (emergency room diagnostic image discussions, etc.).

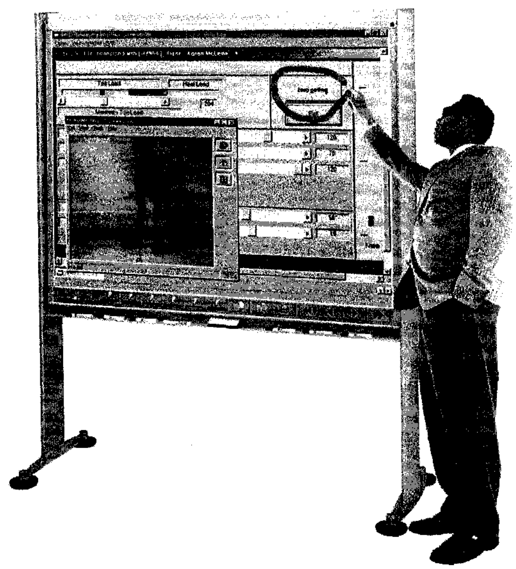


Figure 3: SmartBoard.

In 1999, the Tele-Medicine Benchmark (TMB) application was developed to share X-ray/CT images in a Whiteboard. A region of interest and a line were drawn onto the image, telepointing was performed, and the image was scrolled. Finally, a coloured illustration and a snapshot of an application window were imported into the Whiteboard [3]. This application made collaborative analysis of health data possible. The application was used for joint analyses, image viewing from an optical microscope (i.e., microscope images copied to the Whiteboard), and sharing patient records. While no formal evaluation was performed, the authors indicated that the TMB Whiteboard was an excellent tool for 1-D, 2-D and 3-D joint analysis in telemedicine.

In 2000, a web-based health application was developed for health data sharing (CoMed) [30]. CoMed used a Whiteboard component to share patient X-ray images between Seoul National University Hospital and Cheju Provincial Medical Centre. The Whiteboard enhanced communications between health professionals through visual, real-time, two-way data transmission [31]. After six months validation testing, the results showed that the Whiteboard was accessible from any Internet connection and was an effective visual environment for health image collaborative analysis.

A virtual Whiteboard was built into an existing clinical information system, WebCIS, at New York-Presbyterian hospital (NYPH) [32]. The Whiteboard was used for posting, routing, and tracking communications among nurses and physicians to address deficiencies in current channels used to coordinate care. For this specific function, the Whiteboard was designed to only transfer text-format information. The authors believed that the application would contribute to improving access to information as well as communication and collaboration among and between physicians and nurses. These improvements would ultimately reduce the causes of medical errors and adverse events. An evaluation of this application in clinical settings was planned in order to determine the impact.

The Microsoft NetMeeting Whiteboard transmission speed has also been evaluated within a healthcare environment [3]. Klutke et al. measured Whiteboard transmission speeds by sharing DICOM images with remote Whiteboards. Time delay measurements varied following changes in the data compression approach and changes in the available bandwidth. These measurement results can help system designers decide whether a Whiteboard can efficiently share health images based on computer performance, codec, and network connection.

Whiteboards have also been used in health education for graph annotations within an e-class environment [32]. Physician lecture materials, including images, text and graph annotations, were posted on a proprietary Whiteboard area in real-time. Students could also send their feedback by drawing on the Whiteboard. The shared Whiteboard allowed students and physicians to communicate in a visual setting and to understand each other by sharing visual information. The investigators indicated that this application made medical distance education more possible and efficient, although no formal evaluations were reported.

Lemaire et al. [33-34] used the NetMeeting Whiteboard to support their telerehabilitation activities. Technical and clinical evaluations supported the use of the Whiteboard to discuss patient images; such as, assistive device interfaces, seating positioning, and wound status. The Whiteboard was also effective for Speech Language Pathology tests that involved a set of images. While useful over low-bandwidth connections, large images could take over seven minutes to display at remote sites. This image display lag is less of a factor when connecting over a broadband link. Once the image is on-screen, live mouse movement and annotations are effective over a low-bandwidth connection.

2.2.3 Application Sharing

Application Sharing is another import tool within data conferencing that enables conference participants to run the same application simultaneously. Application Sharing is achieved by sending a constant stream of program window images, mouse position and click information, and keyboard activity to all conference participants. The application itself resides on only one of the conference machines, called the host. A host shares the application with the other partners, the guests. A guest's screen displays the same view as the host's screen and allows interaction as if the application was running on the guest's computer [20.21]. Commercial systems include GRCLive [22], ClickToMeet [23], HelpMeeting [24], and Lotus SameTime [26]. Multiple clients can operate one program and share the screen in real-time. Application Sharing provides more features than Whiteboards, while requiring more bandwidth. Open source software for application sharing is based on the VNC (Virtual Network Computing) project [26, 27]. The VCN software is not T.120 compatible but does provide higher screen refresh rates that increase the usefulness for healthcare-related tasks. Current Application Sharing approaches cannot meet video sharing requirements since the Application Sharing screen image transfer refresh rate is lower than acceptable video rates (usually 15 to 30 time per second).

Application sharing can be used in healthcare for multimedia file sharing, remote device configuration, and 3D image analysis. In 2001, a teleradiology data conferencing system was developed by University of Johann Wolfgang Goethe University, Germany. The system was composed of a DICOM viewer and Microsoft NetMeeting. Application Sharing was used to share the DICOM viewer eFilm application [21]. Remote partners could access a local machine to run the eFilm application and the result was delivered to multiple-points over a 128Kbit/s ISDN connection. The results indicated that images with a wide distribution of grey values and NetMeeting-compression were less effective than images with a simple distribution of grey values (the simplest is black and white). Transmission of images with a wide range of grey values also took longer than black structures.

A video conferencing system developed by the National Research Centre for Environment and Health, Germany and Istituti Oropedici Rizzoli, Laboratorio di Tecnologia Medica (Italy) [21] incorporated an Application Sharing component for epidemiology and orthopedics/radiology research. Intel ProShare was used as the data conferencing platform and a 128Kbit/s ISDN line was used for the data connection. Joint viewing of statistics and medical images was performed using the Application Sharing tool. Statistical, spreadsheet and presentation software were shared for epidemiology tasks. For orthopedics/radiology, a DICOM medical image viewer was shared. Questionnaires results indicated that 53% of the users feedback was good, 31% was sufficient, and 3% was unacceptable. Another article from the same group [18], using ProShare 200™ v2.0, found minimal delay dependence from different file formats. The reason for this low delay was that the only a part of the local screen, the one that displays the shared application, was transferred. Theoretically, this mechanism could allow sharing and visualization of images remotely without any time delay dependence from file formats.

Microsoft NetMeeting has also been used frequently for Application Sharing in health. For physical rehabilitation, Microsoft NetMeeting was used for lower extremity prosthesis control and configuration [36]. The Application Sharing tool was used over 56Kbps, DSL/Cable (300 Kbps), and 10Mbs LAN network connections. Through the tool, health professionals remotely configured the Otto Bock C-Leg prosthesis. A 28Kbps connection was also tested; however, NetMeeting Application Sharing was not able to update the remote software window fast enough to display dynamic bar graphs of peak load and knee angle.

In 1999, another health data conferencing system utilizing Microsoft NetMeeting was deployed in Switzerland. The system used the Swiss academic network (155Mbps) and dial-up (56Kbps) as data connections. CT, VOIs and 3D dose distribution image files in PAPYRUS diagnostic image standard format were shared using Application Sharing. The collaborative environment included a number of clinical centres and a specialized research facility. The ability to exchange health images in real-time was found to provide not only great potential for remotely assessing patients and treatment planning, but also to provided an efficient dialogue between the referring and treatment centres. The author believed that Application Sharing can play a major role in the development, assessment, and efficient utilization of new treatment modalities on radiation oncology. [37]

2.2.4 Volume Data Analysis

Volume data analysis (VDA) is a data conferencing tool for remote analysis of 3D images [3]. 3D image analysis is a demanding task with large data sets that may result in large time delays when transferring and displaying 3D images. VDA overcomes these problems by using Application Sharing. The volume data is stored only one partner's computer and the CPU of this partner's computer performs the 3D operations. Since the on-screen display is in two dimensions (2D), only 2D data has to been transferred by the Application Sharing tool. Reducing the amount of information makes 3D remote image analysis possible since only the host computer requires the power to manage 3D model rendering. VDA has potential for interactive discussions using 3D models generated from MRI, CT or similar diagnostic images.

Volume data analysis is a process of systematically applying statistical and logical techniques to describe, summarize, and compare huge amounts of data. Since health images can have huge file sizes, transferring these uncompressed files results in a great time delay (20 CT images with 512×512 pixels could result in 10 Mb of data [29]. By using a 3D visualization application, interactive working on 3D medical image data assets becomes possible. The huge amount of data is stored only on the host machine and only the host machine performs the 3D operations. The 3D operations are rendered as 2D information for viewing and examination on the remote screen. Only 2D information has to be transferred by application sharing tool. The reduction of the amount of information to be transferred from 3D to 2D makes application sharing feasible for 3D remote image analysis.

2.2.5 Screen Sharing

Screen sharing is a simple yet powerful approach for collaborative visualization within a data conference. The best-known products are VNC (multi-platform) [28] and Microsoft NetMeeting. With the proper settings, these applications can allow all users to see a shared screen or web page but not allow guests to perform any operations (i.e., one-way interaction). Successful implementation of this approach relies on efficient screen frame compression for efficient transmission to a number of viewers, even with frequent screen changes (although too frequent changes reduce the performance considerably) [29]. Screen sharing has the same difficulty as Application Sharing since lower screen refresh rates could cause low-quality video quality on client side, thereby failing to meet physical rehabilitation motion analysis requirements.

Table 2 Collaboration tools and their characteristics (C=Chat, WB=Whiteboard, SS=Screen Sharing, AS=Application Sharing, SW=Simultaneous Work on Documents)

Products	**URL**	**Features**					**Standard**
		C	WB	SS	AS	SW	
GRCLive	http://web.grclive.com/newgrc/index.htm	√	√	√			H.323, T.120
Centra 7™	http://www.centra.com/products/index.asp	√	√	√	√	√	H.323, T.120
Click to Meet™	http://www.fvc.com/eng/products/index.htm	√	√	√	√	√	H.323, T.120
HelpMeeting	http://www.helpmeeting.com/support.htm	√	√	√		√	H.323, T.120
PlaceWare	http://main.placeware.com	√	√	√	√		H.323, T.120
Web-4M	http://www.jdhtech.com/pages/whyweb4m.html	√	√	√			H.323, T.120
NetMeeting	http://www.microsoft.com/windows/netmeeting/	√	√	√	√	√	H.323, T.120
SameTime	http://www.lotus.com/products/product3.nsf/wdocs/homepa	√	√	√			H.323, T.120
CVW	http://cvw.sourceforge.net/cvw/info/CVWOverview.php3	√	√	√			H.323, T.120
infoWorkSpace	http://www.ezenia.com/services-support-iws.php	√	√	√			H.323, T.120
TeamWave	http://www.ezenia.com/services-support-iws.php	√	√	√	√	√	H.323, T.120
WebEx V-Stream	http://www.webex.com	√	√	√	√	√	
iMeet	http://www.netspoke.com/	√	√	√	√	√	
Horizon Live	http://www.horizonlive.com	√	√	√	√	√	

Chapter 3. Objectives

3.1 Primary Objective

Develop a web-browser-based, multipoint conferencing Whiteboard component for collaborative clinical motion analysis. The enhanced Whiteboard component should have the following functions:

- Basic functions: free hand drawing, drawing move, line up, text input, erase, line up with arrow, square and oval, colour pickup bar, line type pickup bar.
- Video control: video file upload, video file play, video file pause, video file stop, video file forward frame by frame, video file rewind frame by frame, sync / async switch.
- Motion Analysis: angle calculation, distance calculation, scale factor setup, duration calculation.

3.2 Secondary Objectives

- Integrate the Whiteboard into a desktop conferencing collaborative environment, including chat and audio/video conferencing features.
- Evaluate how the new conferencing system effects the server and IP network environment including Network connection evaluation from varied locations, and server performance evaluation.

Chapter 4. Rationale

4.1 Communicate More Efficiently

The field of physical rehabilitation will benefit from real-time collaboration and face-to-face communications between clinics, health professionals and patients over a broadband network. With an advanced Whiteboard application, people can share video clips and complete analyses together. All participants can see the results in real-time. With video cameras at the client sites and a robust network in between, participants can see as well as hear others during the conference. A Chat application provides a means for participants to communicate detailed data in a written format.

4.2 Reducing time and financial cost

A health collaboration tool using Internet and computer technology could provide health consulting over a distance. Eliminating distances is a central theme of telehealth. A powerful telehealth system would provide access for health workers in rural and remote areas to specialists who reside in a major health centre. Both patients and local health employees benefit from timely expert's advice, but without the time delay, financial burden, and travel.

4.3 Friendly user interface and delivery to employee's desktop/portable

A telehealth system benefits from with a user friendly graphical interface. The Web browser graphical content interface has smooth navigation and also allows users to customize their preferences, including fonts and color. Built-in function descriptions provides users with brief explanations on how to use function buttons and enter proper data, without having to consult help documentation. Descriptive error messages enable users to quickly identify invalid operation or data entry errors.

A user interface running on a Web browser can provide access for any desktop or portable device that has a web browser running and a network connection. Using a familiar hardware and software interface, health professionals could decrease learning time and increases user acceptance.

4.4 Maximizing the chance of a successful connection

Commonly, computers in health centre are behind firewalls that have high-level security policies. Network administrators usually disable unnecessary communication ports to protect the internal network. Since most data conference software uses more than one special TCP/UDP port to communicate, daily conferences between different health organizations is difficult. Therefore, an "easy to connect" data conferencing system is becoming more and more import in the e-Health field.

A Web-based telehealth system only needs one TCP IP port (80) to transmit data in HTTP protocol. If there is a security required, port 443 can be used to transmit data in HTTPS protocol. Typically, TCP port 80 and 443 are enabled on corporate firewalls and a successful connection can be established.

4.5 Mobile health environment

In a model physical rehabilitation field, health professionals frequently need on-site discussion and consultation. A mobile telehealth system using a tablet PC, Microsoft Windows XP Tablet Edition, and 802.11X wireless technology will be an efficient tool for on-site consulting [10]. Health professionals can record patient's image and video by conference camera and share the information with others over a wireless network.

4.6 Provide health service to the home

The term "Homecare" is becoming more and more popular in the telehealth industry. Homecare's goal is to provide the same health service for patients in home as in hospital. Telecare could be an economical solution that enables physical rehabilitation professionals to diagnosis on-line, thereby helping both patients and health employees. With a web-based telehealth system, an online motion-related diagnosis can be executed using a computer, a camera, a microphone and a high-speed internet link in the patient's home. Combining an enhanced Whiteboard with desktop data conferencing, online diagnosis and consulting becomes more effective and affordable.

Chapter 5. Methods

5.1 Design Criteria

To achieve the objectives, the ideal system should meet the following requirements:

- Compatible with typical workstations, with no configuration or manually installation requirements
- Work across operating systems and across devices
- Work with typical healthcare corporate networks and public internet environments
- User friendly graphical user interface
- Provide motion analysis features for physical rehabilitation
- Use a flexible development tool
- Meet health corporate security requirement

5.2 Development Options

Web applications can meet the requirements of working across platforms on a typical workstation without special configuration and manually installation. Web applications are hosted on a web server, with clients accessing the service via a web browser and a specific Uniform Resource Locator (URL) that is pointed to the web server by a Domain Name System. Online applications do not require manual distribution and installation on each workstation. Two popular cross platform and cross device software applications for online development are Java and Flash. Each has pros and cons for developing online applications.

5.2.1 Java Web Application

Java is used widely to build Internet applications and has robust performance and enterprise capabilities. Java applications are good at query management and security. For this reason, most enterprise online applications are developed in Java.

On the other hand, Java clients need to install the JRE (Java Runtime Environment) to run a Java application. The newest JRE version 1.4.2_14 has a 14.9M installation file size and requires up to 80M disk space for the installation. During runtime, a Java client requires more system resources, including CPU and physical memory at client workstation. This could cause the application to run poorly on workstations with limited physical memory or a lower performance CPU. Another disadvantage of Java applications is that the version of JRE has to meet the SDK (Software Development Kit) version that was used to develop the application. Typically, an application running on the server side only supports a client that is running the same version of JRE as the version of Java SDK. If a client is running a different version of JRE, a pop-up window will notify the user to download the proper JRE version. Problems occur if a user wants to simultaneously connect to multiple Java applications developed with different Java SDK version. In additional, designing a graphical user interface in Java is not easily understood by graphic artists and interface designers.

5.2.2 Flash Web Application

Flash is a popular approach for Web application development. A Flash-based rich-content Web application is hosted by a Web Server and Flash Media Server. Web servers provide HTTP connections and Flash Media Server provides real-time audio, video, and data access via Flash movies.

Flash has evolved from a way to easily create and distribute lightweight animated graphics on the Web to a rich application platform. Users can use the Flash Player to connect to Adobe Media Server for enhanced multimedia streaming capabilities. The client side application, Flash Player, is much lighter than Java JRE. The Flash player works on Microsoft Windows, Macintosh, and Linux operating system, as well as portable device operating systems like Windows Mobile. Flash also provides http tunnelling to successfully connect through firewalls. Secure Sockets Layer (SSL) also has been integrated into the flash transport protocol Real-time Media Portal (RTMP) as RTMPTS (Real-time Media Portal with http tunnelling over SSL).

Flash has a number of advantages for developing light-weight, rich-content Web applications. The following are the main pros from Flash Web application:

5.2.2.1 Light-weight on the client side

The Flash player 8 installation file size is only 1,136 K, which is much smaller than JRE (14.9M). Launching a Flash player only requires 128M of physical memory and an Intel Pentium II 450MHz CPU [56].

The html file that the user downloads to run a Flash application is also lightweight. The media source, including audio, video and application data, is not embedded in the html file. Html files use "embed src" to connect to a swf Flash movie file located on the Web server. The swf file does not include media content data; therefore, Adobe media protocol RTMP is used to connect to a Flash Media server to access multimedia content. All files, including html and swf, are independent of the multimedia content. Usually, the size of a typical Flash Web application's html and swf files are less than 300K. The media content will stream from the Flash Media Server upon demand. Meanwhile, users do not have to wait the for entire media content to download. Once the media buffer is full, the client can start playing the media and the downloading process will run simultaneously in the background.

Thirdly, the client side software Flash player is widespread on the Internet. Adobe reported that Flash Player 8 was available on more than 85% of Internet-accessible workstations in Mature Markets (Canada, the United States, and Europe) as of September 2006. Availability in Emerging Markets also exceeded 83% [7]. In addition, Adobe Flash is the world's most pervasive software platform, reaching over 98% of all connected desktops on the web and distributed with major Web browser vendors including Microsoft, Apple, Netscape, and AOL [12]. Flash offers a consistent media experience across operating systems and browsers.

5.2.2.2 Rich-Content

Flash Web applications can provide rich media content via a Web browser. With Flash Media Server and Flash Player, the following features could be achieved:

- Customized audio/video conferences and chat with shareable components such as text content window, Whiteboards, and data grid

- Video-on-demand applications with customized user interfaces that can include captions and controls with customized skins
- Live broadcasting with customizable user interaction
- Multiplayer games and research simulation tool [54]

5.2.2.3 Easy to establish connection

Flash Media Server uses TCP port 1935 as the default port to stream media between media server and Flash Player client. On the Media Server side, port 1935 is always open to accept client requests. Since port 1935 is not typically used in healthcare, this port is usually disabled on health organizations network firewalls.

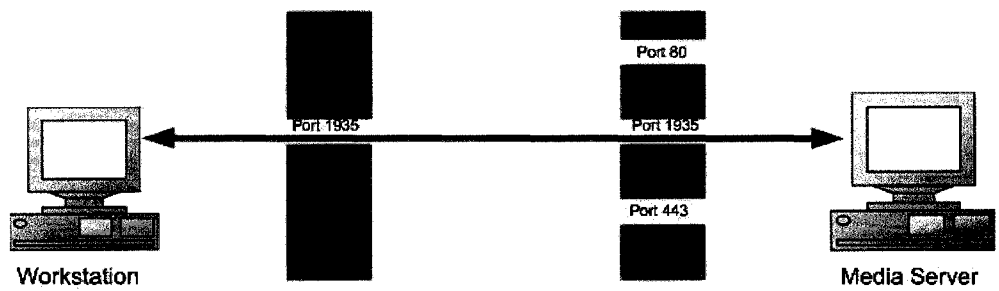


Figure 4: Direct RTMP connection

To overcome the connection issue, an http-tunnel approach was introduced to RTMP. Http-tunnelling is a method that could mask RTMP media packets as http packets to bypass firewalls. If RTMP media packets with an http mask, called RTMPT, are transmitted through TCP port 80, the connection could be established successfully unless an application-layer firewall is also detecting packet payload.

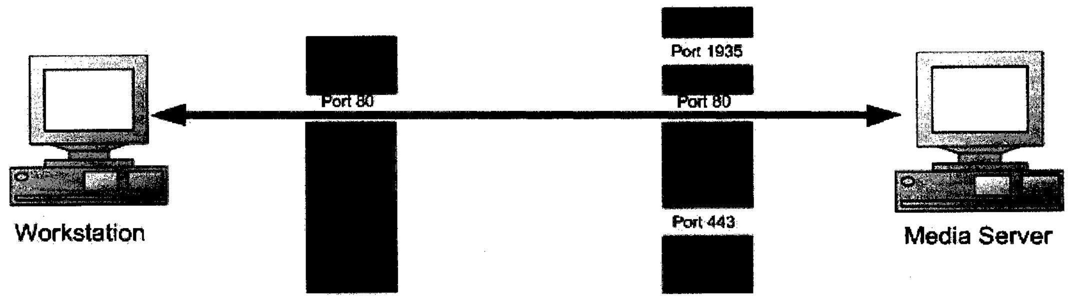


Figure 5: RTMPT connection request with HTTP-Tunnel

5.2.2.4 Security

Since Flash Media Server version 2, the server can deliver encrypted streams to provide media security. Flash players initiate connections with RTMPTS (Real-time Media Protocol Tunnelling over SSL) instead of RTMPT. All packets between clients and the server, including audio, video, and data, are encrypted prior to transport. Once packets are safely delivered to receiver side, Flash player or Media Server decrypts the content in real time. This encrypt/de-encrypt process requires extra CPU resources and may downgrade system performance at both sides.

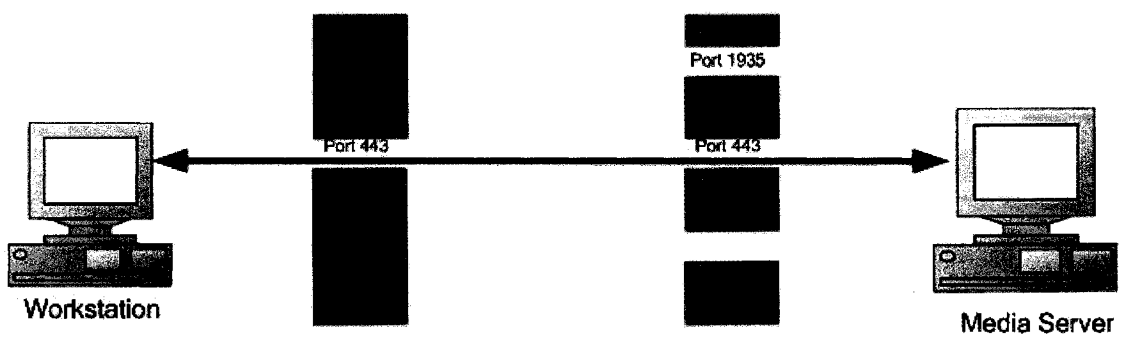


Figure 6: RTMPTS connection request with HTTP-Tunnel over SSL

The protocol is configured with the *connect* method. This method opens a local connection to a remote side HTTP or HTTPS media stream address. The connection type could be configured with RTMP, RTMPT or RTMPTS, accommodating to the network environment. If workstations are behind an application layer firewall with TCP port 1935 disabled, and the firewall also detects HTTP/HTTPS message content, the media steam cannot be established successfully.

5.2.2.5 Cross-platform and cross-device

The flash player could work on Microsoft Windows, Macintosh and Linux operating system, as well as portable device operating system like Windows Mobile. Flash Player 7 for Pocket PC is also compatible with Microsoft Windows Mobile [39]. This Flash version enables users to quickly and easily connect to Flash Media via Windows Mobile 5 of Pocket PC 2003 devices. Users can access available rich-content media and communicate within Pocket Internet Explorer and Flash Player.

5.2.2.6 Powerful development environment

Flash Media Server provides a flexible development environment for creating and delivering innovative, interactive media applications to the largest possible audience. The Flash development environment includes Flash Media Server and Flash 8 Professional. Flash 8 Professional provides a number of function/components for developers and a class-based programming language ActionScript 2.0 that could help developers maximize creativity.

ActionScript is a scripting language based on ECMAScript. ECMAScript is a scripting programming language standardized by Ecma International in the ECMA-262 specification. The language is widely used on the web. As JavaScript, ActionScript is one of dialects of ECMAScripts. As claimed by Adobe, ActionScript 2.0 was implemented with the preliminary edition 3 of ECMA-262 [58-59].

ActionScript is a development language for creating and delivering a broad range of media experiences including Video on Demand, live web-event broadcasts, and MP3 streaming as well as rich media communication applications like video blogging, video messaging, and multimedia Chat environments [41].

ActionScript 2.0 is an Object Oriented Programming (OOP) language which is based on the concept of methods or modular objects. Developers do not need to know what is inside a method. Applications can be built directly use the existing methods. Each method is a self-contained unit that interacts with others. ActionScript programs are made of different self-contained modules, each with a specific job. Developers write classes to build modules with methods defined by Adobe.

With JavaScript, the ActionScript programming language is sufficient for most Flash application development tasks. A content-rich web page created in Flash can display on Microsoft IE (windows version & Mac version), Firefox (Windows version & Mac version), Netscape (Windows version & Mac version), AOL (Windows version & Mac version), and Opera [6]. The Flash Player can be automatically installed on PC and Macintosh systems and all commercial web browsers. The MAT-F software was developed using ActionScript 2.0

within Macromedia Flash 8 and was exported in Flash 8 format (.FLV and .SWF) that can be displayed by Macromedia Flash Player.

In general, using Flash Media Server, Flash Player, and Flash 8 development tools, a web-based motion analysis application could work across platforms, across devices, provide a friendly user interface, and provide advanced motion analysis functions. Flash–based motion analysis tools could also work with typically system and network environments without special network settings and manual software installation. Meanwhile, with protocol RTMPTS, the security requirement could also be met.

5.3 Flash Web Application Platform

The Flash Web Application platform consists of Flash Media Server, Flash Player and a Web Server.

5.3.1 Flash Media Server

Flash Media Server is a server-side application running on a host machine enabling real-time, multi-way communications applications developed with Adobe Flash [54]. The RTMP connection between Flash Media Server and Flash movie running at client side is persistent. This differs from the HTTP request/response model used by browsers to communicate with Web servers. A client side Flash movie is needed to establish the persistent connection via function *nsconnect()* when the movie is launched. Once the server accepts a client connection, the RTMP connection can be used to exchange all kinds of media data including audio, video, and data until either the client or server disconnects [54].

Due to the advantages of streaming multi-media content in real-time with low bandwidth requirements, Flash Media Server is becoming more popular for developing online media applications, including:

- Customized audio/video conferences and chat with shareable components such as text content window, Whiteboards and data grid
- Video-on-demand applications with customized user interfaces that can include captions and controls with customized skins
- Live broadcasting with customizable user interaction
- Multiplayer games and research simulation tool [54]

5.3.2 Shared Objects

The SharedObject is an ActionScript class that is used to access and modify data on the server or client side to maintain persistent connections between server and clients. Real-time data sharing is achieved between multiple Flash movies (*swf* files) running on clients using persistent objects that exist on the Flash Media Server. Objects that bridge data between multiple clients are called shared objects [55].

There are two different types of shared object: *Local Shared Object* and *Remote Shared Object*. Local shared objects are similar to browser cookies that work without server involvement. Remote shared objects are similar to real-time data transmitters that require Flash Media Server involvement [51].

Three common uses of shared objects are maintaining local persistence, storing and sharing data on a server, and sharing data in real time. The pre-requirement for Local shared objects is that the machine Flash Player is running on must allow Flash Player to write to the local hard disk. Flash Player users have privacy controls that can prevent third-party domains (domains other than the domain in the current browser address bar) from reading or writing local shared objects [51].

If *swf* files are stored and run on the local computer, not from a remote server, the file can always write third-party shared objects to disk, even if the user does not allow writing of shared objects to disk by third-party domains. Users can setup the third-party content on the Global Storage Setting panel in Flash Player [55].

5.3.3 Communication Protocols

5.3.3.1 Traditional Media Packets Delivery Protocols

Most media packets are delivered on the Internet with UDP (user datagram protocol). Since the Internet was a congested network when media servers appeared one decade ago, UDP was introduced for broadcasting messages over a network. Since UDP is a packet based, non-order, connectionless protocol, traditional media servers usually chose UDP as the packet delivery protocol.

Pros of UDP for Media Packet Delivery

- **Packet Based**: Each packet is distinct and has a length. No dependence between packets. The previous and next packets do not impact and are not impacted by the current packet.
- **Connectionless**: Same port can be used multiple times at sender side.

- **Non-guaranteed**: If a packet is sent out, the next packet will not be held while waiting for acknowledgment or re-send of the previous packet.

Cons of UDP on Media Packets Delivery

- **Non-guaranteed:** There is no acknowledgment for packet transmitting.
- **Unordered:** The packet order at the receiver side is different than the order at the sender side.
- **Non-guaranteed Unique:** The same packet could be received more than one time at the receiver side.

Media packet delivery emphasizes synchronization more than packet loss. Unlike TCP, UDP does not provide error recovery services, offering instead a direct way to send and receive datagram over an IP network. On congestion networks, a "non-guaranteed" transmitting method could provide less latency and avoid sending acknowledge and re-transmitting packets that could impact network performance.

Two methods can be used to distribute a stream with UDP: unicast and multicast. Unicast is the term used to describe media packets sent from one single point to another single point. Most Internet packets are transmitted in unicast mode (i.e., HTTP, FTP, TELNET, SMTP). Multicast means sending media packets from one point to multi-points simultaneously [50]. The advantage of multicasting is to reduce server CPU load and bandwidth requirements when a server sends the same streams to a group of clients. Unfortunately, most Internet service providers (ISPs) disable multicast on their networks due to security concerns such as the Slammer worm [49]. Consequently, media servers provide a fallback to unicast streaming over UDP. Unicasting means the server must duplicate the stream data and send a separate stream to every client. That can increase server load and network bandwidth requirement dramatically. If the client does not support the unicast stream, media servers typically fall back to sending duplicate stream data to each client over TCP. With "guaranteed mode", the TCP approach can cause more increases on server load and bandwidth requirements. As a result, the media quality is downgraded [50].

5.3.3.2 Flash Media Packets Delivery Protocols

Unlike traditional media servers, Flash Media Server uses TCP as the foundation protocol to transmit audio, video and ActionScript data transfer between clients and server [51]. Packets transmitting over TCP could be simpler than over UDP, but TCP requires a separate server/client TCP connection between each client and server since TCP is a point-to-point protocol. Consequently, Media Server cannot broadcast or multicast packets from the server to multiple clients at the network level. If a live audio stream must be sent simultaneously to multiple clients, the server must send duplicate copies of the audio data to each client over discrete connections.

RTMP (Real-Time Messaging Protocol) is a protocol designed by Macromedia which is built over TCP to transmit packet between Flash Player and the Media Server [61]. RTMP is specially designed to overcome the disadvantage of media transmitting in TCP. RTMP supports dynamic transmission of multiple streams that can contain audio, video, and ActionScript data between server and clients. The protocol priories varied media packets during transmitting. Those packets assigned with lower priority could be dropped first due to inadequate network bandwidth. Packet priority is assigned based on the impact on real-time conversation. Signalling data (ActionScript data) own the highest priority, and then audio and video packets. Data packets, like Whiteboard display and typing, are assigned the lowest priority [62]. Based on the priority system, RTMP can adjust the amount of data, audio and video being transmitted dynamically by dropping audio messages, video frames and media data packets in response to limited network bandwidth. Since RTMP is not a packet based transmitting mode, the protocol is not as effective as UDP-based protocols for delivering streaming media on congested networks. On the other hand, RTMP is more reliable than UDP delivery since RTMP does not drop signalling data (ActionScripts data), thereby preventing catastrophic effects occurring on media applications. On medium quality networks, like current public Internet, RTMP delivers media in a reliable and effective manner [62].

Real-time media packets containing audio and video payloads are buffered separately on the media server. If the number of audio packets in the audio buffer reaches to a certain threshold, the whole buffer is cleaned up to allow space for newly arrived packets. Video packets are managed in a similar manner except that the packets in the buffer are dropped when a new keyframe arrives. Having a clean video buffer once a keyframe arrives ensures that the client never receives partial frame updates for the wrong keyframe. If the client did send partial frames, "the video image would be made up of a mosaic of 8x8 pixel blocks from two different frames" [51].

Video content is coded with a native format designed by Macromedia, Flash Video format (FLV), that is a variant of H.263. Data is encoded using Macromedia's Action Message Format (AMF) [52]. AMF is a binary format designed for the ActionScript object model with the Simple Object Access Protocol (SOAP) format. This format provides an efficient way to serialize/deserialize data so that ActionScript data can be transferred between client and server without encode or decode the data at client side. Server function "NetServices" are requested to serialize data into AMF format. The media server deserializes the incoming AMF messages and serializes the result back to AMF, then sends messages back to the client side. The server-generated AMF message format is identical to the client-generated packet [40].

Standard RTMP does not have security enhanced features. The protocol transmits unencrypted data between client and server, including authentication information such as name and password. To overcome security concerns, a SSL-enabled web server could help RTMP to encrypt data through HTTPS. Meanwhile, an enhanced variant of RTMP, Real-Time Messaging Protocol over SSL (RTMPS), was designed to transmit date in an encrypted manner.

RTMPS is an enhanced protocol adhering RTMP with SSL standards to achieve secure network connections [53]. The protocol offers connectivity through a TCP socket on a secure port. Data passed over a secure connection is encrypted to avoid eavesdropping by unauthorized third parties. Since establishing secure connections requires extra CPU resources, a more powerful server is required prior to enabling RTMPS on media server in order to avoid system performance downgrade [42].

In summary, Flash Media Server provides media streams services with RTMP protocol in a highly reliability and an effective manner. Media Server can be as effective as UDP since the server can dynamically adjust the number of media packets being transmitted in response to network bandwidth and congestion. Meanwhile, Media Server also provides high reliability by prioritizing different data types. The most important signalling data, ActionScripts data that controls of the media application, will never be dropped. Media packets are buffered separately to optimize the quality of real-time multi-media communication. On congested networks, RTMP is not as effective as UDP-based protocols for delivering streaming media, but on medium quality networks RTCP delivers acceptable performance.

5.4 Flash Web Application Platform for Motion Analysis

The software developed in this thesis, Motion Analysis Tools Flash Version (MAT-F), could promote greater communication and understanding in several ways. Participants can run a motion analysis session on the Whiteboard. Since all conference participants can control the video and draw on the window, all analysis data could be transmitted to all client sites in real-time. Meanwhile, participants can see remote side nonverbal signals via the video camera, signals that they could miss in a traditional telephone conference call. Video analysis capability brings functionality beyond typical telerehabilitation video conferences by integrating digital video control, real-time analysis, and video annotation to allow people to demonstrate a rehabilitation video clip as you explain analysis results to others in the conference.

5.4.1 Development Concept

MAT-F shares data in real time using the SharedObject.getRemote() method to create a shared object on a Flash Media Server. The application uses a remote shared object, named MAT, to share information in real time between Flash movies running on different clients. If a movie updates the value of a data member within the shared object MAT, the same data member is updated in every other movie connected to MAT. When any MAT data member changes, each client is notified of the change automatically on a real-time basis.

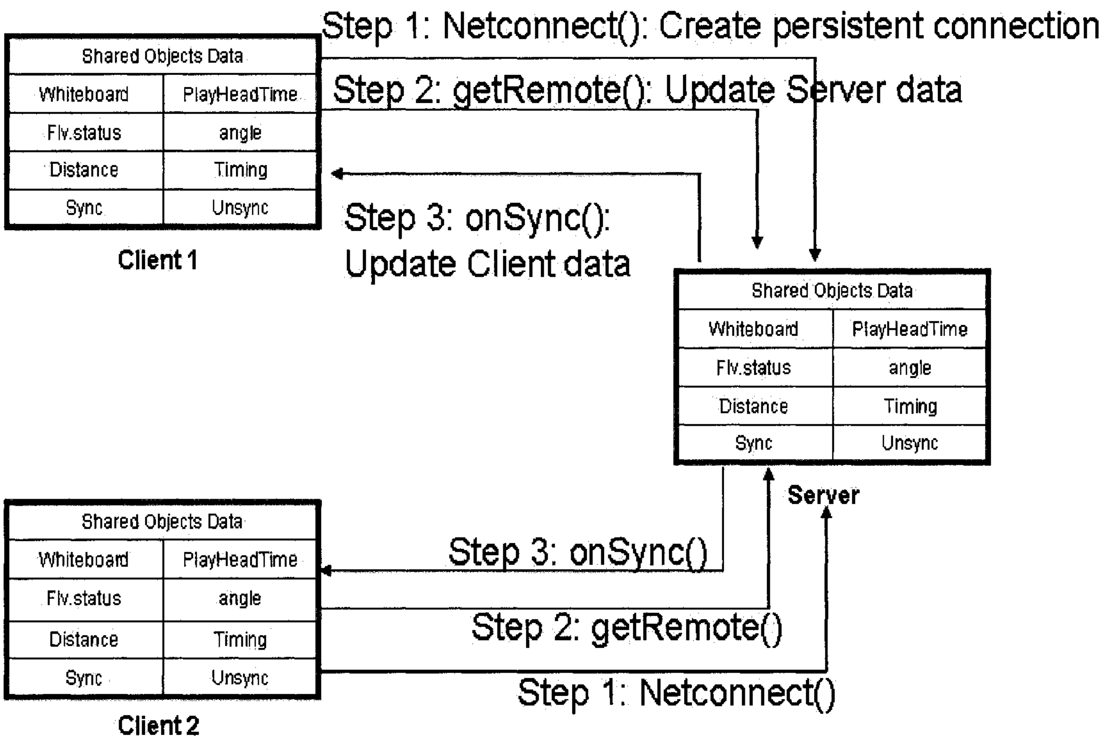


Figure 7: Shared Object Chart flow

The shared object MAT within the MAT-F data conferencing application was used to:

- Notify all movies connected to MAT-F
- Update the position of Whiteboard elements
- Hold the position, shape, and color information of each graphical element in the Whiteboard
- Update the Flash Video file current play header time and video status
- Update the biomechanics motion analysis data display

- Hold the data for each form element in MAT-F that is used to define the current state of the MAT-F application.

5.4.2 Programming

The MAT-F application was developed with ActionScript, JavaScript and ASP.NET languages. JavaScript is a compact, object-based scripting language for developing client and server Internet applications. JavaScript can be run on all commercial web browsers. In this project, JavaScript was used to access local resources; for example, open another browser, explore local files, open a new web browser from the Flash User interface. Active Server Pages (ASP) language was used to develop the file upload function. ASP is a server side script language. A user's browsers can call ASP code from the MAT-F ASP-enable web server.

5.4.2.1 Network Connection

Network Connection handles the server connection and all communication object connections. Network Connection also provides a user log-in interface. The method "SimpleConnect" was used.

Bandwidth Setting used the SetBandwidth component to automatically adjust the quality of microphones and cameras that are published to match the available upload bandwidth. The "Setbandwidth" method was used.

Connection Status provides visual feedback on the client connection state. A light icon on the user interface turns green when the client is connected to a conference, red when disconnected, and yellow if the connection latency is too high. The light doubles as a button that toggles a display box, providing detailed information about the connection (data latency rate, instantaneous upload and download rates). The method "ConnectionLight" method was used.

5.4.2.2 Audio and Video Conference

Audio and Video Conference enables multiple users to interact with each other using audio and video. The Microphone could be muted/un-muted. The quality of audio and video depends on the connection bandwidth and the upload bandwidth setting on the remote side. The audio/video quality could degrade if the remote side connection setting is mismatched to the real connection. The method "VideoConference" was used.

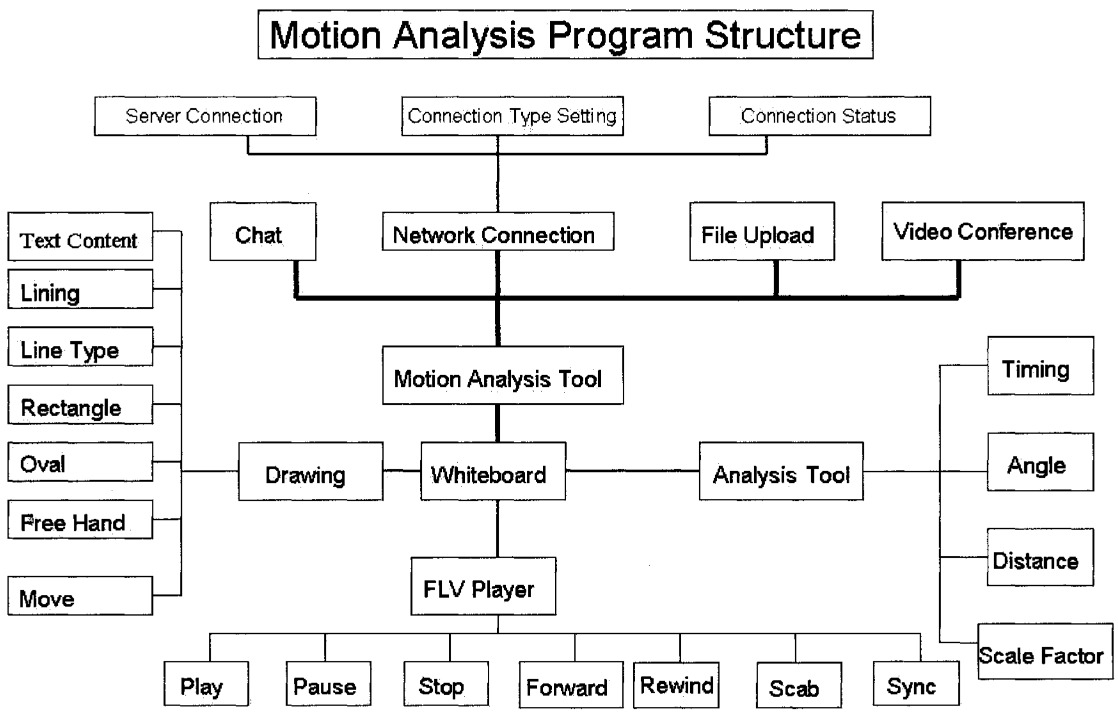


Figure 8: Motion Analysis Program Structure

5.4.2.3 Chat

Chat provides a text messaging area for conference participants. Users can pick a text colour to aid communication. All participants are listed on a text area. Chat could help users exchange detailed information like numbers or web site addresses. In scenarios with very limited bandwidth, Chat could be a backup for audio /video communication. The record of Chat data is saved on the Media Server. The method "Chat" was used.

5.4.2.4 FLV File Upload

FLV File Upload provides an interface for users to upload a Flash Video file from the local machine to the Media Server in order to share the file with other conference participants. The component was developed with ASP.NET. On the server side, the ASP.NET Framework should be installed. The method "HtmlFileInput" was used.

5.4.2.5 Whiteboard

Whiteboard provides a shared drawing area with create/edit text, boxes, and lines in a real-time shared environment. The MAT-F Whiteboard also provides a Flash Video player that is controlled by the remote shared object MAT (on Media Server). If the operation mode is set to "synchronization", the FLV player status is set by a server side shared object. A biomechanical motion analysis function was also build in the Whiteboard component for distance measurement, timing, angle calculation and distance scale factor functions.

- **Drawing** : Text Content Input, Lining, Line Type Option, Rectangle, Oval and Cycle, Free Hand Draw, Move, Erase. Users could move or erase any sharp on draw area. The Free Hand Draw function provides a flexible tool to draw any shape.

- **FLV Player**: Play, Pause, Stop, and Forward Frame by Frame, Rewind Frame by Frame, Scan, and Synchronize/Un-Synchronize. Flash Video Player could stream a FLV file existing on server side. Users do not have to wait for the whole file download before the local machine plays the video. The FLV player is controlled by local control buttons or server side remote shared object. If the operation mode is set to "synchronization", the player is controlled by local button and a shared object. If the operation mode is set to "non-synchronization", the player is fully controlled by local buttons. The value of "mat.status", a remote shared object, is used to control the status of client side FLV play.

MAT-F Program Flowchart

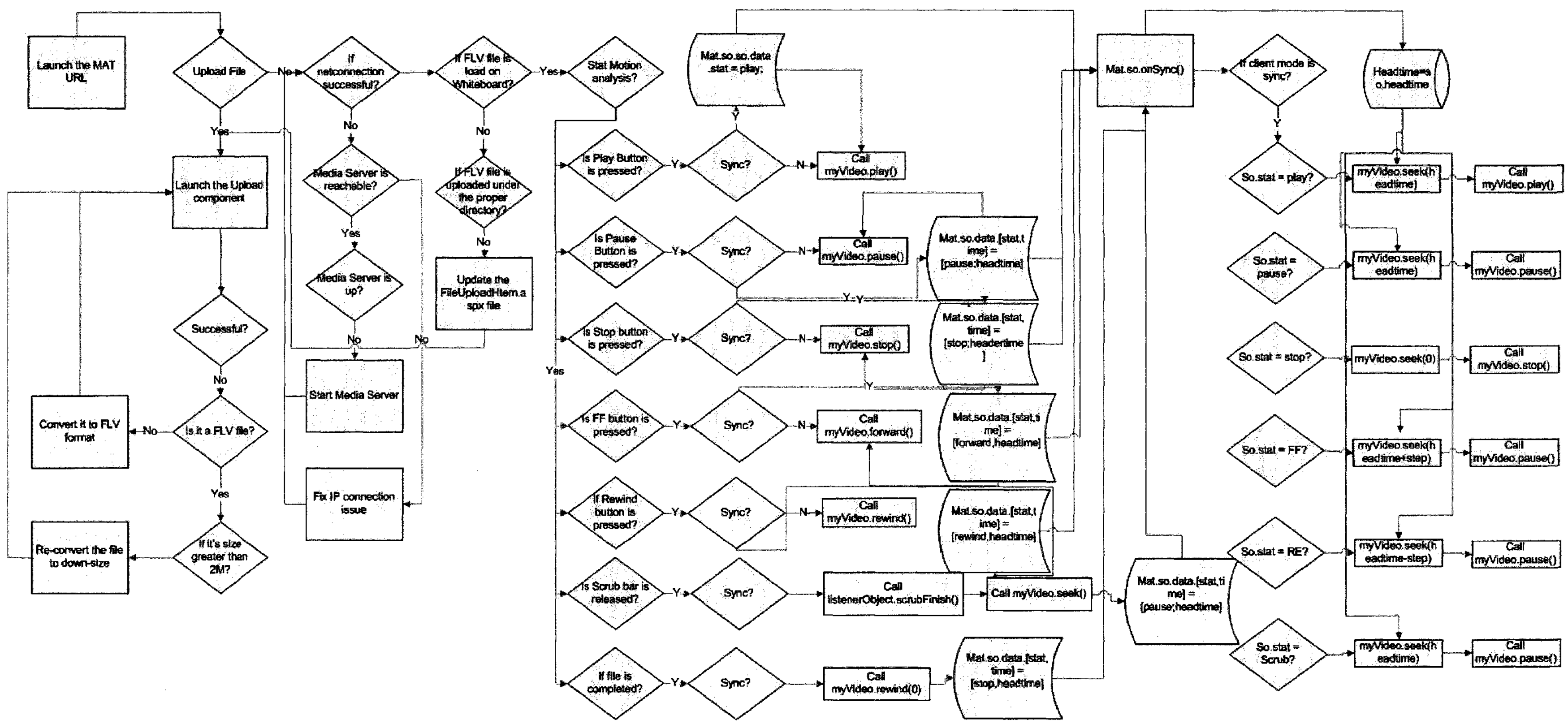
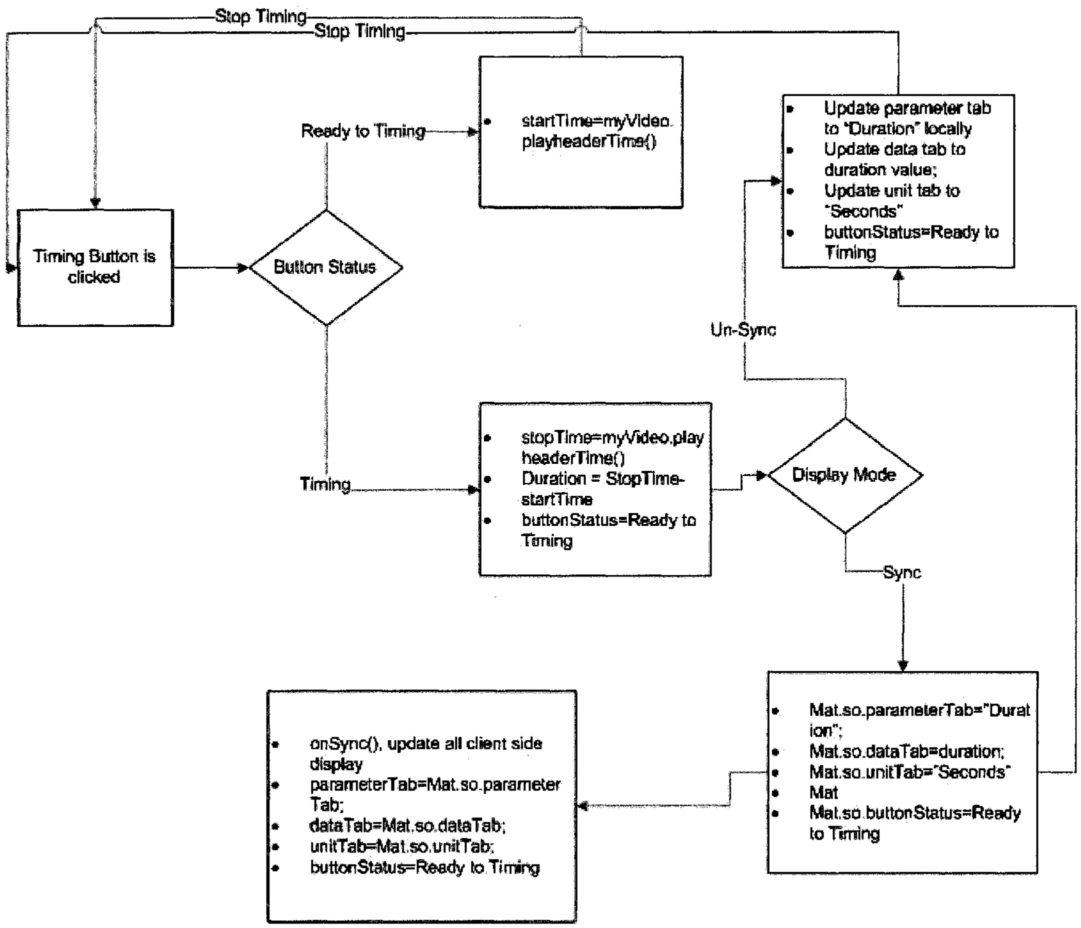


Figure 9 : MAT-F program flowchart

5.4.2.6 Motion Analysis Tools

Motion Analysis Tools provides timing (duration), angle calculation, and distance measurement based on a scale factor setting.

1. **Timing**: The timing function was designed with a method "playheadtime". "playheadtime" gets the FLV file's current timestamp. Once the timing button is clicked, the application checks the button's status. If the status is "timed", the current play header time is recorded by a temporary variable "startTime", and the button status changes from "timed" to "timing". If the status is "timing" when the button is clicked, the current play header time is recorded in another temporary variable "stopTime", and the button status is reset to "timed". The duration value is calculated by stopTime minus startTime. If the display mode is sync, a getRemote() call is sent to the server to update the server side data. If the display mode is "un-Sync", the application will only update the display locally.



Timing Function Flowchart

Figure 10: Timing Function Flowchart

2. **Scale Factor**: Scale factor is a feature to convert the on screen distance into a real physical distance. The length of the scale factor bar on screen can be adjusted by dragging the mouse. An event listener monitors mouse actions. Once the mouse is released over the scale factor bar, the length between the coordinate of release point and the coordinate of bar's start point is calculated using Equation 1 (x = x-axis screen coordinate, y=y-axis screen coordinate, z=length).

$$z = \mathrm{sqrt}\left((x2-x1)^2 + (y2-y1)^2\right) \quad \textbf{(Equation 1)}$$

The units are pixels. Next, the distance setting value (SFD) is obtained from the text field "scale distance setting" and the scale factor unit value (SFU) from the test field "scale factor unit setting". The scale factor (SF) equals *SF=z/SFD (pixel/SFU)* and the updated scale factor value is displayed on the information bar as "1 SFU: SF pixel".

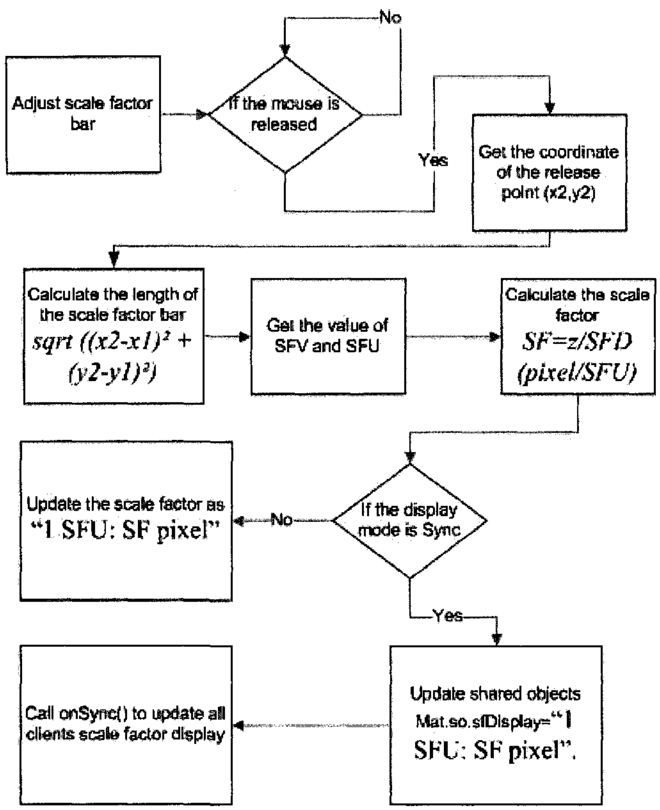


Figure 11: Scale Factor Flowchart

3. **Distance:** The distance measurement function calculates the distance between two points on screen and then divides the results by the scale factor. The screen coordinate was retrieved with a method "_xmouse" and "_ymouse". The distance between two points is calculated using Equation 1

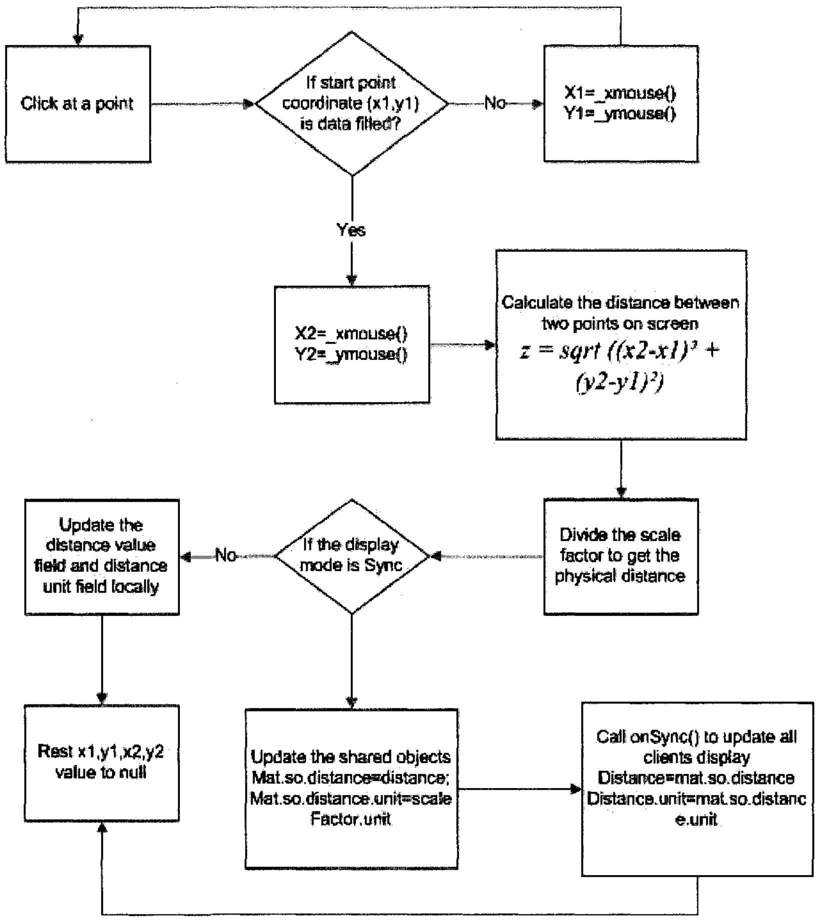


Figure 12: Distance Function Flowchart

4. **Angle**: Angle is calculated with the coordinates of three points (point A, B and C). This function captures the coordinates of three points, saves the data to temporary valuables, then calculates the angle using Equation 2. The math.ceil() method is used to round the real number up to the closest integer.

θ= 180×acos ((AB+BC-CA)/(2*sqrt(AB)*sqrt(BC)))/ ∏ **(Equation 2)**

If the display mode is set to "Sync", the shared objects data will be updated. Otherwise, the value of angle will be displayed only on local machine.

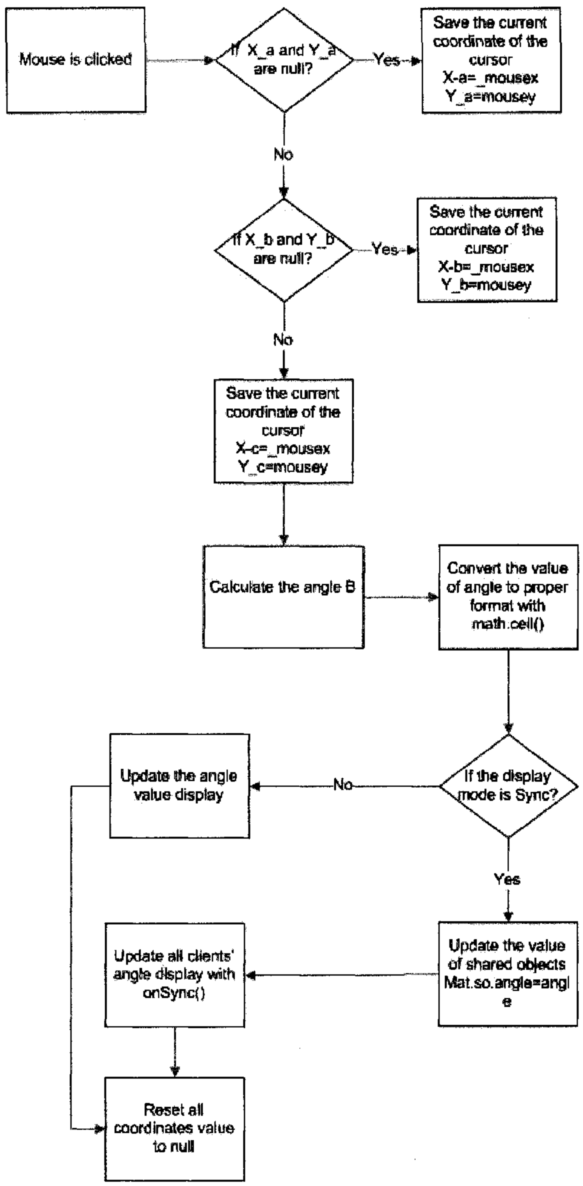


Figure 13: Angle Calculation Function Flowchart

5.4.3 Network Environment

MAT-F uses two servers: Web server and Media Server. Both servers were located on the same subnet. The Web server was a Windows 2003 Server with IIS6.0 and the Media Server was Flash Media Server 2. HTTP Tunnelling on Port 80 was the communication port for both servers. Microsoft Windows 2000/XP were supported as client side workstation operating systems.

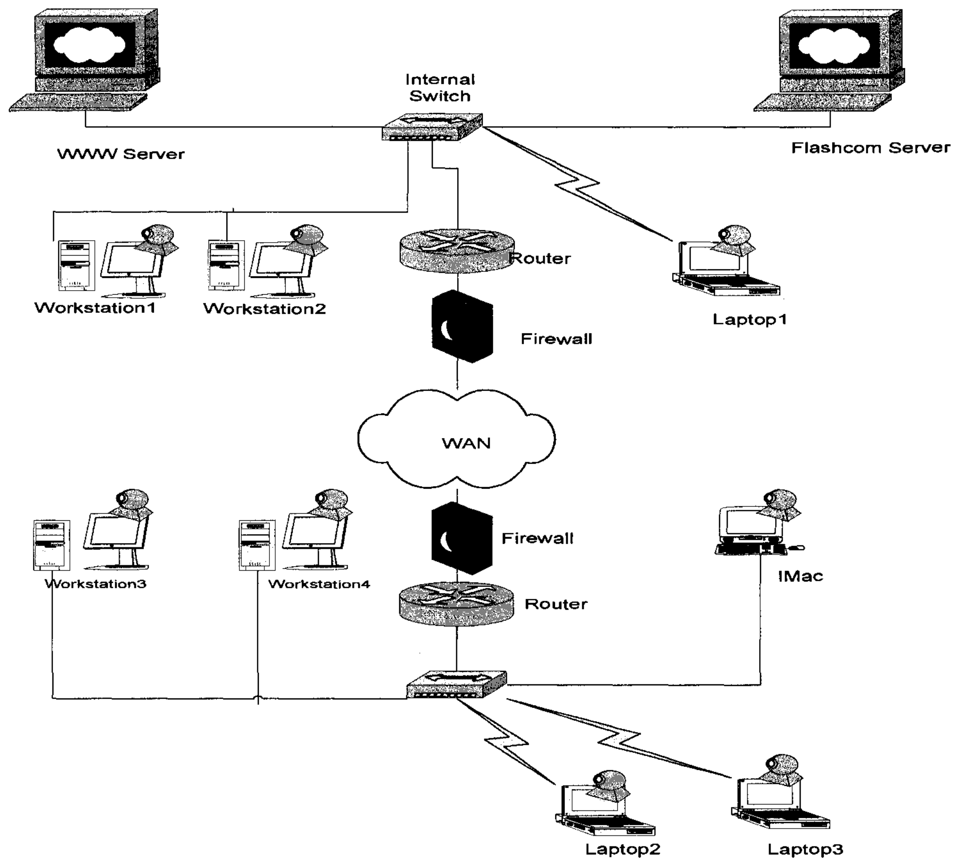


Figure 14: MAT-F Network

5.5 Evaluation

Telerehabilitation system evaluation was divided into two categories. "The first derives from the biomedical and clinical field and it encompasses issues of clinical effectiveness, efficacy, and safety. The second type or set of research questions derive from health services research. These questions concerns Access, Quality and Cost of Care, for which the measures are utilization, referral, convenience, opportunity cost." [43-44].

Motion Analysis Tool Flash Version (MAT-F) evaluation was conducted in the health services research mode and focussed on the measurement of Access and Quality of Service. Cost of Care is out of scope of this evaluation activity.

5.5.1 Accessibility

Accessibility refers to the method by which patients or health professionals are able to obtain and maintain needed medical services [45]. The access evaluation of MAT-F was conducted with the following categories:

- **Web page load speed**: Evaluate the time to load the MAT-F page on different workstations that have different Internet connections. Microsoft Internet Explore (IE) version 6 and Firefox version 2.0 were evaluated from Web browser perspective.
- **File upload speed**: Measure the FLV file uploading with ASP.net service running on Windows 2003 server. The threshold of file size was set to 2MB.
- **Easy to establish media stream connection**: Validate three kinds of media stream connections including RTMP, RTMPT, RTMPTS from different network environment. The CPU usage date of Media Server and workstations were also

collected to evaluate the impact from HTTP Tunnelling and SSL encryption/decryption.

5.5.2 Quality

Quality of care is defined as "health care that is compatible with current medical knowledge and that increases the probability that an individual will achieve a desired health outcome." [46]. The strategy of MAT-F quality evaluation is to evaluate the key functions with traffic running on Flash Media Server.

The Flash Media Server 2 Developer Edition was used during the development and evaluation phases. The maximum capacity of the Developer Edition is 250 kilobits of peak bandwidth per second (kbps) or 10 simultaneous connections, whichever limit is reached first [47].

The number of simultaneous connections was set to 10. To reach the ceiling of the number of connection, five workstations accessed the Media Server via different connection types (ASDL and LAN), two connections per each workstation.

Video conference and MAT-F were run simultaneously to generate media traffic to reach 250 kilobits of peak bandwidth per second (kbps). The measurement threshold for Flash Media Server performance was a Message drop rate less than 2% when the Flash Media server CPU usage of the server is more than 50%; Active connections is 10; Bandwidth usage is more than 240 kilobits.

MAT-F quality evaluation focussed on the quality of the shared media, including Flash Video and Whiteboard. The following key quality indicators were evaluated during activity:

- **Synchronization:** Video synchronization between clients was measured and evaluated. Synchronization is the ability of the environment to maintain the same video frame on all clients connected to the Media Server. One test variable, video timestamp, was created to record flash video time at the client whenever an operation event occurred. The difference between flash video times from individual workstations was the synchronization error. Synchronization error less than 0.1 seconds was considered acceptable. The error accumulation test between five workstations was also conducted during the evaluation. The mean synchronization errors between five clients were calculated.

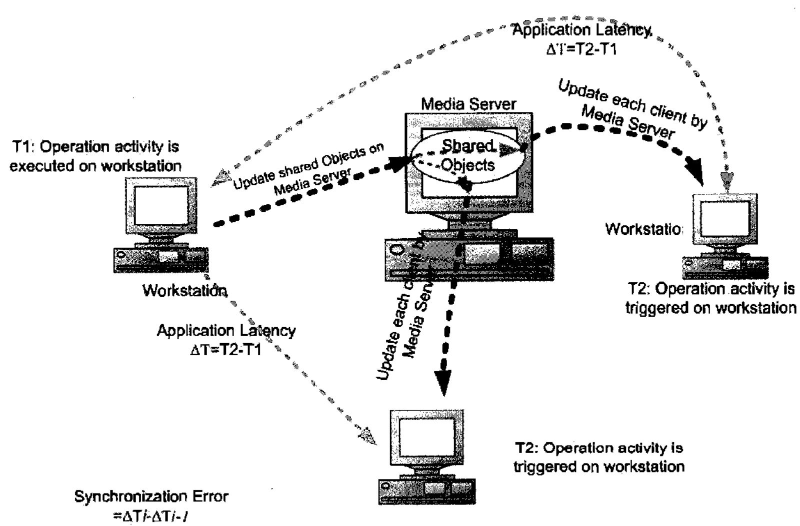


Figure 15: Synchronization Error Measurement

To evaluate the impact from network latency on synchronization error, a test was executed from two PCs (PC1 and PC2) that have individual physical network connections. Third-party software, Ethereal, was used to capture AMF format TCP packets at layer 2. Both PC local times were synchronized by a network timer server (NTS). The synchronization frequency was set to 1 minute to keep the two PC local times synchronized at the millisecond level. The following steps calculated video control network latency:

- Start Ethereal to capture TCP packets on both PCs. The capture filter was set to "tcp.port==1935" to only capture AMF packets between Web Server and two PCs;
- Press "Play"on PC1, let the video play for at least 1 second, and then" Pause" on one PC;
- Stop Ethereal trace on both PCs;
- Find the AMF packets that "*mat.status*" showed as "play" and "pause";
- Record the time of "play" and "pause" on both PCs;

The network latency of "Play" was the time difference between "play" commands on both PCs (i.e., "*play_latency*"). Similarly, the "Pause" network latency was the time difference between "pause" on both PCs (i.e., "*pause_latency*"). The difference between "*play_latency*" and "*pause_latency*", called "*video_control_latency*" were calculated, and the current video frame time, termed "Timestamp" in this thesis, on both PCs were recorded for evaluation. Fifteen trials were executed.

- **Video Control Functions**: Video Control Functions Play, Pause, Stop, Forward by Frame and Rewind by Frame functionality were evaluated. The Play button can trigger the flash video to be played. Pause button can trigger the flash video to be paused. Stop button can trigger the flash video file to be stopped and auto-rewind to the header. Forward button can trigger the flash video file forward by one frame. Rewind button can trigger the flash video file to rewind by one frame. The FLV file was auto-rewound to time 0 once the play head time reached the end of the file.

- **Whiteboard functions**: Enhanced Whiteboard functions Freehand draw, Move, and Erase were evaluated. The Freehand draw allows users to draw any shape with the mouse. The Move feature moves individual shapes on stage to different locations. The Erase function erases all shapes on stage, except the video.
- **Motion Analysis Function**: Motion analysis function evaluation was done by comparing motion analysis angles and distances with a known angle value and distance. Validate the angle measurement result (variance less than 0.1 degree); validate the distance measurement result (variance less than 1cm with scale factor); validate the timing (less than 0.1 seconds);
- **Display mode**: Verify that video and biomechanical analysis display is synchronized on each client when the mode is set to "synchronisation" and that the video and biomechanical analysis display is independent between browsers when the display mode is set to "Asynchronous".

Chapter 6. Results

The MAT-F system was successfully developed. As shown in figure 16, all required application components were arranged as a web-application in a user-friendly orientation.

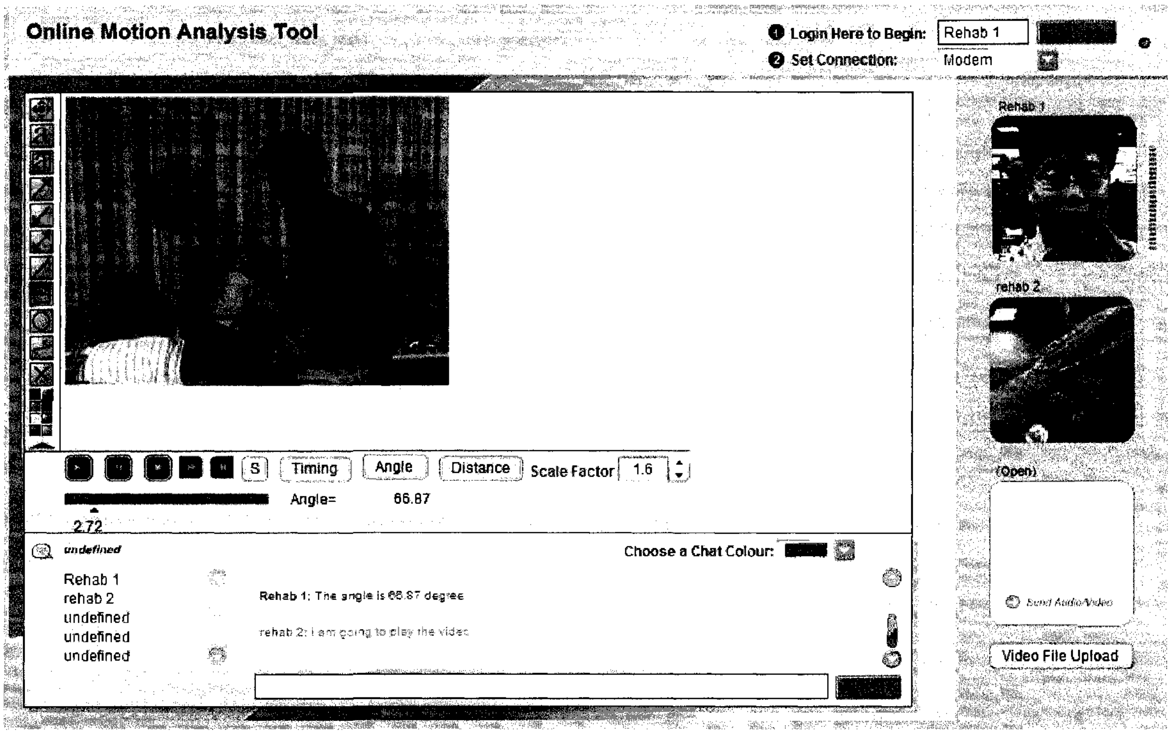


Figure 16: Motion Analysis Tools snapshot

6.1 Accessibility Test Result

6.1.1 Web page load speed

Web page load speed was evaluated with a third-party tool SelfSEO [57]. The testing was conducted at different times via different network connections. Results are shown in table 3.

Table 3: Web Page Load Speed Test Result

#	Page Size (KB)	Load time (s)	Average Speed per MB (s)	Type of Connection
1	34.15	0.61	0.02	ADSL
2	34.15	0.56	0.02	10M LAN
3	34.15	0.60	0.02	ADSL
4	34.15	0.60	0.02	10M LAN
5	34.15	0.56	0.02	10M LAN

6.1.2 File upload speed

The file upload function (ASP.NET program HtmlInputFile.aspx) was evaluated with different network connections and Flash Video files (table 4).

Table 4: File Upload Speed Test Result

#	File Size (KB)	Load time (s)	Average Speed (KB/s)	Connection Type
1	1189	28.1	42.3	10M LAN
2	3600	82.0	43.9	10M LAN
3	1189	30.3	39.24	ADSL
4	3600	94.5	38.09	ADSL

6.1.3 Connection

For the Public Internet test, a work station was connected to the public Internet via Bell Canada Sympatico Hi-Speed Internet connection with an average download bandwidth of 2.0 Mb and average upload bandwidth of 600kbps. Ports 80, 443 and 1935 were enabled at the home router. As shown in figure 16, the Flash Player established a connection successfully via RTMP with port 80 or 1935, RTMPS with port 443, RTMPT with port 80, RTMPTS with port 443 from home office network.

Address A	Port A	Address B	Port B	Packets	Bytes	Packets A->B	Bytes A->B	Packets A<-B	Bytes A<-B
192.168.1.101	2503	216.104.212.54	1935	1	54	1	54	0	0
192.168.1.101	2504	216.104.212.40	https	1	54	1	54	0	0
192.168.1.101	2507	216.104.212.54	http	1	54	1	54	0	0
192.168.1.101	2502	216.104.212.40	1935	1	54	1	54	0	0
192.168.1.101	3013	216.104.212.40	http	18	7699	9	3921	9	3778
192.168.1.101	3014	216.104.212.40	https	18	7700	9	3922	9	3778
192.168.1.101	3015	216.104.212.54	1935	18	7701	9	3923	9	3778
192.168.1.101	3016	216.104.212.54	1935	18	7696	9	3918	9	3778
192.168.1.101	3011	216.104.212.54	http	38	8225	22	5403	16	2822
192.168.1.101	3012	216.104.212.54	http	39	8273	22	5397	17	2876
192.168.1.101	2505	216.104.212.54	http	53	11895	28	5598	25	6297
192.168.1.101	2506	216.104.212.54	http	54	12088	28	5593	26	6495

Figure 17: Public Internet Connection Test Result

For the University of Ottawa network test, the work station was connected to the campus network via 100 Mbps LAN connections. As shown in figure 17, the Flash Player established a connection successfully via RTMP with port 80 or 1935, RTMPS with port 443, RTMPT with port 80, RTMPTS with port 443 from the network of the University of Ottawa.

Address A	Port A	Address B	Port B	Packets	Bytes	Packets A->B	Bytes A->B	Packets A<-B	Bytes A<-B
192.168.1.101	2503	216.104.212.54	1935	1	54	1	54	0	0
192.168.1.101	2504	216.104.212.40	https	1	54	1	54	0	0
192.168.1.101	2507	216.104.212.54	http	1	54	1	54	0	0
192.168.1.101	2502	216.104.212.40	1935	1	54	1	54	0	0
192.168.1.101	3013	216.104.212.40	http	18	7699	9	3921	9	3778
192.168.1.101	3014	216.104.212.40	https	18	7700	9	3922	9	3778
192.168.1.101	3015	216.104.212.54	1935	18	7701	9	3923	9	3778
192.168.1.101	3016	216.104.212.54	1935	18	7696	9	3918	9	3778
192.168.1.101	3011	216.104.212.54	http	38	8225	22	5403	16	2822
192.168.1.101	3012	216.104.212.54	http	39	8273	22	5397	17	2876
192.168.1.101	2505	216.104.212.54	http	53	11895	28	5598	25	6297
192.168.1.101	2506	216.104.212.54	http	54	12088	28	5593	26	6495

Figure 18: U of Ottawa Campus Network Connection Test Result

6.2 Quality Test Result

6.2.1 Synchronization

Flash video sharing synchronization testing was conducted from two aspects: network latency impact evaluation and client/server processing speed evaluation. The client/server processing speed evaluation was tested with two modes: all operations done at same workstation and operations done at different workstations. To understand how network latency impacts synchronization error, a real-time network latency measurement test had been conducted.

The network latency test evaluated the impact on synchronization error. AMF packets were captured on two PCs. Time of each packet sent/received in millisecond were recorded. "*play_latency*", "*pause_latency*", and "*video_control_latency*" were calculated. The test was repeated 15 times.

Table 5: Network Latency and Synchronization Error

Trial #	Synchronization Error (ms)	Video Control Latency (ms)	"Play" Button Latency (ms)	"Pause" Button Latency (ms)
1	0	-5	20	15
2	40	7	28	35
3	0	-2	14	12
4	0	0	19	19
5	-40	-70	86	16
6	40	1	2	3
7	0	0	12	12
8	0	0	17	17
9	40	0	16	16
10	0	0	8	8
11	-40	-33	43	10
12	0	-1	13	12
13	0	-10	17	7
14	0	1	12	13
15	0	-1	16	15

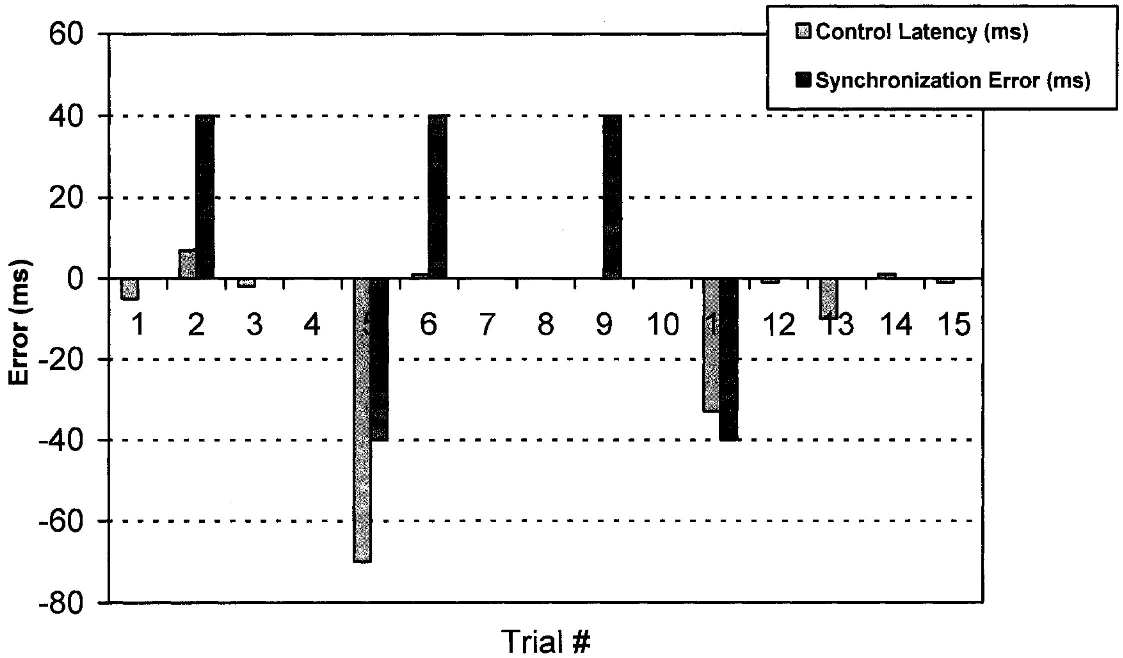


Figure 19: Network Latency and Synchronization Error

To understand how client/server processing speed affected synchronization, accumulation tests were conducted. The first test was to accumulate the synchronization error by repeating "play" and "pause" at one workstation ten times. Five trials were conducted. Timestamps of the other four clients were recorded, and the means of five synchronization error trials on each PC were calculated.

Table 6: Synchronization Error Accumulation Test Result - All operations done at the same workstation (s=standard deviation, MAX=maximum).

Synchronization Error on Each PC (Frame)							
Trial	ΔPC2	ΔPC3	ΔPC4	ΔPC5	s	Mean	MAX
1	1.2	0.8	0.8	0.8	0.200	0.90	1.2
2	2.2	2.0	2.2	2.2	0.100	2.15	2.2
3	2.8	2.4	2.6	2.6	0.163	2.60	2.8
4	4.0	3.8	4.0	4.0	0.100	3.95	4.0
5	4.4	3.6	4.2	4.0	0.342	4.05	4.4
6	4.8	4.6	4.8	4.6	0.115	4.70	4.8
7	5.2	5.4	5.2	5.0	0.163	5.20	5.4
8	4.6	5.2	4.8	4.6	0.283	4.80	5.2
9	4.8	5.2	5.2	4.8	0.231	5.00	5.2
10	5.8	6.0	5.8	5.4	0.252	5.75	6.0

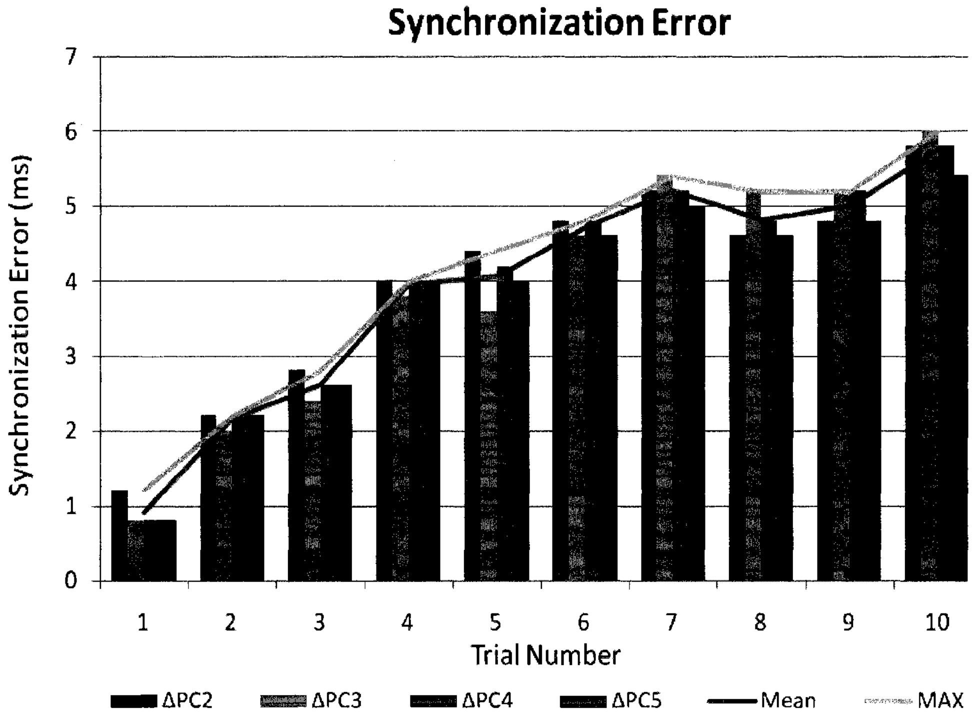


Figure 20: Synchronization Error - Accumulation done at the same PC

The second test accumulated synchronization error by pressing "play" and "pause" on computer 1, 2, 3, 4, and 5 in sequence. Ten trials were completed. Fifty play/pause operations were preformed. The synchronization error on each PC was recorded. Table 8 shows the synchronization error and the mean and maximum error per trial.

Table 7: Synchronization Error Accumulation Test Result - All operations done at different workstations (s=standard deviation, MAX=maximum).

Synchronization Error (Frame)							
Trial #	**ΔPC2**	**ΔPC3**	**ΔPC4**	**ΔPC5**	**Mean**	**Max**	**s**
1	0.4	0.4	0.4	0.4	0.4	0.4	0.00
2	1.2	1.6	1.4	1.4	1.4	1.6	0.16
3	0.8	0.6	0.4	0.0	0.5	0.8	0.34
4	1.0	0.4	0.8	0.6	0.7	1.0	0.26
5	0.4	0.8	1.4	0.4	0.8	1.4	0.47
6	1.0	1.4	2.0	1.4	1.5	2.0	0.41
7	0.4	1.0	0.8	1.4	0.9	1.4	0.42
8	1.0	0.8	1.0	0.4	0.8	1.0	0.28
9	1.8	1.4	1.6	2.0	1.7	2.0	0.26
10	1.2	1.0	2.0	1.0	1.3	2.0	0.48

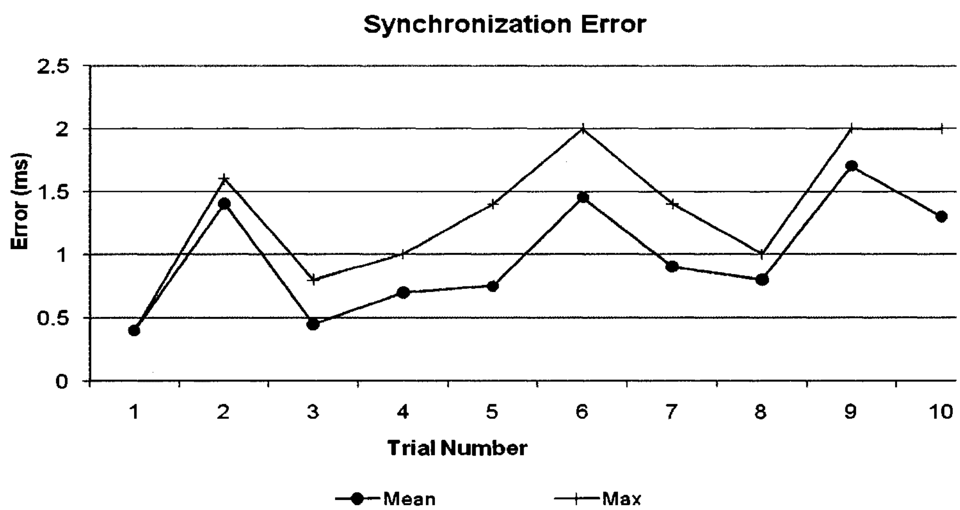


Figure 21: Synchronization Error - Accumulation done at different PC.

6.2.2 Video Control Functions

The video control function was validated from two aspects: user experience and program level. The video control function test result indicated that video control button "Play", "Pause", "FF", "Rewind" and "Stop" work as we expected. If the display was set to "Sync", the shared object *mat.status* was updated properly when an event occurred at the client side. Flash video status on workstations was synchronized by the shared object property.

The event listener monitored the client side video status and mouse status. When the flash video file was at the end, the event listener detected a mouse event and auto rewound the flash video. When a mouse release event occurred over the seek bar, the shared object mat.status was updated automatically. As well, clients were immediately updated by the *onsync()* function.

6.2.3 Whiteboard functions

The Whiteboard function test result showed that the three customized features Freedraw, Erase and Move features work as required. These three features were fully integrated into the Adobe Flash Whiteboard component.

Table 8: Video Control Functions Test Result

Operation Mode	Function	User Experience Test Result	Program Level Test Result
Synchronized	Play	Flash Video was played at all workstations at Sync mode	Mat.play() was called Mat.sync()=true
	Pause	Flash Video was paused at all workstations at Sync mode	Mat.pause() was called Mat.sync()=true
	FF	Flash Video was forwarded 0.1 second at all workstations at Sync mode	Mat.forward() was called Mat.sync()=true
	Rewind	Flash Video was rewind 0.1 second at all workstations at Sync mode	Mat. rewind() was called Mat.sync()=true
	Stop	Flash Video was stopped and rewind to header at all workstations at Sync mode	Mat. stop() was called mat.seek(0) was called Mat.sync()=true
Non-Synchronized	Play	Flash Video was played at all workstations at non-sync mode	Mat.play() was called Mat.sync()=false
	Pause	Flash Video was paused at all workstations at non-sync mode	Mat.pause() was called Mat.sync()=false
	FF	Flash Video was forwarded 0.1 second at all workstations at non-sync mode	Mat.forward() was called Mat.sync()=false
	Rewind	Flash Video was rewind 0.1 second at all workstations at non-sync mode	Mat. rewind() was called Mat.sync()=false
	Stop	Flash Video was stopped and rewind to header at all workstations at non-sync mode	Mat. stop() was called mat.seek(0) was called Mat.sync()=false

Table 9: Whiteboard Functions Test Result

Functions	User Experience Test Result	Function
Handfree	User can draw any shape with mouse	Whiteboard.Freehand
Erase	User can erase all shapes on stage except Flash Video	Whiteboard.Erase
Move	User can move any shapes on stage expect Flash video	Whiteboard.Move

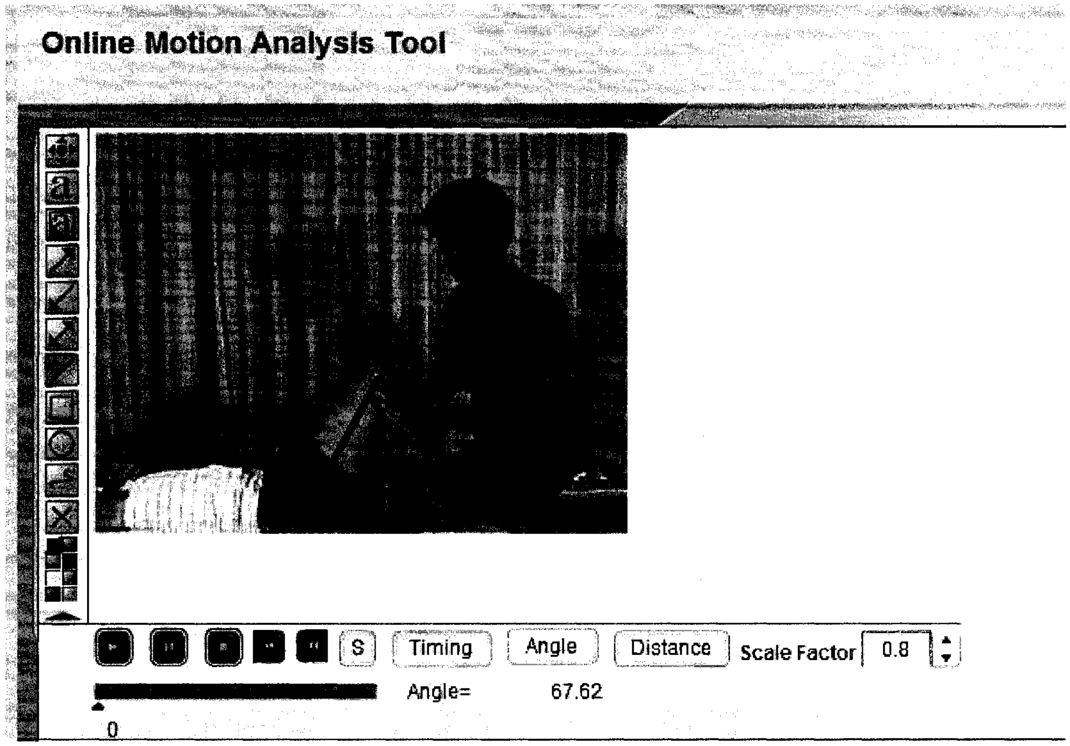


Figure 22: Whiteboard Snapshot

6.2.4 Motion Analysis and Display Mode Functions

The motion analysis function evaluation compared motion analysis angles and distances with a known angle value and distance. "In the test video trial, a subject carried a metal square through the field of view (vertical 561.0 mm, horizontal 540.7 mm). The square provided a distinct 90-degree angle form for measurement" [60].

Regarding the motion analysis results, as shown in Table 11, the average percent error for angle measurements was 2.7% of the criterion measure. The average length measurement error was 1.95 cm. The average percent error for length measurements was 3.0% of the criterion measure. Average root mean square errors were 2.5 degrees for angle measures and 1.952 cm for length measures. Motion analysis errors under 5% are generally considered acceptable [60].

Table 10: Motion Analysis Test Result

Display Mode	Task	User Experience Test Result	Function
Sync	Distance Measurement	Distance value was displayed on the date grid panel	mat.distance
	Angle Measurement	Angle value was displayed on the date grid panel	mat.angle
	Timing Measurement	Duration value was displayed on the date grid panel	mat.timing
Non-Sync	Distance Measurement	Distance value was displayed on the date grid panel	mat.distance
	Angle Measurement	Angle value was displayed on the date grid panel	mat.angle
	Timing Measurement	Duration value was displayed on the date grid panel	mat.timing

Table 11: Angle (Degree) and Length (CM) Results for Comparisons between Measured and Calculated

	Square (90 degrees, 61.03 cm, 40.7 cm)
Angle (average)	92.53 degrees
Angle (RMSE)	0.71
Vertical length (average)	62.95 cm
Vertical length (RMSE)	1.95
Horizontal length (average)	42.57 cm
Horizontal length (RMSE)	1.55

6.2.5 Media Server Performance Test Result

The message drop rate was 1.17% when the Media Server was running at the maximum capacity (10 active connections, 250Kbits bandwidth, and 50% CPU usage).

Server	Cor	Connects	Disconnects	Mbits In/Out	Msgs In/Out/Dropped
defaultVHost	8	105	97	0.548 / 2.59	59074 / 274346 / 3914

Figure 23: Flash Media Server Performance Test Result

Chapter 7. Discussion

7.1 Accessibility Test

Web page load speed depends on the client network connection type. If the connection is dial-up, the page load time increased to 13.2 seconds. If the connection is a typical Digital Subscriber Loop (DSL), the page can be completely loaded in 1 second. The test result showed that the MAT page can be completely loaded in less than 0.61 seconds with an ADSL connection of 1.25bs download speed, which is less than the 6.0 second industry standard.

From the MAT Web application perspective, there is no bottleneck on Web page load speed with a limited number of concurrent connections. If a user experiences problems with low speed on web page loading, the network connection download speed should be checked.

File upload speed also depends on the client network connection. A typical rehabilitation flash video has a size of 1M to 3M. Thirty to 100 seconds are required to upload the flash video file with a 10M LAN or DSL connection.

The connection test showed that Flash Media Server could provide the media streaming service in RTMP, RTMPT and RTMPT over SSL with TCP port 80, 443 or 1935. The test result indicated that the Flash web application can easily establish the connection on a typical network environment including public internet and corporate network like the University of Ottawa campus network.

7.2 Quality Test

Each video control procedure included five sub-procedures: data update at local, data transfer to server, server update, data transfer to each client and data update at each client. We named each sub-procedure as:

- T1: Local data update on Flash Player
- T2: Data transfer from client to server
- T3: Media Server data update
- T4: Data transfer from server to client
- T5: Flash Player data update

T2 and T4 are the time spent on data transmission on the network. Results from network latency and video synchronization testing indicated that varied network latency caused synchronization errors. If network latencies were different between "play_latency" and "pause_latency", synchronization errors could occur. A 70ms network latency difference between the "Play" and "Pause" processes caused 40ms synchronization error on trial 5. As well, a 33ms latency difference caused 40ms synchronization error on trial 11.

If the network latency is stable, network latency will not cause synchronization error between clients. For example, for trials 1, 3,4,7,8,10,12,14 and 15, average network latency was 20ms, but no synchronization error occurred.

The result of trials 2, 6, 9 indicated synchronization errors even through network latencies were very low (less than 7ms). Especially on trial 9, the synchronization error was 40ms with zero network latency difference between "Play" and "Pause" actions. The result indicated the synchronization error was not only caused by varied network latency, but also caused by the client side Flash Player update process.

T1, T3 and T5 are impacted by server performance, client workstation performance and the efficiency of the program. To understand how Flash Player and Media Server performance impacts synchronization, two error accumulation tests were conducted (figure 7, 8).

As shown in figures 19 and 20, the synchronization error increased with multiple play-pause iterations. If all mouse presses were done at the same PC, the error followed increases in trial number. If mouse presses were done on different PC, as shown on Figure 21, the error does not always increase following an increasing of trial number. This is because the error could reduce if the operator PC is changed. For example during the trial 2, PC1 was one frame behind PC2 at step 1 (play and pause button were pressed on PC1). At step 2, play and pause button were pressed at PC2, and the error fell back to zero. If more than 20 operations done, as shown on Figure 20, the error may increase to 6 frames (0.24 second). This would not meet the rehabilitation motion analysis requirement. In order to overcome this issue, a hard synchronization procedure should be added into the pause button script. Once pause button is pressed, the script sync all clients to a same timeframe.

The motion analysis display mode can be controlled with a "Sync" button. If the mode was set to 'Sync", the analysis results were shown on all clients connected to the Media Server. If the display mode was set to "Local", the analysis results were displayed locally, and the shared object was not updated on the Media Server. The result met the design requirements.

The Media server packets drop rate was 1.2% when the maximum capacity was reached. The result met the requirement of 2%.

Chapter 8. Conclusion

The focus of this thesis was to develop and evaluate a cross-platform, cross-device rehabilitation motion analysis tool. The research discussed design criteria, what problematic areas are encountered and what kinds of solutions were proposed to the faced problems.

In chapter 1 and 3 we defined the specific research problems and objects to be addressed later in the methods (Chapter 5) and analysis phase (Chapter 7). A Flash Web Application was selected as the platform for online motion analysis for rehabilitation. Flash 8 was selected as the development tool to implement the design. Different media delivery protocols of Flash application were also been discussed in the methods.

Selecting the Flash Web Application platform to develop a web-based collaboration tool for rehabilitation was an innovation in this field. Sharing and controlling video in real-time via shared objects cross public network is different than existing methods.

To evaluate the Flash based motion analysis tool, accessibility and quality evaluations were conducted. The key challenge of sharing video over network, synchronization, was evaluated from network latency and error accumulation aspects. The evaluation result indicated that the video synchronization error of the web-based motion analysis tool over the Flash Web application platform can meet the real-time collaboration requirements. Acceptable results occurred with a typical network connection. The maximum synchronization error was less than 160 ms.

The motion analysis functions developed with ActionScript have also been evaluated. The average percent error for both angle and length measurements were considered as appropriate for clinical motion analysis.

In summary, the Flash Based Motion Analysis Tool (MAT-F) achieved the objectives. The tool provides an enhanced Whiteboard, using a Flash user interface that integrates Whiteboard features, Flash video sharing, and motion analysis together to offer cross platform motion analysis functions. The tool can work within regular network environments with no extra application installation required at the workstation level. Meanwhile, the enhanced Whiteboard component was integrated with a data conferencing platform including enhanced Whiteboard, chat, and audio and video conferencing system.

One aspect to improve in the next develop phase is synchronization error accumulation. To achieve the goal of keeping the total accumulate latency with a large number of operations at an acceptable level (less than 50 ms), a hard synchronization function must be integrated into the current solution in the next phase development.

References

1. A Primer on the T.120 Series Standard. URL: http:/www.dtic.mil/ieb_cctwg /contrib-docs/T.120/T.120-WP.html [Last checked 26 April 2007].
2. Lemaire ED. A low-cost, easy-to-use platform for telehealth needs. Canadian Healthcare Technology. 2003; 19: 23
3. Klutke PJ, Mattioli P, Baruffaldi F, Toni A, Englmeier KH. The Telemedicine benchmark – a general tool to measure and compare the performance of video conferencing equipment in the telemedicine area. Computer Methods and Programs in Biomedicine, 1999; 60(2): 133-141
4. Firewall TCP Ports. URL: http://www.linnetsol.co.uk/port-filter.asp [Last checked 26 April 2007].
5. Lee M. Telehealth in Canada: Clinical Networking, Eliminating Distances. URL: Http://www.canarie.ca/funding/ehealth [Last checked 26 April 2007].
6. Liu G, Lemaire ED. Data Conference in Health Care. Journal of Telemedicine and Telecare. 2005; 11 (7): 339-346.
7. Adobe Flash Player Version Penetration. URL: http://www.adobe.com/products /player_census/flashplayer/version_penetration.html [Last checked 26 April 2007].
8. Lee M. Telehealth in Canada: Clinical Networking, Eliminating Distances. URL: Http://www.canarie.ca/funding/ehealth [Last checked 26 April 2007].
9. Macromedia Flash Player: The universal rich client for delivering effective experiences. URL: http://www.macromedia.com/software/flashplayer/ [Last checked 26 April 2007].

10. Macromedia Flash Player Statistics. URL: http://www.macromedia.com/software /player_census/flashplayer/ [Last checked 26 April 2007].
11. Macromedia Flash Player and Browser Matrix. URL: http://www.macromedia.com /software/player_census/flashplayer/bundling_matrix.html, [Last checked 26 April 2007].
12. Lemaire ED, Greene G. Continuing Education in Physical Rehabilitation Using Internet-based Modules. Journal of Telemedicine and Telecare, 2002; 8: 19-24
13. Microsoft NetMeeting. URL: http://www.microsoft.com/windows/netmeeting, [Last checked 26 April 2007].
14. Windows Networking. URL: http://www.wown.com/articles_tutorials /tcpipprt.html. [Last checked 26 April 2007].
15. WebEX. URL: http://www.webex.com, [Last checked 26 April 2007].
16. Netspoke. URL: http://www.netspoke.com, [Last checked 26 April 2007].
17. Wimba. URL: http://www.horizonlive.com, [Last checked 26 April 2007].
18. Lemaire ED. Macromedia Flash: A low-cost, easy-to –use platform for telehealth needs. Journal of Canadian Healthcare Technology, 2003; 19: 23
19. Smart Technologies. URL: http://www.smarttech.com, [Last checked 26 April 2007].
20. Webopedia. URL: http://systems.webopedia.com/TERM/A/application_sharing.html. [Last checked 26 April 2007].
21. Mattioli P, Klutke PJ, Baruffaldi F, Viceconti M, Toni A, Englmeier KH. A study of the application sharing capabilities in telemedicine. Journal of Computer Methods and Programs in Biomedicine, 1999; 58: 89-97

22. Grclive. URL: http://web.grclive.com, [Last checked 26 April 2007].

23. SCOPIA® Desktop Video Conferencing. URL: http://www.radvision.com /Products/Desktop. [Last checked 26 April 2007]

24. HelpMeeting. URL: http://www.helpmeeting.com[Last checked 26 April 2007]

25. Lotus Sametime. URL: http://www.lotus.com/products/product3.nsf/wdocs/homepage, [Last checked 26 April 2007]

26. VNC. URL: http://www.uk.research.att.com/archive/vnc/, [Last checked 15 October 2004]

27. TightVNC Software. URL: http://www.tightvnc.com, [Last checked 26 April 2007]

28. RealVNC. URL:http://www.realvnc.com, [Last checked 26 April 2007]

29. Brodlie KW, Duce D A, Gallop JR, Walton JPRB, Wood JD. Distributed and collaborative visualization. Computer Graphics Forum, 2004; 23(2): 223-251

30. Sung MY, Kim MS, Kim EJ, Yoo JH, Sung MW. CoMed: a Real-time Collaborative Medicine System. International Journal of Medical Informatics, 2000; 57(2-3); 117-126

31. Mendonca EA, Chen ES, Stetson PD, McKnight LK, Lei J, Cimino JJ. Approach to mobile information and communication for health care. International Journal of Medical Informatics, 2004; 73: 631-638

32. Ortega EM, Burgun A, Le Diff F, Le Beux P. Collaborative environment for clinical reasoning and distance learning sessions. International Journal of Medical Informatics, 2003; 70: 345-351

33. Lemaire ED, Boudrias Y, Greene G. Low-Bandwidth, Internet-based Telehealth for Physical Rehabilitation Consultations. Journal of Telemedicine and Telecare, 2001; 7: 82-89

34. Lemaire ED, Boudrias Y, Greene G. Technical evaluation of a low-bandwidth, internet-based approach for telehealth consultations. Journal of Telemedicine and Telecare, 2000; 6: 163-167

35. Klutke PJ, Baruffaldi F, Mattioli P, Toni A, Englmeier KH. Guidelines for multipoint videoconferencing using low-cost, PC-based equipment. Journal of Telemedicine and Telecare, 1999; 5(3):198-202

36. Lemarie ED, Fawcett JA. Using NetMeeting for remote configuration of the Otto Bock C-Leg: technical considerations. Prosthetics and Orthotics International, 2002; 26: 154-158

37. Lomax A, Grossmann M, Cozzi L, Tercier PA, Boehringer T, Schneider U, Logean M, Volken W, Ratib O, Miralbell R. The exchange of radiotherapy data as part of an electronic patient-referral System. International Journal of Radiation Oncology Biology Physics, 2000; 47(5): 1449-1456

38. Bergh B, Schlaefke A, Pietsch M, Garcia I, Vogl TJ. Evaluation of a "no-cost" Internet technology-based system for teleradiology and co-operative work. European Radiology, 2003; 13(2): 425-434

39. Flash Player 7 for Pocket PC. URL: http://www.adobe.com/products/flashplayer_pocketpc, [Last checked 26 April 2007]

40. Using the Flash Remote MX. URL: http://livedocs.adobe.com/flashremoting/mx/Using_Flash_Remoting_MX/intro2.htm, [Last checked 26 April 2007]
41. Adobe Feature Tour. URL: http://www.adobe.com/products/flashmediaserver/productinfo/features/#f-1-1, [Last checked 26 April 2007].
42. Flash Media Server. URL: http://livedocs.adobe.com/fms/2/docs/wwhelp/wwhimpl/common/html/wwhelp.htm?context=LiveDocs_Parts&file=00000117.html, [Last checked 26 April 2007].
43. Harpal D, Pamela G. Forducey. Implementation and Evaluation of Information Technology in Telemedicine. Processing of the 39th Hawaii International Conference on System Sciences, 2006
44. Bashshur R, Shannon G, Sapei H. Telemedicine Evaluation. Telemedicine and e-Health. 2005; 3(2): 97
45. Grigsby J, Angela G. Brega, Patricia A. Devore. The Evaluation of Telemedicine and Health Services Research, Telemedicine and e-Health, 2005; 11(3): 317-328
46. Lohr KN. Medicare: a strategy for quality assurance. Washington, DC: National Academy Press, 1990.
47. Adobe Product Info. URL: http://www.adobe.com/products/flashcom/productinfo/editions/, [Last checked 26 April 2007]
48. Collaborative Care. URL: http://www.collaborativecare.net/, [Last checked 26 April 2007]

49. Worldwide ISP Security Report. URL: http://www.arbor.net/downloads/Arbor_Worldwide_ISP_Security_Report.pdf, [Last checked 26 April 2007]

50. Unicast, Broadcast and Multicast. URL: http://www.erg.abdn.ac.uk/users/gorry/course/intro-pages/uni-b-mcast.html, [Last checked 26 April 2007]

51. Introducing the Flash Communication Server. URL: http://www.devarticles.com/c/a/Flash/Introducing-the-Flash-Communication-Server/3, [Last checked 26 April 2007]

52. Using Flash Remote MX. URL: http://livedocs.adobe.com/flashremoting/mx/Using_Flash_Remoting_MX/intro2.htm, [Last checked 26 April 2007]

53. Developing Secure Application. URL: http://livedocs.adobe.com/fms/2/docs/wwhelp/wwhimpl/common/html/wwhelp.htm?context=LiveDocs_Parts&file=00000117.html, [Last checked 26 April 2007]

54. Flash Media Server Feature. URL: http://www.adobe.com/products/flashmediaserver/productinfo/features/, [Last checked 26 April 2007]

55. What is local shared object?. URL: http://www.adobe.com/cfusion/knowledgebase/index.cfm?id=tn_16194, [Last checked 26 April 2007]

56. Flash Player 8. URL: http://www.adobe.com/products/flashplayer/productinfo/systemreqs/flashplayer8/, [Last checked 26 April 2007]

57. SelfSEO. URL: http://www.selfseo.com/website_speed_test.php, [Last checked 26 April 2007]

58. ActionScript. URL: http://en.wikipedia.org/wiki/ActionScript, [Last checked 26 April 2007]

59. ECMAScript. URL: http://en.wikipedia.org/wiki/ECMAScript#Dialects, [Last checked 26 April 2007]

60. Lemaire ED. A Shockwave Approach for Web-Based Clinical Motion Analysis. Telemedicine and e-Health, 2004; 10(1): 39-43

61. RTMP. URL: http://en.wikipedia.org/wiki/Real_Time_Messaging_Protocol, [Last checked 11 May 2007]

62. Information priority within the RTMP protocol. URL: http://www.adobe.com/cfusion/knowledgebase/index.cfm?id=tn_16458, [Last checked 26 April 2007]