MCS: Modular, Cross-platform and Secure application for Remote Assistance and Monitoring Tasks

Xavier Vallejo	Eusebi Calle	Antonio Bueno	Jose L. Marzo
Institute of Informatics	Institute of Informatics	Institute of Informatics	Institute of Informatics
and Applications	and Applications	and Applications	and Applications
University of Girona	University of Girona	University of Girona	University of Girona
17071 Girona	17071 Girona	17071 Girona	17071 Girona
xvallejo@atc.udg.edu	eusebi.calle@udg.edu	antonio.bueno@udg.edu	joseluis.marzo@udg.edu

Abstract

MCS is a multimedia tool for rehabilitation specialists, that allows remote assistance and monitoring of patient's activities. This tool is the evolution of the work done in 2005-06 between the BCDS research group of the UdG and the Multiple Sclerosis Foundation (FEM in Spanish) in Girona under the TRiEM project. Multiple Sclerosis (MS) is a neurodegenerative disease (ND) that can provoke significant exhaustion in patients even by the simple act of going to the medical centre for rehabilitation or regular check ups. The tool presented in this paper allows the medical staff to carry out distance patient consultations and other activities from their home, minimizing house calls and travel time for the medical consultant. It has been enhanced by developing new components to improve modularity and security. MCS also has a hybrid P2P architecture and consists essentially of a cross-platform videoconference system, with audio/video recording capabilities. The system can easily be extended to include new capabilities such as, among others, asynchronous activities whose result can later be analyzed by the medical personnel.

1 Introduction

This paper presents an application to assist medical personnel in telerehabilitation tasks using basic broadband Internet connections and readily available hardware. It is the continuation of the work carried out in the TRiEM project, a joint effort in Girona in 2005-06 between the BCDS research group of the UdG [1] and the Multiple Sclerosis Foundation (FEM in Spanish and Catalan), to develop a tool for MS telerehabilitation.[1]

1.1 The Illness

Multiple sclerosis (MS) is a chronic, inflammatory disease that affects the central nervous system. MS can cause a variety of symptoms, including changes in sensation, visual problems, muscle weakness, depression, difficulties with coordination and speech, severe fatigue, cognitive impairment, problems with balance, uncontrollable rises in body temperature, and pain. MS will cause impaired mobility and disability in more severe cases [2]. The name "multiple sclerosis" refers to the multiple scars (or scleroses) on the myelin sheath (the fatty layer that surrounds and protects neurons, helping them carry electrical signals). This scarring causes symptoms which vary widely depending on which signals are interrupted.

The prevalence of MS is significant [3]; in equatorial areas it is around 0.0001% while

[1] This work was supported in part by the MCYT (Spanish Science and Technology Ministry) research project TEC2006-03883/TCM and the Generalitat of Catalonia's research support program (SGR 00296).

in southern Europe the rate is 6-14 times higher and in northern Europe it is 30-80 times higher. Numbers in North America are on the same range as in Europe.

1.2 TRiEM project

The first prototype [4] developed in TRiEM (Catalan acronym for "telerehabilitation and MS") allowed medical specialists to carry out patient consultations remotely, with the double objective of making a better use of the funds in order to provide longer and more continuous follow-ups and to improve the patient's quality of life.

The system was also designed, at the outset to use as far as possible standard available low-cost infrastructure; both in terms of computer equipment (CPU, graphic cards, webcams) and communication systems. For example, the prototype can sustain acceptable two-way audio and video quality with a basic broadband connection, which in Spain currently consists of 6 Mbps downstream and 300 kbps upstream.

The tool was built on top of an open source instant messaging application [5] and started as a standard videoconference application. In the initial prototype, in order to adapt it as far as possible to specialist's needs and patient's characteristics, additional capabilities of audio/video recording and multimedia management were also included.

As for related work, the field of telerehabilitation has generated abundant literature. For example, Lauderdale and Winters [6] already explored the possibility of using the then current teleconferencing products for telerehabilitation in 2000. In contrast, the American Occupational Therapy Association expressed, in 2005 [7], concern about the lack of published research regarding the use of telerehabilitation methods for specific occupational therapy follow-up services. Nowadays, it is still difficult to find telerehabilitation tools that work on more than just one aspect of the rehabilitation process.

The paper is structured as follows: After this introduction, the development and status of the tool is discussed in Section 2 and Section 3 is dedicated to explaining the main features of the application in more detail. The paper concludes with the customary conclusions and future work.

2 Development and status

Once we had the first prototype, the BCDS group continued to work on a newer version with more features that will be detailed in section 3.

This section is structured into two parts: First, the architecture of the application is explained in a general view. After that, we will introduce the technology that MCS uses.

2.1 Architecture

The MCS application uses a hybrid P2P architecture (see figure 1): a server-centric component for session management, information and remote control messages, and a peer-to-peer component for the higher bandwidth consuming live audio and video streams.

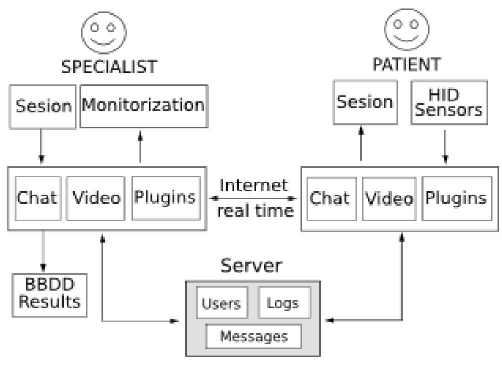


Figure 1: General structure of the application.

Peer-to-peer connections can also be used to transmit information of other activities, apart from audio and video streams. The most significant advantage of this type of connection is a better bandwidth and less latency. However, one of the users has to open one or more ports of the user's router - a not altogether trivial operation - .

2.2 Technology

The application is written in Java. Java is an Object Oriented language that facilitates quick development for the developer and it includes a suite of ready to use libraries. According to RIAStats [8], 68,96 % of the computers connected to the Internet have Java 1.5 or greater installed in their computers. In addition, thanks to JNI technology, Java can execute native code (for instance, calling a DLL library).

As for the communications part, MCS makes extensive use of the XMPP open standard [9]. This protocol transmits lightweight, human-readable and easily understandable messages in XML form and it was easily extended for the additional features of the TRiEM prototype.

Using Java as a programing language, and XMPP as a communications library, a developer can create new functionalities for the application in a few weeks using the plugin system of MCS application. Each plugin (or extension) can add new possibilities to the application. A plugin can work alone, or with the collaboration of others plugins. An automatic system for adding/updating plugins has also been added to our application.

3 Modularity

After the first prototype, the specialists of the FEM were very interested with new ideas to apply in a telerehabilitation session [10]. In this section, two important ideas are explained: a subset of new activities to be done in a specialist-patient session, and the use of sensor devices to monitor patient data.

3.1 Activities

One of the needs of the specialists was that all the activities can be done in both online and offline mode. We refer to online mode when the specialist supervises in real time the patient's activities. Additionally, with MCS the specialist can send the patient a number of activities to do. In offline mode, the patient can do the exercises whether the specialist is connected or not. Once the user does the exercise, the results are sent back to the doctor.

A brief classification of the activities that are implemented in MCS application, are as follows:

- Cognitive: Memory, Labyrinth.
- Speech and language therapy (SLTs): Basic movements.
- Oculomotor nerve (using a joystick with a hand): Road, Labyrinth, Memory, Figures, Fruit Salad pairs.
- Inferior extremities: Dance mat.

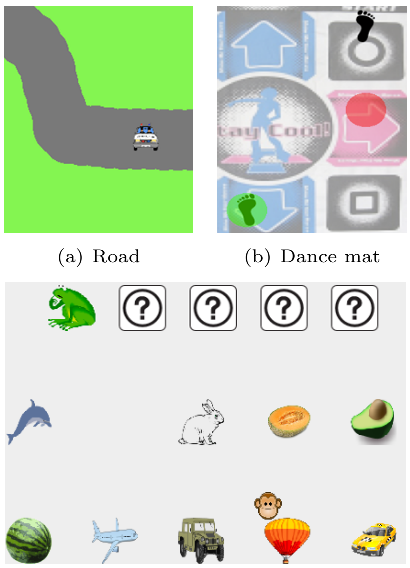
In this section, we will explain four activities that we consider important to understand how they work.

- Memory: The activity consists of memorizing a group of elements that appear on the screen in a limited time period. The next step is to recognize and identify the initial objects inside a bigger group of elements. The user has to select the elements using a joystick.
- Road: On screen appears a road that the patient has to drive along on the correct path to the goal. There is no time limit, but how much time is needed to complete it is recorded. To create a circuit you simply need to draw it with a drawing program.
- Dance mat: Instead of using a joystick, the patient needs to have a dance mat. The specialist sends a sequence that the

patient has to reproduce. As for hardware compatibility, we have tested this with both PC and Playstation 2 dance mat with satisfactory results.

- Labyrinth: The patient has to find the end of the maze moving a character that is trapped inside it. They cannot see all the labyrinth, so they have to explore it within a certain time limit. Another feature is that the patient can ask for help: the program will indicate, with an arrow, where the exit is.

In figure 2 we show three of the previously explained activities.



(a) Road

(b) Dance mat

(c) Memory

Figure 2: Images of three activities in use on the application.

3.2 Sensor devices

All of the previously presented activities use a joystick to interact with the user (except the dance mat activity). Although using these devices can help in our context (patients with MS), it is clear that the platform is open enough to add more activities and devices. Moreover, the MCS application can be expanded to other environments such as the Ambient Intelligence or Tele-monitoring.

Using different sensors, we can monitor more aspects of a person (acquire and process data). At the moment, the BCDS group has been focused on monitoring inferior extremities. For instance, we can create a low-cost sole foot with FSR (Force Sensing Resistors, see figure 3) or piezoelectric sensors.

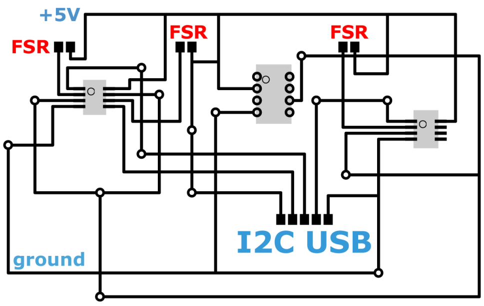


Figure 3: A simple schematic design to use FSR sensors.

As for related work, there are others projects which involve pressure sensors. For example, Taherian, Pias et al. [11] created a wireless sensor system for athletes. In Spanish, Battistella et al. [12] designed a sole foot using piezoelectrical sensors and an acquisition card connected in a PC.

As an example, we will present two sensor devices that we are working on:

- The Ultrasensor: This device can measure short distances of up to 100 meters (see Figure 4). It works with the I2C protocol and the data is read by a USB-SERIAL UART device.
- The Accelerometer: An accelerometer is a device that measures proper acceleration, i.e. the acceleration experienced relative to freefall. For our tests, we use the ADXL332 chip from Analog Devices. It is a 2-Axis low-power accelerometer.

The following figure (figure 4) shows what the components are like.

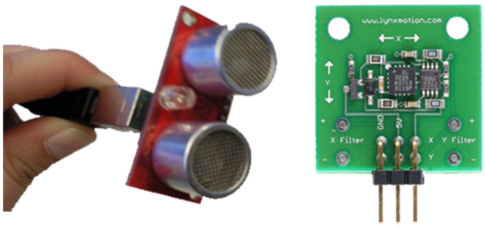


Figure 4: The SRF08 ultrasensor and the 2-axis ADXL332 chip.

As shown in figure 5, a simple proof of concept consists of two graphs representing each axis (X and Y), and below a 3D box that it moves according to the values obtained of the accelerometer.

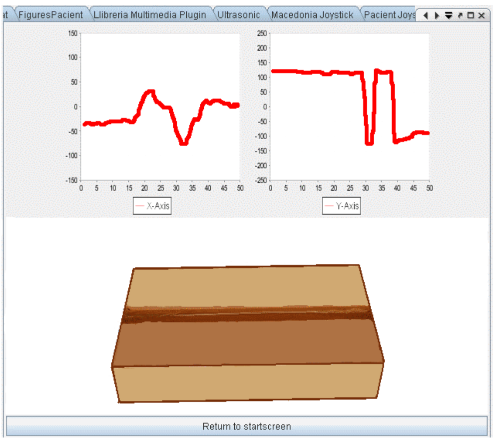


Figure 5: The ADXL332 chip with the X-axis and Y-axis graphs.

4 Cross-platform

Due to the nature of Java and the JVM (Java Virtual Machine), one of the main features of MCS is its compatibility with various Operating Systems. MCS can be executed in Windows (XP and Vista), Linux (Ubuntu, OpenSuse, etc.) and Mac OS X (Leopard and Snow Leopard) environments.

The next two figures (see figures 6 and 7) show the application running a video-conference session between two users. Each component can be set in a flexible way as can be seen by looking at the pictures.

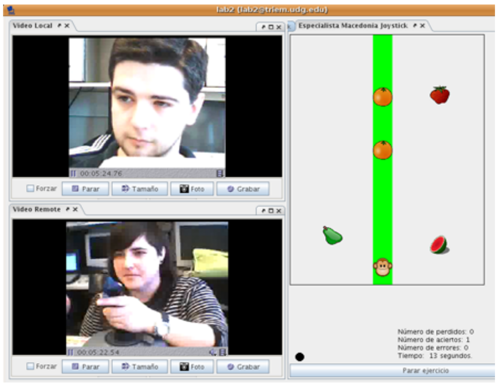


Figure 6: A live sesion running in Ubuntu.

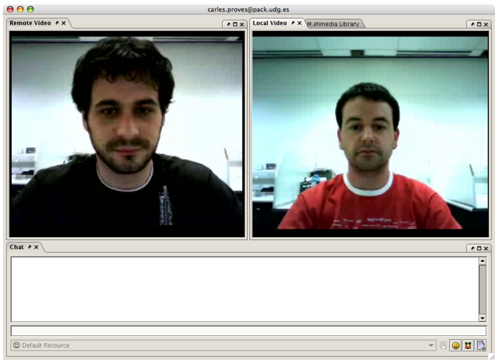


Figure 7: Running in Mac OS X with a different view.

With the cross-platform idea in mind, all extensions developed are compatible with Windows, Linux and Mac OS X (if it is possible). However, depending on the needs of the plugin, making it multi platform is not an easy task.

As an example, we will introduce a new plugin explaining the needs and all the options we looked at, taking into consideration the cross-platform issue.

4.1 Video by Streaming

- Objective: A user (the patient) could watch a pre-recorded video recommended by a contact (the specialist).
- Approaches:
  1. Peer-to-peer: If one user has the video on their computer, they could try to send it using a video stream. The main disadvantage is the upload bandwidth which is very limited on a typical home connection.
  2. From a server: Here all the videos would be stored in a server. The most important advantages are not being limited by upload bandwidth, the possibility of repositories, and 24/7 availability. One disadvantage, however, is that the server will need to have enough bandwidth.

  The second option seems to fit better into our context, and so it is the chosen one.
- Technical options:
  1. Use a Flash player and play a Flash movie.
  2. Incorporate a media player in Java and try to reproduce video files (i.e. mp4).
  3. Integrate a modern Web Browser with multimedia capacities (i.e. adding a Flash plugin) into our application.

As a bonus, the third option allows more actions rather than only watching videos. The next table shows all the different options we tested to integrate a Web Browser:

1. Flash Player:
a1 = JFlashPlayer
a2 = JFlash

2. Media Player:
b1 = JMF
b2 = JavaFX

3a. Native Web Browser:
c1 = JEditorPane
c2 = JDIC
c3 = JDIC Plus
c4 = Lobo
c5 = SWT
c6 = The Flying Saucer Project

3b. Firefox Runtime Web Browser:
d1 = JREX
d2 = JX Browser
d3 = Jazilla
d4 = Mozswing [13]

	Multi?	Free?	Comments
a1	✗	✗	Only Windows
a2	✓	✓	Only Flash 3
b1	✓	✓	Abandoned
b2	✓	✓	Not compatible
c1	✓	✓	Can't play video
c2	✗	✓	Abandoned
c3	✗	✓	Only Windows
c4	✓	✓	Needs JavaFX
c5	✓	✓	Not compatible
c6	✗	✓	Can't play video
d1	✗	✓	Old Engine
d2	✓	✗	Expensive
d3	✗	✓	Abandoned
d4	✓	✓	Selected!

5 Security

Last but not least, user information must be protected from any unauthorized access. In addition, medical data has to comply with a strict series of laws and rules. All the connections that MCS make to the server are encrypted using TLS/SSL algorithms.

Three topics will be discussed in this section: the use of digital certificates, the mobility of the data in an external device and the cipher of the data.

5.1 CA (Certificates Authority)

A digital certificate is a document that a third entity trusts the link between the user identity and their public key. One of the most commonly used standards, and the one that we use to implement CA, is X.509.

Our application supports authentication by X.509 certificates using the Openfire [14] server (an open source XMPP server). If the application can not find a certificate, it can try to authenticate by user and password.

The next picture (figure 8) shows the process needed to create a CA. There are two main actors: the user (the requester) and the server (the Certificate Authority).

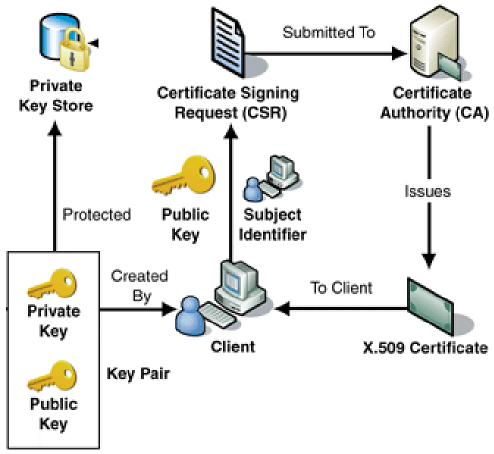


Figure 8: Diagram showing the cycle to create a CA.

For our tests, we made a self-signed server certificate, and then we generated the rest of certificates for the users. To create them we used the OpenSSL [15] tool.

5.2 External storage

Instead of saving all the user data in the computer, we added an extra option to store this information in an external device such as a pendrive or an SD card. For the user, using the device would be like the credit card (with some differences).

- The external device will have a unique user CA.
- External memories have high capacity and low cost.
- In case of theft, personal data would be safe from thieves.
- If there is any problem with the computer or the application, all the data can be extracted immediately.

5.3 Cryptography

If a user wants extra security, we can easily add an extra security layer encrypting the external device. There are many tools to encrypt information, but we decided to use Truecrypt [16] for the following reasons:

- It creates a virtual encrypted disk within a file and mounts it as a real disk.
- Encryption is automatic, real-time (on-the-fly) and transparent.
- Encryption algorithms: AES-256, Serpent, and Twofish.

We did some performance tests using a pendrive with encryption and the results were satisfactory.

6 Conclusion

This paper has presented the MCS application, an extensible tool for remote assistance, and monitoring for rehabilitation. A high-level description of the application development as well as specific technical details have been presented, with particular attention to the tool's extensibility, cross-platform design and security.

In the short-term we plan on evaluating different wireless transmission methods in order to enhance the sensor devices and offer better user autonomy. In parallel, this tool is being used under medical supervision with the double objective to check the suitability with real patients and improve the performance of the application.

References

[1] *Broadband Communications and Distributed Systems (BCDS) research group*. 2010. University of Girona. 11 march 2010. http://bcds.udg.edu/.

[2] *What is Multiple Sclerosis? What Causes MS?*. 2010. DailyStrength. 15 march 2010. http://www.webmd.com/multiple-sclerosis/guide/what-is-multiple-sclerosis.

[3] Blake, D.J., Bodine, C., *An overview of assistive technology for persons with multiple sclerosis*, Journal of Rehabilitation Research and Development, vol. 39 Number 2, March/April 2002, p. 299-312.

[4] Bueno, A., Guadall, C., Marzo, J.L., Harle, D., *TRiEM: An application for MS TeleRehabilitation*. Proceedings of the First International Conference of E-medical Systems. October 2007.

[5] *JBother, a Java Jabber Client*. 2006. Adam Olsen. 15 march 2010. http://www.jbother.org/.

[6] Lauderdale, D.E., and Winters, J.M., *Evaluation of nine IP teleconferencing products (2000)*. Proceedings of the RESNA 2000 Annual Conference: Technology for the New Millennium.

[7] Wakeford, L., Wittman, P.P., White, M.W., Schmeler, M.R., *Telerehabilitation position paper*. Am J Occup Ther 2005 Nov-Dec; 59(6): 656-60.

[8] *Rich Internet Application Statistics*. 2010. Dreamingwell. http://riastats.com/#.

[9] *XMPP Standards Foundation*. 2010. XMPP Standards Foundation. 15 march 2010. http://xmpp.org/.

[10] Bueno, A., Marzo, J.L. and Vallejo, X. *AXARM: An Extensible Remote Assistance and Monitoring Tool for ND Telerehabilitation*. In Electronic Healthcare, First International Conference, eHealth 2008, London, UK, September 8-9, 2008. Revised Selected Papers, p. 106-113.

[11] Taherian, S., Pias, M., Harle, R. *Profiling Sprints using On-Body Sensors*. In Proc of Sixth IEEE International Workshop on Sensor Networks and Systems for Pervasive Computing, Mannheim, Germany, March 2010.

[12] Battistella, Manuel., Lescano, Claudia., Rodrigo, Silvia. and Rodolfo, Rodrigo. *Registrador de Presiones Plantares en Condiciones Dinámicas*. 17th Argentinian Congress on Biomedical Engineering, Ciudad del Rosario, October 14-16, 2009.

[13] *MozSwing*. 2008. mozdev.org . 23 march 2010. http://mozswing.mozdev.org/.

[14] *Openfire Server*. 2010. Ignite Realtime. 23 march 2010. http://www.igniterealtime.org/.

[15] *OpenSSL: The Open Source toolkit for SSL/TLS*. 2010. The OpenSSL Project. 23 march 2010. http://www.openssl.org/.

[16] *TrueCrypt - Free open-source disk encryption software*. 2010. TrueCrypt Foundation. 23 march 2010. http://www.truecrypt.org/.