We are IntechOpen,
the world's leading publisher of
Open Access books
Built by scientists, for scientists

6,900
Open access books available

186,000
International authors and editors

200M
Downloads

Our authors are among the

154
Countries delivered to

TOP 1%
most cited scientists

12.2%
Contributors from top 500 universities

WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Mconf: An Open Source Multiconference System for Web and Mobile Devices

Valter Roesler[1], Felipe Cecagno[1],
Leonardo Crauss Daronco[1] and Fred Dixon[2]
[1]Federal University of Rio Grande do Sul,
[2]BigBlueButton Inc.,
[1]Brazil
[2]Canada

1. Introduction

Deployment of videoconference systems have been growing rapidly for the last years, and deployments nowadays are fairly common, avoiding thousands of trips daily. Video conferencing systems can be organized into four groups: Room, Telepresence, Desktop and Web.

1.1 Groups of videoconference systems

Room videoconference systems are normally hardware based and located in meeting rooms or classrooms, as seen in Fig. 1, which shows examples of a Polycom[1] equipment. Participants are expected to manually activate and call a remote number in order to begin interacting. Other solutions of room videoconference systems are Tandberg[2] (which is now part of Cisco), Lifesize[3] and Radvision (Scopia line)[4].

Telepresence videoconference systems are a variation of room systems in that the room environment and the equipments are set in order to produce the sensation that all participants are in the same room, as shown in Fig. 2, which shows the Cisco Telepresence System[5]. To accomplish the "presence sensation", the main approaches are: a) adjust the camera to show the remote participant in real size; b) use speakers and microphones in a way that the remote sound comes from the participant position; c) use high definition video in order to show details of the participants; d) use a complementary environment, as the same types of chairs, same color in the rooms, and same type of table on the other sides. The same vendors of room videoconference systems provide also telepresence solutions.

---

[1] http://www.polycom.com/

[2] http://www.tandberg.com/

[3] http://www.lifesize.com/

[4] http://www.radvision.com/

[5] http://www.cisco.com/

Desktop videoconference systems are a variation of room systems in that instead of dedicated hardware equipment to perform the videoconference, they use software and off-the-shelf webcams, as showed in Fig. 3, illustrating Vidyo[6] and Vsee[7].

Fig. 1. Polycom room videoconference systems

Fig. 2. Cisco telepresence videoconference system

With desktop systems, a user installs and launches the software on their computer. With the rapid advancement of personal computer hardware and high-definition webcams, the gap in quality between room video conferencing and desktop video conferencing is narrowing. Some other examples of desktop videoconference solutions are EVO[8], Skype[9], ooVoo[10], Megameeting[11], Ekiga[12] and Sightspeed[13]. Previously hardware-only vendors are also

---

[6] http://www.vidyo.com/

[7] http://vsee.com/

[8] http://evo.caltech.edu/evoGate/

[9] http://www.skype.com/

[10] http://www.oovoo.com/

[11] http://www.megameeting.com/

[12] http://ekiga.org/

providing desktop systems as well, like Radvision (Scopia Desktop line) and Polycom (Polycom PVX). Desktop systems can be used as a means of performing personal videoconferences, or, when used with a LCD projector, they serve as low-end room videoconferences or even low-end telepresence videoconferences.


Fig. 3. Vidyo and Vsee Desktop videoconference systems

One example of a desktop system is IVA (Interactive Video and Audio), whose architecture is depicted in Fig. 4, showing one local "Telecentre", where the teacher and local students stand, and many "Remote Points", where the remote students stays. Telecentre and Remote Points communicate through superior quality (roughly 2 Mbps each point) to provide high-quality audio and video, allowing to the teacher perform his/hers classes synchronously keeping the class pedagogical quality [Roesler, 2009] [Roesler, 2010]. The teacher sees each remote point through a 46" television, and each remote point sees the teacher through a 46" television and also the teacher´s presentation, as his/hers computer screen is also transmitted.
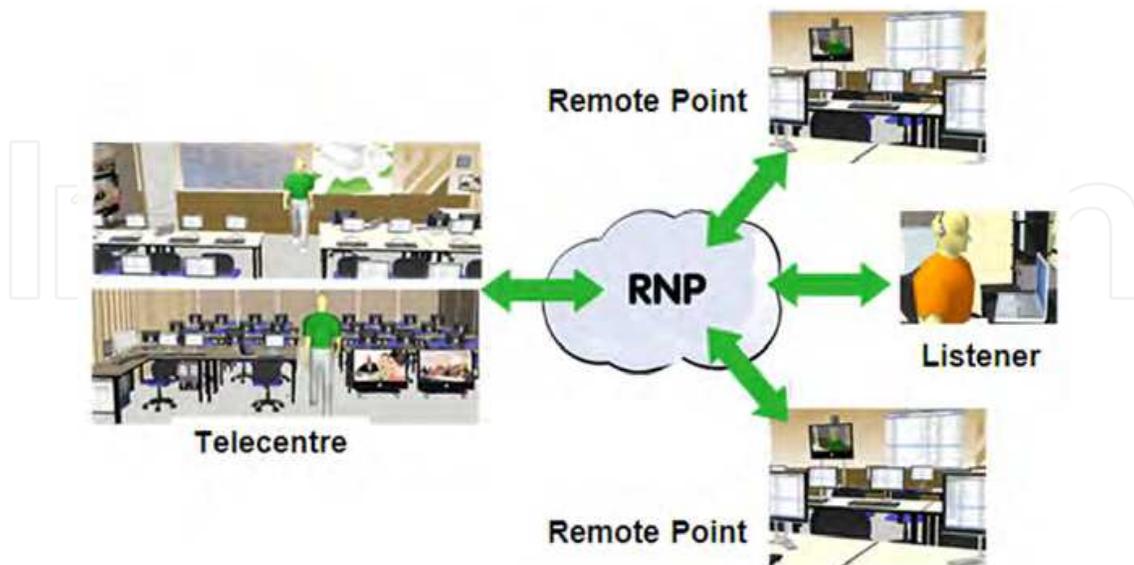

Fig. 4. IVA main architecture

---

13 http://www.sightspeed.com/

Web-based videoconference systems, or web conference systems, run within a web browser, as seen in Fig. 5, which shows Adobe Connect[14] and BigBlueButton[15]. The advantage for users of web conferencing systems is the simplicity of deployment: users need only open a hypertext link within their browser to launch the system. Another advantage is interoperability among different operational systems: users may be running Chrome in Linux, Internet Explorer in Windows, Safari in MacOS, and so on, but everyone still has the same experience.

Fig. 5. Adobe connect and BigBlueButton web videoconference systems

While some web conferencing systems are open source, such as BigBlueButton, or free to use, such as Google plus hangout[16], most require purchase of licenses or subscription from a hosting vendor. Examples of hosted systems are, Webex[17], GotoMeeting[18], Vyew[19], Anymeeting[20], Mikogo[21], and Buddymeeting[22].

1.2 Other issues related to videoconference systems

One issue with using any of the conferencing systems is lack of interoperability. In order to accomplish interoperability among different vendors, common standards must be followed. The two most prominent standards are H.323 [ITU-T H.323, 2009] and SIP (Session Initiation Protocol) [Rosenberg, 2002], and both are used worldwide to allow interoperability among different systems. The basic idea of these protocols is to have some common messages to initiate the session and exchange capabilities. Using the information exchanged, the different parties in the conference can agree in the same group of audio and video codecs, as well as the data protocol, allowing the interoperability.

Sometimes the MCU (Multipoint Control Unit) is responsible for adapting different user bandwidths and codecs, composing many user videos in a single transmission, however, this adaptation have its costs in terms of processing power and delay inserted in the videoconference.

---

[14] http://www.adobe.com/products/adobeconnect.html
[15] http://www.bigbluebutton.org/
[16] https://plus.google.com/
[17] http://www.webex.com/
[18] http://www.gotomeeting.com/
[19] http://vyew.com
[20] http://www.anymeeting.com/
[21] http://www.mikogo.com/
[22] http://www.buddymeeting.com/

Some videoconference systems, like Vidyo and Google plus hangout, follow a scalable video encoding. Scalable video encoders are designed to include more flexibility to multimedia systems. Basically, a scalable encoder generates a multi-layer output, e.g. an output composed by multiples streams. A special stream, called as *base layer*, allows the video reconstruction with minimum detail level, while the others streams, called *enhancement layers*, can increase the video quality in a complementary way (each layer enhances the video generated by the previous one) [Huang, 2007].

The scalable approach is well suitable to multi-rate users, as represented in Fig. 6 for three layers. A user in a low bandwidth environment (a congested 3G connection, for example), receives only one layer, getting the worst quality. Other user in a better connection can receive 2 layers, improving the quality, and a third user in a good network receives the three layers, presenting a superior quality.

The advantage of Scalable Video Coding is that the MCU (or other central element) does not need to transcode video anymore, as the video encoding is performed by the end users, so, it need only to redirect network packets, minimizing its processing demands and the network delay.
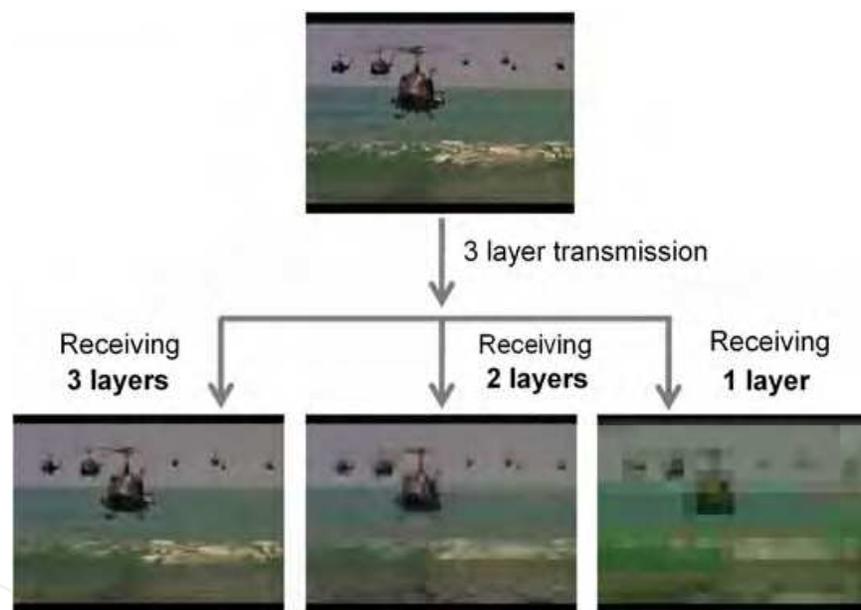
Fig. 6. Scalable video playback

1.3 Requirements of videoconference systems

Related to the features expected in a videoconference system, the following items list some metrics to compare different systems according to the user objective.

- Meeting recording for asynchronous access through video on demand (VOD)
- Whiteboard
- Application and Desktop sharing, as well as documents presentation
- Public and Private chat
- SIP or H.323 standards support for interoperability
- LMS support / integration to facilitate exchange of files and definition of agenda

- Conference scheduling and participants invitation
- Notes for registering the meeting agenda, for example
- Polls in order to vote for some issue in debate
- Moderation, authorization, access rights
- Federated networks support allowing the user to use the same password in different applications, including videoconference
- Exchange of files
- Administrative users with knowledge and power to change remote settings, to help someone who do not know how to set its microphone, for example
- Feedback about the network conditions in order to take some measure if the bandwidth is too low;
- Scalability to hundreds or thousands of simultaneous users in different virtual rooms
- Cryptography
- Mobile support

The focus of this chapter is in a web conference system called Mconf (Multiconference system for interoperable access between web and mobile devices), composed of three main parts: web portal; web conference server; web conference client in the browser / mobile devices.

This chapter aims to describe in a simple way some technical issues and user possibilities regarding the Mconf project. As the project is open source, it is expected to bring some information and motivation to its use.

The following sections will present these three parts in more detail. Section 2 presents the Mconf system in a general way. Section 3 focuses on BigBlueButton, the web conference system integrated with Mconf. Section 4 presents the web portal. Section 5 presents the Mconf mobile application, while Section 6 brings some conclusions and future directions.

2. Mconf overview

This session presents an overview of the Mconf project and its main building blocks, seen in Fig. 7.

The conferencing component of Mconf is BigBlueButton (BBB), an open source web conferencing system developed for distance education, providing the real-time sharing of voice, video, slides, desktop, and chat, besides recording. While focused on distance education, it has the same features at the core than other commercial web conferencing systems. Since BigBlueButton is open source, the Mconf team has the option to improve the code and contribute those improvements back to the BBB team.

The second block of the Mconf architecture is the web portal (Mconf-Web), a Ruby on Rails application that provides creation of virtual rooms, discussion forums, and event scheduling.

The third block is the Android client, which is compatible to both BBB and Mconf-Web. The Android client allows web conference managing and fully participation through mobile devices.
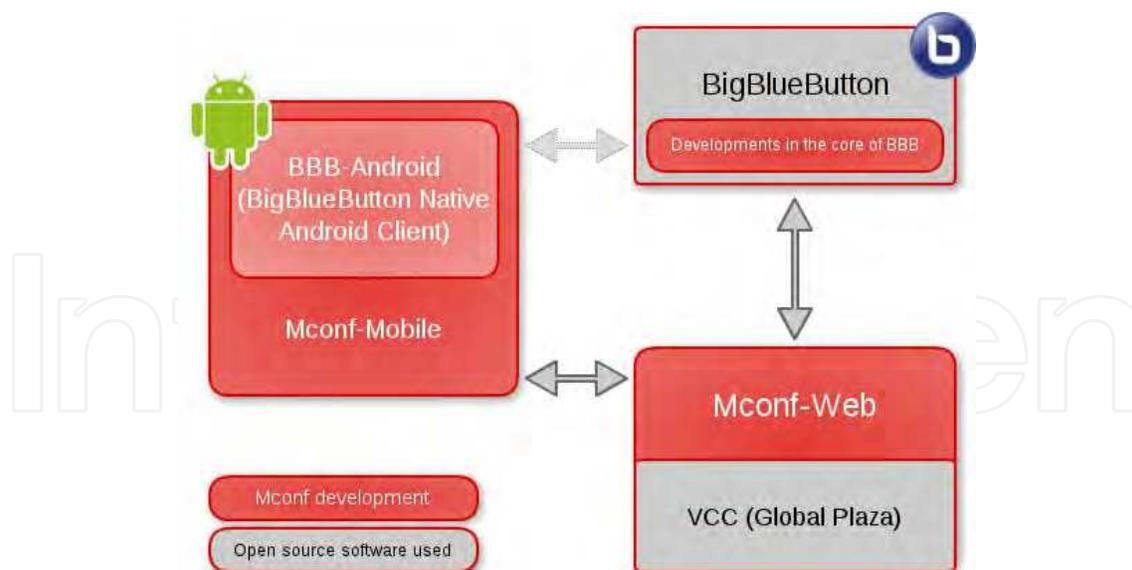
Fig. 7. General view of the Mconf project

Besides that, it is worth mentioning the possibility of federated service and scalability. These issues are described in the following paragraphs.

Mconf-Web has recently been integrated with a Shibboleth[23] module, providing federated login in the website. Shibboleth is a middleware architecture that allows users of different security domains to employ their security information to access a federated service.

Mconf-Web is able to obtain user identity from any identity provider that follows the Shibboleth architecture with few configurations, providing access to any user that is registered in this federation. Currently, Mconf-Web is one of the services provided for the Brazilian federation named CAFe (Federated Academic Community)[24].

Related to scalability, one BigBlueButton server is designed to support about 25 simultaneous users with video and audio, so, a scalability strategy must take place to increase the service capacity. Of course, a powerful hardware could be used to increase the number of users, but it will not scale to 1000 users, for example.

For the audio communication, BigBlueButton uses FreeSWITCH[25], an open source telephony platform that is used as a VoIP server. FreeSWITCH is already scalable, allowing the VoIP service to be spread across multiple FreeSWITCH servers, which would increase a lot the capacity of the web conferencing service since the voice processing is very CPU intensive. However, there isn't yet a way to scale the other components of BigBlueButton through multiple servers.

The solution being developed by Mconf includes a monitor module on BigBlueButton that generates statistics like CPU, memory and network, that are collected by the web portal. When there is a new room to be created, the web portal decides which server is more suitable to the target users, based on statistics plus geographic location and delay.

---

[23] http://shibboleth.net/

[24] http://www.rnp.br/servicos/cafe.html

[25] http://www.freeswitch.org/

Besides that, it is possible to use this solution through a commercial or private cloud computing infrastructure. Virtual machines of web conferencing servers are enabled or disabled depending on the demand in a given moment, all of them managed automatically by the web portal scalability module.

3. BigBlueButton web conferencing

BigBlueButton is an open source web conferencing system for distance education. The BigBlueButton project started in 2007 at Carleton University (Ontario, Canada) as an effort to replace an existing commercial web conferencing system. The project has since evolved to provide the core features of synchronized presentation, chat, video, audio, and desktop sharing. Fig. 8 shows an example of its interface presenting its main windows, which are described in the figure.

The BigBlueButton client runs within the Adobe Flash Player[26]. The BigBlueButton server manages the individual sessions, keeps all users in sync, and records the sessions for later playback.

Fig. 8. Example of a BBB web conference

Users enter a BigBlueButton session in one of two roles: viewer or moderator. A viewer has the capabilities to a) interact with others through chat, b) share their webcam, c) join the audio conference, and d) raise their hand. A moderator has all the capabilities of a user, and in addition can a) mute/unmute users, b) eject a user from the conference, and c) make any user the current presenter.

[26] http://www.adobe.com/products/flashplayer/

The presenter has the ability to a) upload slides to the presentation area, b) control the slide view for all users, c) annotate the slide with the whiteboard, and d) share their desktop.

Compared with commercial web conferencing systems, BigBlueButton lacks some features, such as remote control of desktops; however, the project has stated its goal to focus on simplicity and ease when implementing the core features for web conferencing.

3.1 BBB architecture

BigBlueButton has a client and a server component. The client is written in Action Script. The server components are mostly written in Java. The three main components of the BigBlueButton server are the real-time server, application server, and voice conferencing server. Fig. 9 presents its main building blocks and communications.
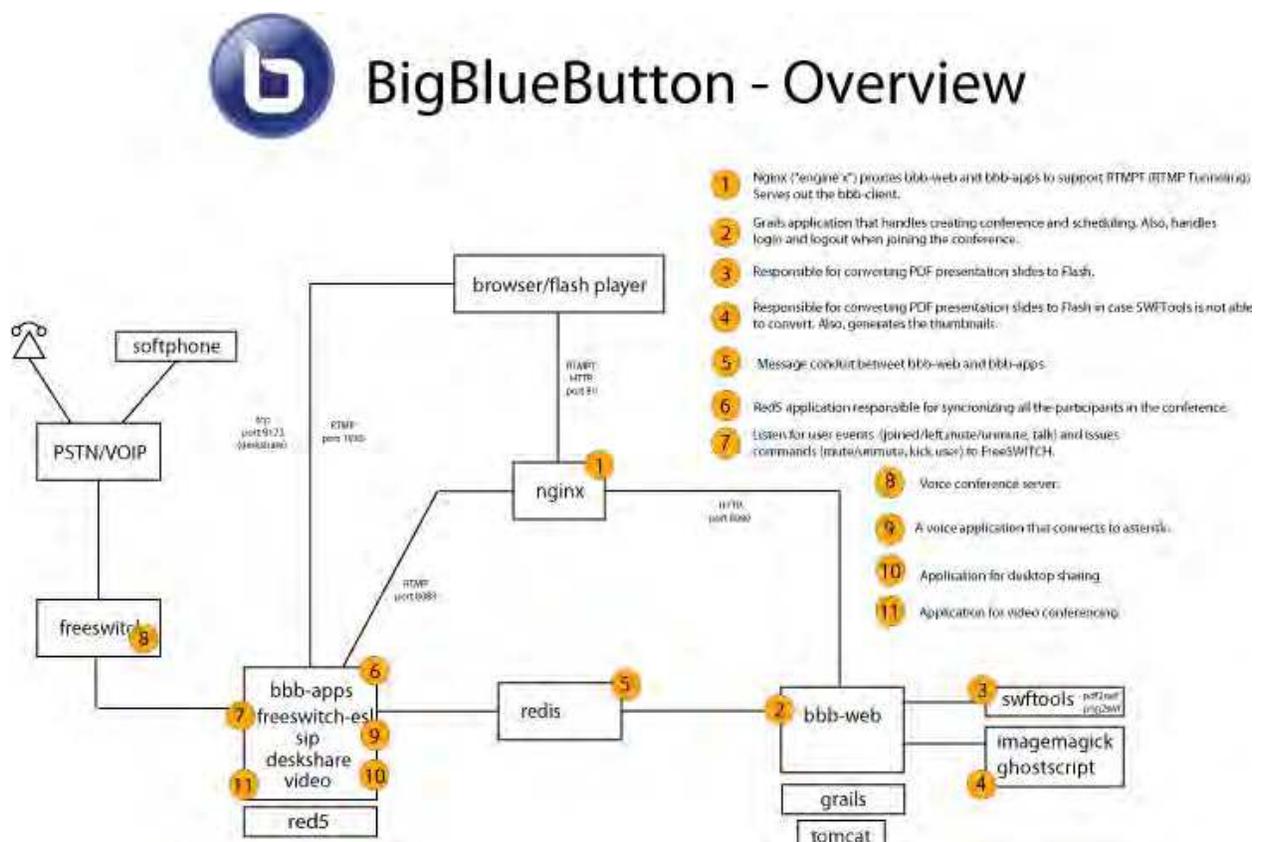
Fig. 9. BBB overview

The real-time server of BigBlueButton is based on red5[27], an open source implementation of Adobe's Flash Media Server[28]. BigBlueButton implements a number of server-side applications, called bbb-apps (see in figure), that keeps the video, audio, and chat in sync between the remote BigBlueButton clients and the application server.

---

[27] http://red5.org/
[28] http://www.adobe.com/products/flashmediaserver/

The application server, called bbb-web (see in figure), is a Java-based application running within an Apache Tomcat[29]. The bbb-web application handles all API calls and requests by the client to upload slides.

The voice conferencing server is built on FreeSWITCH[30], a scalable open source cross-platform telephony platform. Incoming audio packets from the BigBlueButton client are routed to FreeSWITCH through red5phone, an open source voice over IP (VoIP) phone that uses the Session Initiation Protocol (SIP) to communicate with FreeSWITCH.

3.2 Integration of BigBlueButton with other software

The BigBlueButton project does not bundle any built-in web-based applications, such as a system for scheduling sessions; rather, it provides an API that others can use to integrate with 3rd party applications.

BigBlueButton provides an HTTP-based API that includes commands to create, join, and end a meeting. Since the API is HTTP-based, a 3rd party application can control the BigBlueButton server with simple URL requests. Integration with a 3rd party application requires the writing of a plug-in for the 3rd party application (such as Moodle). Many of the most popular open source applications have a plug-in architecture specifically for integration with external systems such as BigBlueButton.

For example, Fig. 10 shows the BigBlueButton plug-in for Moodle, which enables a teacher setup and schedule BigBlueButton sessions from within the Moodle interface[31].

Fig. 10. Integration BBB and Moodle

---

[29] http://tomcat.apache.org/

[30] http://www.freeswitch.org/

[31] http://moodle.org/

Using the BigBlueButton plug-in a teacher can configure properties for the BigBlueButton session. Examples of properties are the name of the link in Moodle, restricting student's access to an on-line session until a moderator (teacher) has joined, or restricting access to specific times within the Moodle calendar.

When the record property is set to true, the BigBlueButton server will record the audio, slides, and chat from a meeting for later playback within an HTML5 compatible browser (such as FireFox and Chrome), as seen in Fig. 11.

Fig. 11. Playback of a recorded lecture within BigBlueButton

4. Mconf web portal

The objective of the Mconf web portal is to provide user authentication, permission control, scheduling, among others. These kinds of tasks are left in BBB to third-party applications, such as the integrations with Moodle and Joomla, for instance. Furthermore, the web portal aims to provide facilities to manage multiple servers and provide load balance, for example.

The most important roles of Mconf-Web are listed below:

- User authentication
- Permission control
- Ability to manage multiple meetings (and meeting rooms)
- CRUD operations (create, read, update, delete) on rooms
- Ability to manage multiple servers
- CRUD operations on servers
- Monitoring interface
- Load balancing
- Conference scheduling
- Usage statistics

- Asynchronous access to the recordings (audio, video, presentations)
- Pre-upload of documents

Mconf-Web can be compared to other integrations already existent for BigBlueButton, except that it has a different focus and interaction model. BigBlueButton has a strong focus on distance education, but Mconf focuses mainly on virtual meetings, not necessarily for education. Different focuses will result in different application requirements, especially for the user interface, in this case the web application Mconf-Web.

Mconf-Web uses a web application called Global Plaza as the basis for Mconf-Web. Global Plaza is an open source web application that aims to enable virtual meetings through an event based social network. Its model fits very well the requirements imposed in Mconf. In the following of this section we will explore each of the points commented above, starting with a description of Global Plaza and then Mconf-Web.

4.1 About global plaza

Global Plaza is a web application developed in Ruby on Rails[32], providing events as social media in the Internet [Quemada, 2010]. Global Plaza was developed in the European Global Project[33] as an open source application and is also available for use in the Internet[34]. The GLOBAL project has ended on October 31, 2010, but as an open source application Global Plaza is still available for use.

Fig. 12. Video inside the TF-Media's space in Global Plaza

---

[32] http://rubyonrails.org/

[33] http://www.global-project.eu/

[34] http://globalplaza.org

In Global Plaza, the registered users can create three types of events: (a) in person, where you just have a date and agenda in the website; (b) virtual meetings, possibly with streaming and recording; and (c) virtual congress, large virtual meetings, usually needing an on-site operator. These events happen inside spaces (or communities), that are a places where people with similar interests can interact. Fig. 12 shows one space inside Global Plaza where a recorded video is being played through the web interface.

The Global Plaza application is just the web interface in a stack of applications used to provide the full set of features. For web conferencing it uses the Isabel system [Quemada, 2005], that also requires several components.

From the feature set needed by Mconf, Global Plaza provides user authentication, permission control, management of meetings (inside events), scheduling, and access to the recording. Some other features, such as usage statistics, can be easily leveraged from the framework used by Global Plaza. It is important to observe that most of these features are tightly coupled with Isabel and had to be adapted for its use with BigBlueButton.

4.2 Mconf web portal details

This section will detail the main features of Mconf web portal. Some features are inherited from Global Plaza, and the differences between Mconf-Web and Global Plaza will be discussed later.

Mconf-Web is a social network of users, spaces and events. Users can create and join spaces (communities) and can create and participate in events. All the events are held inside a community from a defined start date to an end date.

Fig. 13 shows the user home page in Mconf-Web. This page has links to the most used features in the website. The label "Web conference room" shows the access point to the user´s web conference room (to start a meeting and invite people to his meeting, for example). The label "Communities" allows access to the user communities (or spaces). The label "Recent activity" shows messages, events, news and attachments related to the user. The label "Agenda" depicts the section where the user can see the upcoming events, and the label "Inbox" shows the latest internal messages which arrived. These features will be further explained in the following of this section.

A community in Mconf-Web can be public or private. A public community can be viewed by anyone, even by visitors that do not have an account in Mconf yet. But, even though everyone can view it, only authorized people can alter it (create events and post messages, for example). Private communities, however, can only be viewed by authorized people. A private community can be created by a user and the access can be given only to people from the same workplace, for instance. Nobody else is able to see the activity inside this community (even though everybody is able to know that the community exists).

4.2.1 Web conferences

In Mconf-Web there are two types of web conference rooms: users' rooms and communities' rooms. When a user registers in Mconf web portal, he/she automatically receives a permanent web conference room (a user room). This room can be shared with anyone the

user wants to, even with people that do not have a Mconf account. The user rooms are public by default, so anyone with its link will be able to access it. But it can be made private so that only people with the defined password can join the web conference.

Every community created in Mconf also has a permanent web conference room associated (a community room). If the community is public, the room is public, otherwise the room is private and there will be a password associated to it.
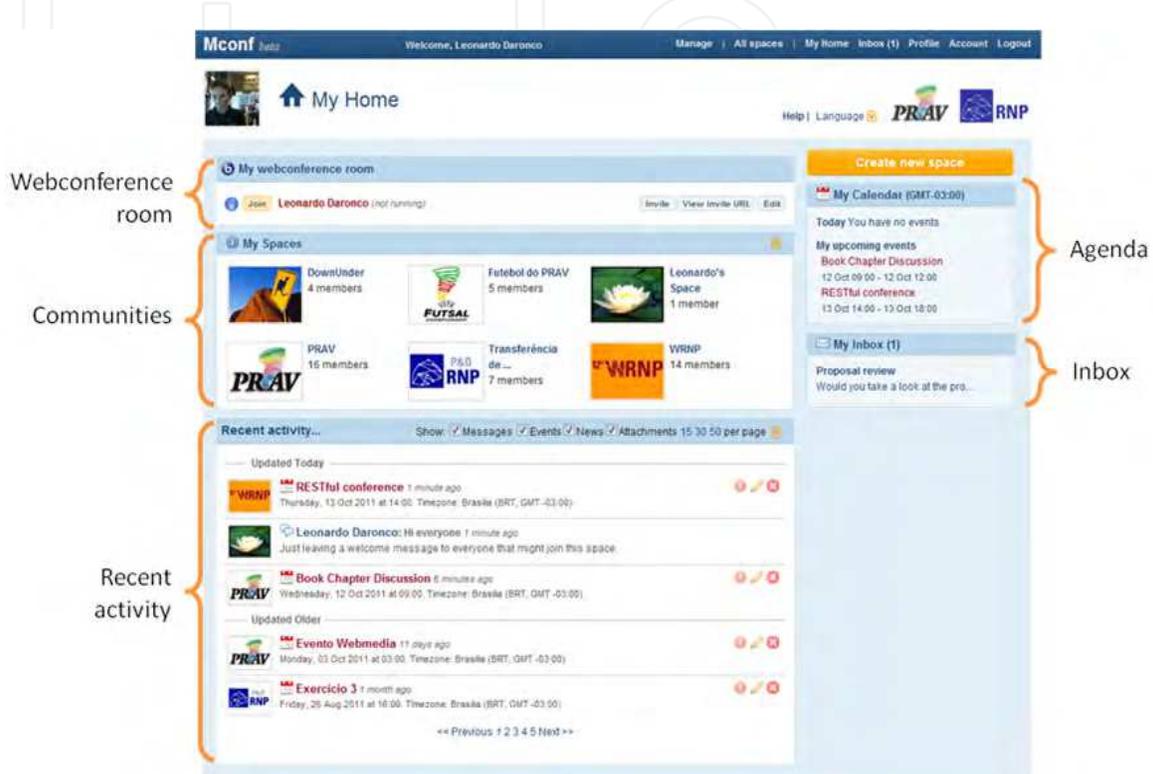

Fig. 13. Main page of the Mconf web portal

The permissions inside a web conference will depend on several factors, but mostly on these:

- Who is the user?
- Is it a user room or a community room?
- Is the room public or private?

The permissions available are the ones already discussed in the BigBlueButton section (section 3): moderator, viewer and presenter. The presenter role is the only one that can be defined during the meeting using the application interface. The other roles are defined before the user joins the conference, and it's Mconf-Web that should decide if the user is a moderator or a viewer.

In a user room, the owner will always be a moderator and the only one that can start the meeting in the room. If the owner is not in the meeting, no one else will be able to join. Everybody else will have the viewer role.

An exception to this rule happens when the room is made private. When the user marks the room as private, he/she will be asked for two passwords: the viewer password and the

moderator password. Any person invited must use one of these passwords to join the conference, and the password will define the role of this person. Fig. 14 exhibits the web page shown when an anonymous user tries to join a private web conference.

In a community room all members who belong to the community are moderators, so any of them can start a meeting in this room. In a public community, everybody else will be able to join and will receive a viewer role (including anonymous users). In a private community, a password will be asked when a non member tries to join the meeting and this password will define the user's role in the conference.

Fig. 14. Anonymous user invited to a private web conference

In Mconf-Web any user can start a web conference with just a few clicks, as shown in Fig. 15. After logging in the website, the user can simply click in the "join" button and the web conference will start. The same is valid for a community: after creating a community, any member can simply click in the "join" button and start the web conference.
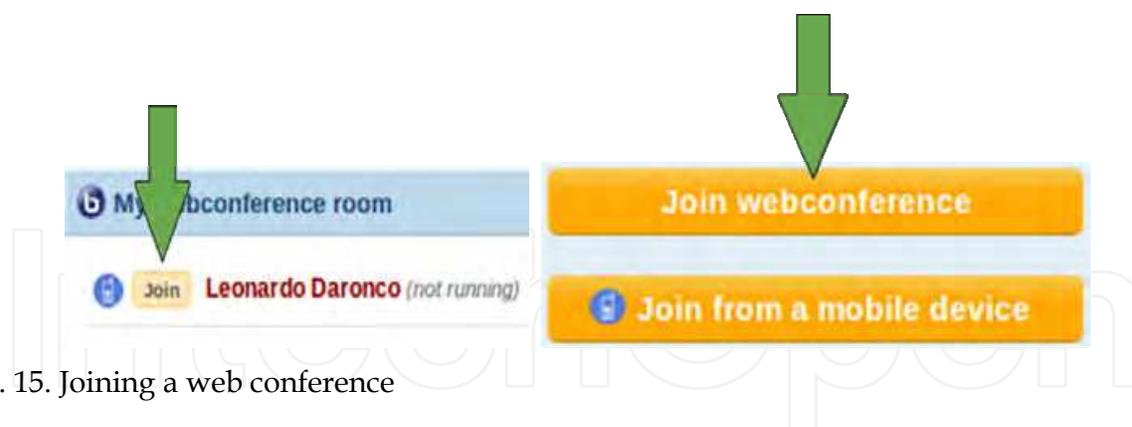
Fig. 15. Joining a web conference

4.2.2 Other features

Mconf-Web has still some other features not specifically related to web conferencing but that add value to the application and can improve the user experience.

One of these features is the document sharing, where users can share any kind of documents inside a community. These documents can be accessed by any member of the community and, if the community is public, they can be accessed by anyone. Documents can also be tagged and versioned, so that it is easier to search for them later on. Fig. 16 shows the documents sharing page.
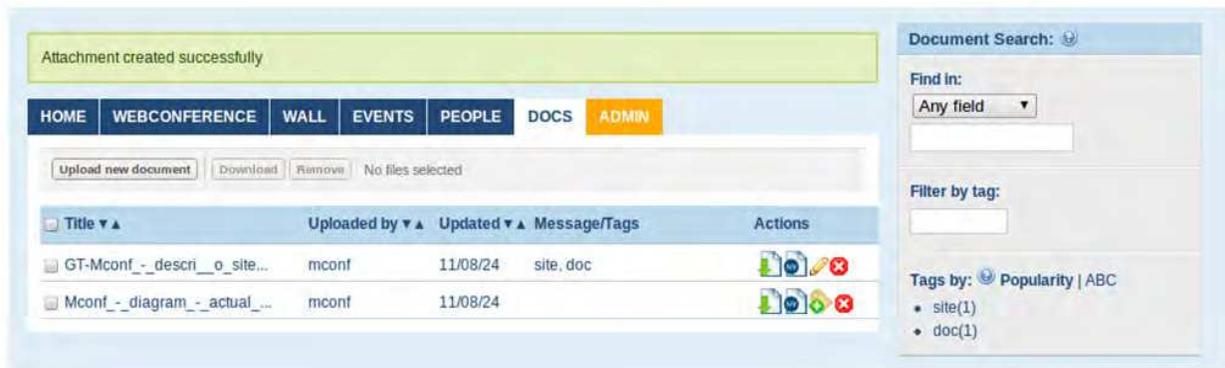
Fig. 16. Documents sharing page

Communities have also a wall board and a notice board, where any kind of text messages can be posted and all members will be notified. In the user's homepage, there's a digest of the recent activity that includes the latest documents shared, messages and news posted, and events created. The user also has a calendar that shows the next events, as illustrated in Fig. 17.
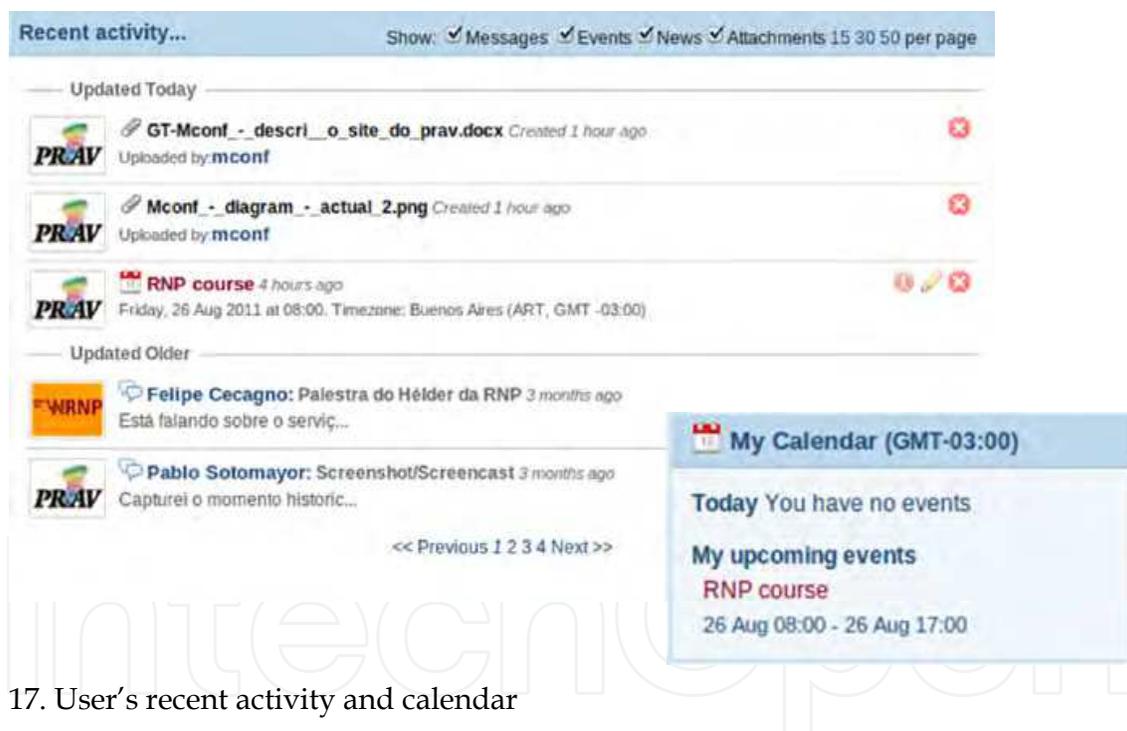
Fig. 17. User's recent activity and calendar

The website has also an internal messaging system, that works similarly to an email system and that users can use to communicate with each other.

Mconf-Web is fully available in English and Portuguese (BR) and partially translated to Spanish, French and German.

4.3 Main differences between Mconf-Web and global plaza

The most important difference between Mconf-Web and Global Plaza is the use of BigBlueButton as a replacement for Isabel. BigBlueButton is very different from Isabel and

follows a different approach to reach the same goal: provide a web conference environment. Mconf uses BigBlueButton due to all its features already detailed in the section 3 and due to its open source nature and great community support and activity.

BigBlueButton has many advantages over Isabel, so this can be considered not only as a different approach but as an evolution in the web application specially for the open source community.

One downside of this replacement is that some features in Global Plaza had to be disabled before they were adapted to be used with BigBlueButton. This is the case of the streaming and recording features. BigBlueButton has only recently included recording in its core and this still needs to be integrated in the web application in the near future.

As already mentioned, the project supporting Global Plaza has ended in the end of 2010, so it's development has been greatly reduced since then. It is originally developed using the version 2.3.4 of Ruby on Rails, an old version that was since 2010 replaced by the version 3. So other of the most important differences included by Mconf-web was the migration to Ruby version 3.

The migration for Ruby on Rails 3 was especially important for the libraries developed to integrate the web application with BigBlueButton. These libraries will be detailed in the following sections, but it is important to consider that they were developed in the latest version of the framework so that they can be used with any other web application that uses the same framework.

The way a user can create and participate in web conferences has also been vastly modified. Global Plaza is a very event-centered application, and users need to create an event to be able to hold a web conference. Everything revolves around events.

In Mconf-Web, web conferences are easier to be held. The user can create a web conference from his homepage or from a community very easily, while events are only a mean to schedule a web conference inside a community.

Besides that, several things were also changed in the application's interface in order to make it simpler and easier to be used.

4.4 Implementation details

The most important thing that should be added regarding the implementation is that all the integration between the framework Rails and BigBlueButton was developed in two libraries (called "gems" in Ruby). With these libraries, the front end application (in this case Global Plaza) can be replaced by any other Rails application and all the integration with BigBlueButton would still be functional. In other words, it is now easy to start a new front end application and integrate it with BigBlueButton.

Fig. 18 shows how the application, the gems and BigBlueButton interact with each other. The gems were built in two levels: (1) a basic library for Ruby that simply performs HTTP calls to contact BigBlueButton's API; and (2) a library for Rails that is targeted for web applications and contains common data models and controlling logic to interact with BigBlueButton.
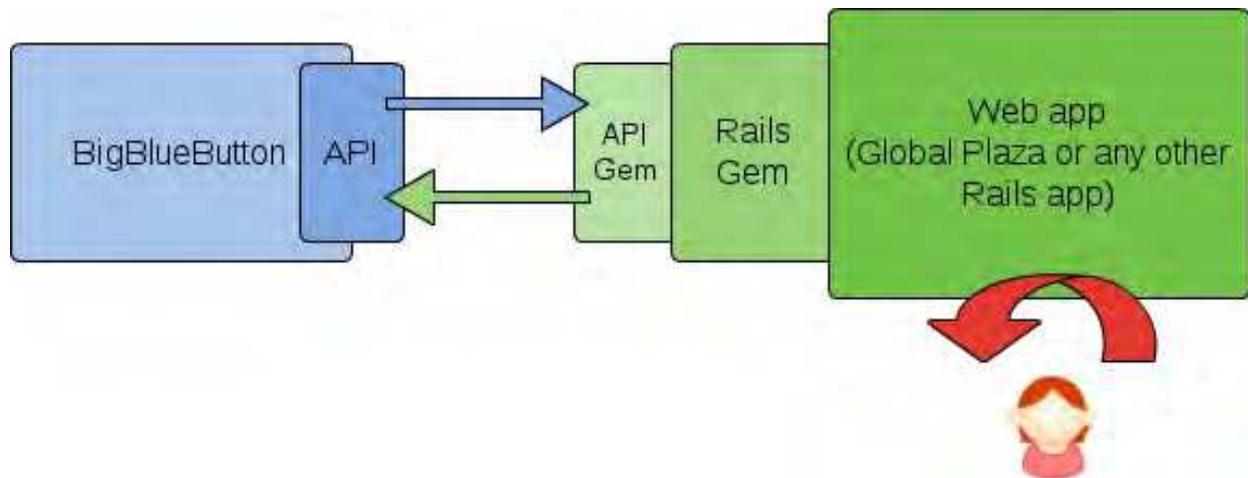
Fig. 18. A front end Rails application and the gems to interact with BigBlueButton

The basic gem (1) is called "bigbluebutton-api-ruby"[35] and can be used by any Ruby application (it is not necessary to be a Rails application). It contains methods to perform every API call possible in BigBlueButton and utility functions to adapt the response (XML files) to be friendlier to a Ruby application.

The other gem is called "bigbluebutton_rails"[36] and is specific for Rails. It implements several features that a web application needs to use BigBlueButton. The most important features are:

- Provides data models to manipulate BigBlueButton servers and conference rooms;
- Allows multiple servers and multiple conference rooms;
- Full API access using bigbluebutton-api-ruby;
- Easy way to join conferences: simply create a room and call a defined URL;
- Possibility to create private rooms, that require a password to join;
- Deals with visitors (users that are not logged), allowing (or forbidding) them to join rooms;
- Server activity monitor;
- Includes default views (web pages) used in the interaction with BigBlueButton (create rooms and servers, join a room, invite people to join, and others).

5. Mobile devices integration in Mconf

This session will discuss the development details of the Mconf Android client. There are two possibilities of interaction using Android devices, shown in Fig. 19 and Fig. 20: the first one is through a web browser, and the system works like in a regular PC, using Adobe Flash Player (shown in Fig. 19a); the other way is through the Android native application (shown in Fig. 19b and Fig. 20). There are two Android native applications, named Mconf-mobile and BBB-Android, which are described in section 5.2.

---

[35] https://github.com/mconf/bigbluebutton-api-ruby
[36] https://github.com/mconf/bigbluebutton_rails

Fig. 19. Examples of web conference in a tablet using: a) flash or b) the Android application
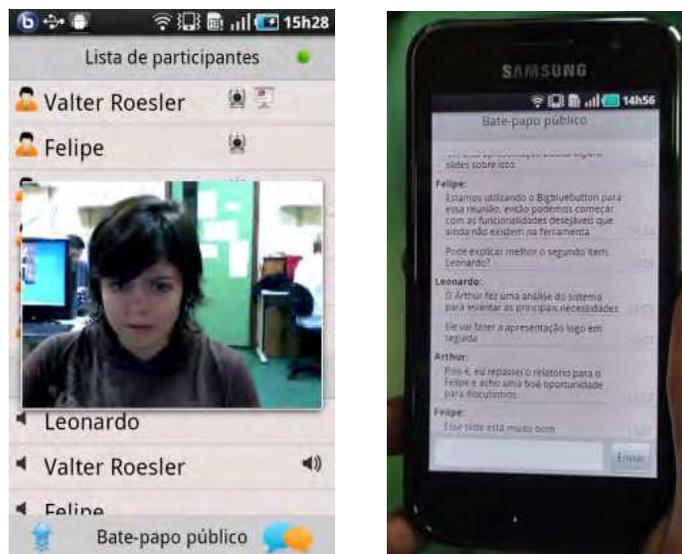
Fig. 20. Examples of web conference in a smartphone using the Android application

The Android operational system was chosen because of its big growth and popularity, in addition to a non-bureaucratic development and easy distribution for the applications.

The BigBlueButton web application used on PCs also runs on Android 2.2 and above, since these Android versions support the Adobe Flash Platform. However, there are several usability limitations when adapting the user interface to a (often small) touch screen display. It would be an option to develop a new flash based client adapting the user interface to mobile devices, but since audio and video capture is still not supported by Adobe Flash Player on Android, it would be impossible to develop a client with all the desired features, so, the group in Mconf two Android native applications were developed (see section 5.2).

Most features are the same in the web-based application and the Android application, however, a few are still in development on the native application, such as the presentation, the whiteboard and the possibility to display several participant videos in the same screen. These capabilities will be gradually added in order to cover all the features existent on the web application, with the advantage of having audio and video capture since the beginning.

Another advantage of the adopted approach is that the native application can be distributed in the official Android marketplace, what is not possible with a web application. The native Android application developed is compatible with Android 2.0 and above, and the details will be presented hereafter.

5.1 Architecture

The Mconf for Android is built upon many open source applications and libraries. Also, the support for mobile devices doesn't require any modification on the server side. The solution architecture is presented in Fig. 21. The figure depicts the software layers of both mobile applications developed (BBB-Android and Mconf-Mobile – see section 5.2), and each block is detailed hereafter.
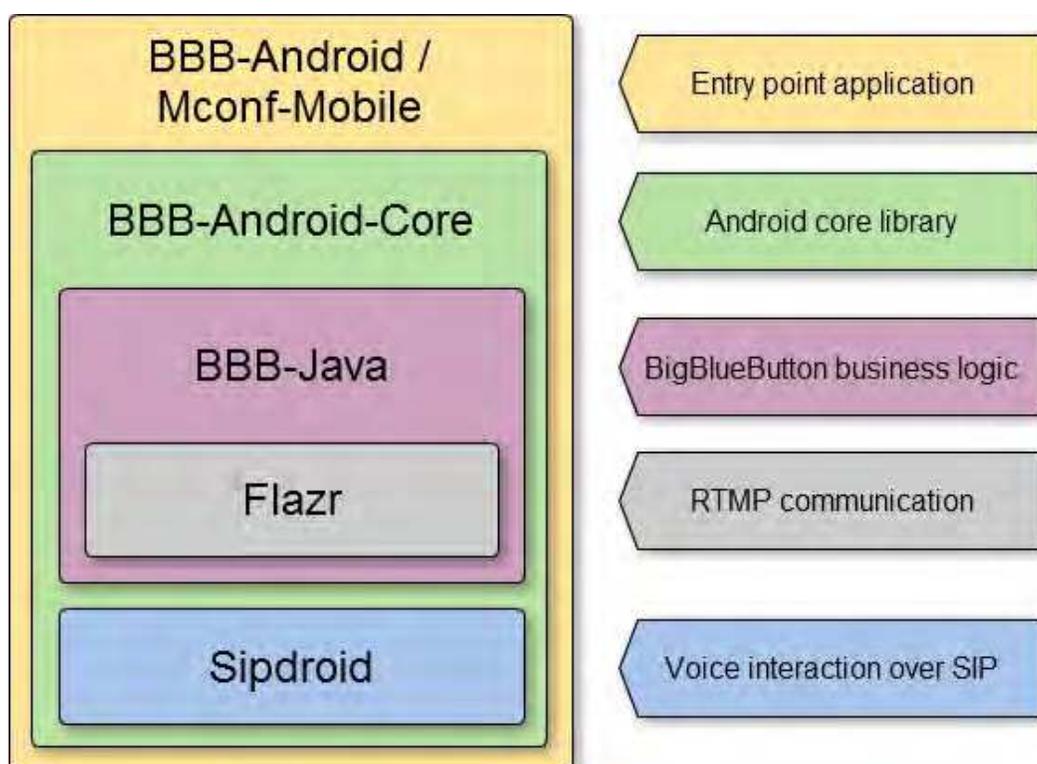
Fig. 21. Mconf mobile architecture

5.1.1 Sipdroid[37]

Considering that the BigBlueButton system uses a VoIP server to handle the audio streams, the strategy of integrating an existing VoIP open source solution to the application was adopted. The Sipdroid is one of the most popular SIP phones on Android Market, and the access to the source code has enabled the Sipdroid to be transformed in a library, which could be consumed by another Android application or library.

---

[37] http://sipdroid.org

Speex[38], the default audio codec of BigBlueButton, is among the audio codecs supported by Sipdroid. It is implemented in C and compiled through the Android NDK.

This integration of Sipdroid with Mconf has enabled the addition of the audio interaction feature with a very low development cost.

5.1.2 Flazr[39]

All the RTMP (Real Time Messaging Protocol) [Adobe, 2009] communication between the application and the Red5[40] server is done through the Flazr library, a Java open source implementation of protocols used for multimedia streaming. This library implements important parts of the RTMP protocol, such as the initial handshake and handlers for messages of control, audio, video, and also Remote Procedure Calls.

However, the library does not support Remote Shared Objects (RSO), an indispensable requirement for the complete communication with the server. For example, information such as join and leave of participants, video transmitting or the "raise hand" features are actions broadcasted through the *participantsSO* object. Public chat message exchanges are broadcasted through the *chatSO* object.

Then, an extension of the library was developed to add support for RSO. The implementation of the shared objects management and multiplexing/demultiplexing of messages were based on the Red5 source code and the RTMP protocol specification.

5.1.3 BBB-Java

The BBB-Java is a library developed in the Mconf project to be used by the Android client. This library implements the communication logic between a generic Java application and a BigBlueButton server. The library is loosely coupled with the Android application, in order to enable and encourage the development of new third party applications integrated to the web conference system. Also, the library can be used to create robot applications, which would help to perform stress and behavior tests.

5.1.4 BBB-Android-Core

BBB-Android-Core is another developed library which implements the session handle after the user joins a meeting, and it uses BBB-Java to interact with the BigBlueButton server. This library is written mainly in Java, using the Android Software Development Kit (SDK), and a small fraction of CPU intensive code was developed in C/C++ over the Native Development Kit (NDK).

An optimized compilation of the FFmpeg[41] library was used to handle video encoding and decoding, and it was compiled only with the H.263 codec enabled (the default video codec of BigBlueButton). FFmpeg is a library written in C which implements many audio and video codecs.

---

[38] http://www.speex.org
[39] http://flazr.com/
[40] http://red5.org/
[41] http://ffmpeg.org

5.2 Mobile applications

In Mconf two web conference client applications for mobile devices were developed. One is called *Mconf-Mobile*, and the other is *BBB-Android*. Both are depicted in Fig. 22.

Mconf-Mobile is an application integrated with the Mconf web portal. It implements the login screen of the session, and using it the user authenticates and looks for the meetings he has permission to join. After joining a meeting, the session is handled by the *BBB-Android-Core* library.

BBB-Android is an application for direct use with BigBlueButton. Through it, the user can join, as participant or moderator, any running meetings, and also create new meetings.

(a)   (b)

Fig. 22. Mobile applications: (a) Mconf-Mobile; (b) BBB-Android

The main difference between *BBB-Android* and *Mconf-Mobile* is that *BBB-Android* connects directly to a BigBlueButton instance, without any interaction with the web portal. Because of that, it is necessary to know previously the server address, differently from *Mconf-Mobile*, where the user has to know simply the web portal URL.

5.3 Main functionalities

This section presents the main functionalities of the Android application.

5.3.1 Chat

Just like the BigBlueButton web client, the Android application provides chat interaction through public and private message exchange. The public chat is implemented as a sliding bar on the main screen, and the private chat is accessed pressing on the participant name with whom the user wants to start a conversation.

5.3.2 Audio

When the user joins the voice conference, two buttons are presented on the main screen in order to enable the user to interact: the buttons *Lock speak* and *Push to talk*, as seen in Fig. 23. *Lock speak* means that the microphone is always open, and *push to talk* means that the microphone is only open when the user is pressing this button. They are essential for a good quality interaction, because the high microphone sensitivity on mobile devices can result in noise and acoustic echo problems.
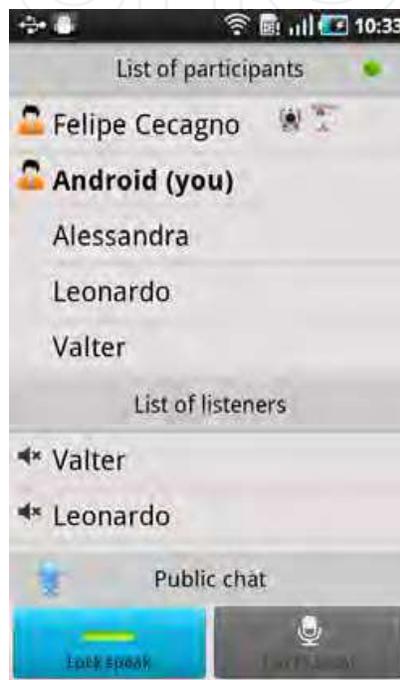
Fig. 23. Voice interaction

5.3.3 Video

When a participant is publishing his video to the session, a webcam icon appears in the main screen, as seen for "Felipe Cecagno" in Fig. 24. The Android application enables the user to playback any participant's video being published, as well as capturing and publishing its own video camera. If the user puts the mobile in horizontal mode, it shows the video in full screen.

If a user joins a session with the moderator permission, he/she will be able to perform some management actions, such as turning any participant into presenter, muting or unmuting a noisy participant, lowering hand of a participant that wants to talk or even kicking out a participant from the session.

These kind of features suggest a new use case for the application: a professor using the white board to explain the lesson for a group of students connected remotely doesn't need to stay in front of his computer during the lecture, but can use the mobile device as a remote control to manage the conference.

6. Conclusion and future directions

This article presented two main contributions: a) one survey of videoconference systems and trends in the area; b) technical and implementation details about the Mconf project, which involves a complete web conference solution, contemplating the triptych web portal / web conference environment / mobile devices.

Initially, the survey presented different deployments of videoconference systems and some trends, as scalable video, high definition and usability.
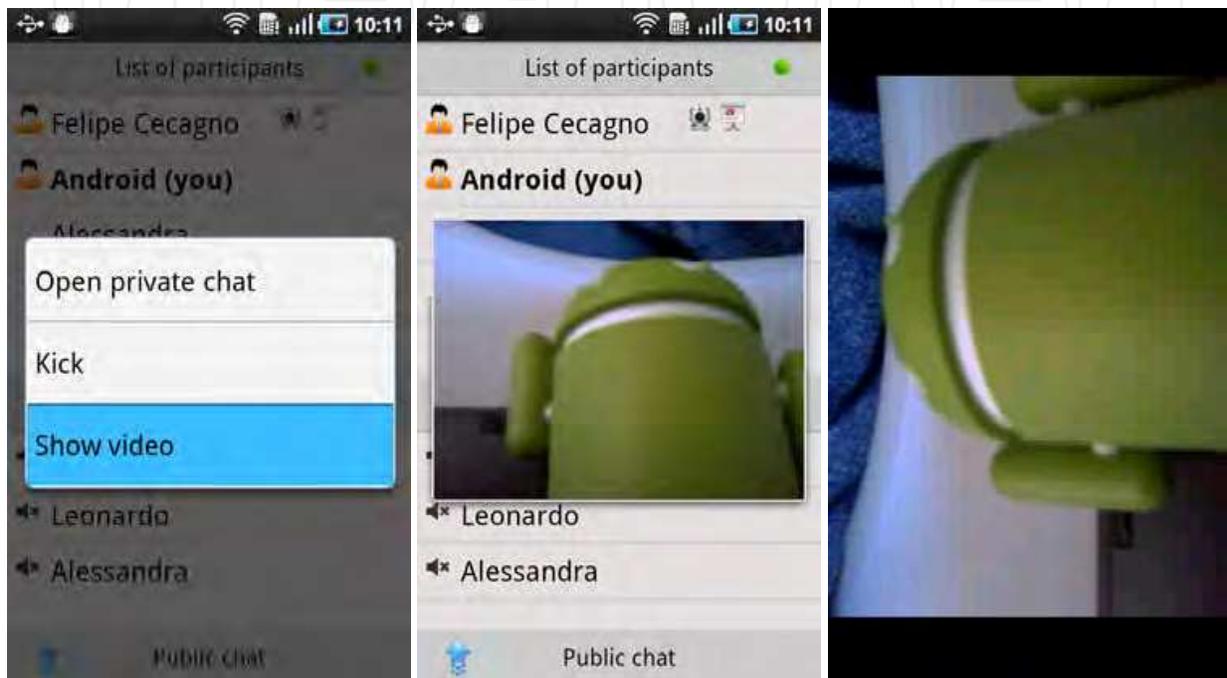
Fig. 24. Video interaction allows playback and also sending the mobile video

After that, the chapter focused on the Mconf system, which provides an open source web conference solution based on BigBlueButton, Global Plaza and some own developments, like the mobile solution.

Mconf provides a tool with good quality, scalability, simplicity, mobile access and recording. Scalability is provided by balancing the various web conference rooms among different servers, monitoring constantly the servers' activity to better choose the best one to start the new room. This is the principle of cloud computing, and it is very suitable to this application. One issue is the cloud delay, and the server must be chosen taking also the delay into consideration. Security is provided through the use of Federated login and data encryption. Federated login is a reality nowadays, simplifying greatly the permission handling.

There is a long road ahead, however, and the main future developments are:

- To create an HTML5 client, in order to have web conference in other mobile platforms such as iOS and Windows Phone;
- To simplify the web portal, allowing the user to know who is available in each community and create easily a web conference with them;

- To allow interaction with room systems like Polycom, using H.323 or SIP.

All the source code of the Mconf project is available under open source licenses in https://github.com/mconf. The main links are:

- Mconf version of BigBlueButton: https://github.com/mconf/bigbluebutton
- Mconf-Mobile (and BBB-Android): https://github.com/mconf/mconf-mobile
- Mconf-Web: https://github.com/mconf/mconf-web

The project information and updates are available in the Google Code wiki at http://code.google.com/p/mconf/wiki/Home. The main subjects are related to the Mconf installation, Mconf-mobile, Mconf-Web and the customized BigBlueButton. Also there is a virtual machine with the entire environment already installed.

Furthermore, there is a demo server available at http://mconf.org. This demo server has about 100 communities and 250 users worldwide.

7. Acknowledgment

8. References

Adobe Systems Incorporated. Real Time Messaging Protocol Chunk Stream. At: http://wwwimages.adobe.com/www.adobe.com/content/dam/Adobe/en/devn et/rtmp/pdf/rtmp_specification_1.0.pdf. June, 2009. Accessed in Sept, 2011.

Huang H-S, Peng W-H and Chiang T. Advances in the Scalable Amendment of H.264/AVC. Proc. of. Advances Visual Cont. Analysis and Adaptation for Multimedia Communication, jan. 2007. pp 68-76.

ITU-T H.323 v7 H Series: Audiovisual and Multimedia Systems: Infrastructure of audiovisual services - Systems and terminal equipment for audiovisual services. Dec, 2009.

Quemada, Juan. Presentation at the 2nd TF-Media Task Force meeting, May 30, 2010. At: http://www.terena.org/activities/media/meeting2/slides/20100530-tnc-juan.pdf

Quemada, Juan; et al. Isabel: an application for real time collaboration with a flexible floor control. International Conference on Collaborative Computing: Networking, Applications and Worksharing, IEEE Computer Society (2005).

Roesler, Valter, Coelho, Luiz. Distributed Classes: Convergence of distance learning and presence learning through a videoconference system. In TNC 2010: Terena Networking Conference 2010. Vilnius, Lithuania. May, 2010.

Roesler, Valter; Husemann, Ronaldo; Costa, Carlos H. A new multimedia synchronous distance learning system: The IVA study case. In Proceedings of the 24th Annual ACM Symposium on Applied Computing, SAC2009. Honolulu, Hawaii. March, 2009.

Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A., Peterson, J., Sparks, R., Handley,
     M., and Schooler, E. SIP: Session Initiation Protocol. IETF: Internet Engineering
     Task Force. RFC 3261, Jun, 2002.
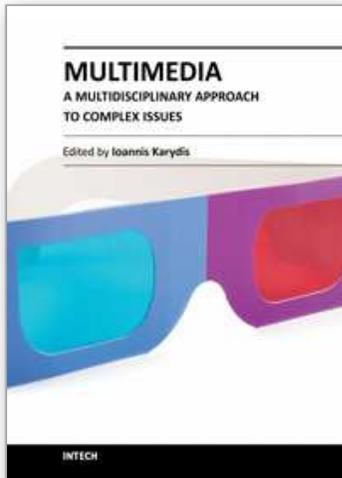
Multimedia - A Multidisciplinary Approach to Complex Issues
Edited by Dr. Ioannis Karydis

The nowadays ubiquitous and effortless digital data capture and processing capabilities offered by the majority of devices, lead to an unprecedented penetration of multimedia content in our everyday life. To make the most of this phenomenon, the rapidly increasing volume and usage of digitised content requires constant re-evaluation and adaptation of multimedia methodologies, in order to meet the relentless change of requirements from both the user and system perspectives. Advances in Multimedia provides readers with an overview of the ever-growing field of multimedia by bringing together various research studies and surveys from different subfields that point out such important aspects. Some of the main topics that this book deals with include: multimedia management in peer-to-peer structures & wireless networks, security characteristics in multimedia, semantic gap bridging for multimedia content and novel multimedia applications.

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Valter Roesler, Felipe Cecagno, Leonardo Crauss Daronco and Fred Dixon (2012). Mconf: An Open Source Multiconference System for Web and Mobile Devices, Multimedia - A Multidisciplinary Approach to Complex Issues, Dr. Ioannis Karydis (Ed.), ISBN: 978-953-51-0216-8, InTech, Available from: http://www.intechopen.com/books/multimedia-a-multidisciplinary-approach-to-complex-issues/mconf-an-open-source-multiconference-system-for-web-and-mobile-devices

INTECH
open science | open minds