Some solutions for an interpreter enabled multimedia conferencing system

Ferenc Sárközy
Computer Integrated Manufacturing Laboratory
Computer and Automation Research Institute of the Hungarian Academy of Sciences
Kende u. 13-17, H-1111 Budapest, Hungary
sarkozy@sztaki.hu

Abstract – As videoconferencing will be an every day used tool in the near future, the demand can arise to implement an interpreter enabled videoconferencing system. Despite of the fact that the technology nowadays is mature enough to accomplish this, there is not any available product on this field. In this paper we will introduce the Global Conference Network (GCN) which is a distributed, multi-lingual, multimedia conferencing system. GCN was designed to have a modularized architecture, in order to enable us to develop and use different multimedia streaming and processing modules applying different approaches to the media handling problem. These modules were based on different open source tools and in this article we will present the investigation of their applicability on this specific field.

A. INTRODUCTION

Video conferencing is getting more and more widely used nowadays. The expensive curiosity of executive officers reached the desktop, literary everybody with a mainstream computer and a broadband (xDSL, cable) internet connection has possibility to try this kind of communication. This achievement has three important components. The rapid development of hardware performance/price ratio provides a solid and cheap foundation to build these systems. This was followed by the steady development of standards and their compliant applications to utilize the existing infrastructure. The third factor, the relatively cheap so called broadband connections, were missing till the few past years. Despite that the service parameters of the affordable telecommunication possibilities were progressed remarkably, still they bottleneck some of the real professional use of video conferencing systems. The recent stall of this progress has mostly business caused reasons, therefore the push of the market make their further development probable as it can be seen for example in Japan. The enhancement of the local loop bandwidth and other QoS parameters can open up the way for such demanding applications as interpreted video conference (IVC).

The additional requirements of an IVC are twofold. First the system must meet the demand of the ISO 2603 standard. Considering the voice signal, its frequency characteristic demand challenges the capabilities of the generally used voice communication methods. It can be said that the interpreters require the highest available quality of sound, image and the synchronization of them. The other problem is that the delays of the interpreted audio signals can two times or – in the case of indirect interpreting – even three times more than in a usual video conference. These difficulties can justify that there are no available solution for video conferencing with simultaneous interpretation accepted by the interpreter community.

In this article we will introduce the design and operation of an interpreted video conferencing system, the Global Conference Network. First of all it was built to function as a production system. It was obvious that the widely affordable possibilities of video conferencing have serious trade-offs. In order to choose the best of them GCN also served as a framework to test different kind of media handling solutions. These are based on open source software and solve the media handling problem with different methods. The introduction is followed by the design of the whole GCN system, then comes the description of the different media handling plug-ins and the third part contains the conclusion.

A. *Goals*

The political, economical interdependence of European small nations and nationalities, maintenance of the linguistic variegation of Europe demand faster and more accurate communication technologies from the coming generations. Since the appearance of both the natural and artificial global language lag behind the forecasts, and also violates national identities, the demand for the development of an interpersonal communication technology bridging the multi-lingual gap can be felt, which can evolve a break-through. The GCN system attempts to establish audio and video connection independently from language and actual place of residence. In order to achieve this general goal, GCN has to meet the following requirements:

- It has to use IP networks as communication medium – it is the necessary facility to use GCN on the whole scale of networks from LAN to WAN.
- It should integrate interpreters in the conference – this is the fundamental difference between GCN and the currently available video conferencing solutions.
- It should be able to take over the role of the existing local conference systems as the software of standard hardware elements.
- It has to integrate any kind of participant through an appropriate internet connection – this means that in an extreme case, all of the participants (chairmen, speakers, interpreters and observers) can take part in the conference from a remote location.
- It should integrate local network based conferences and individual participants through internet connection – this case is the most probable application of GCN. It includes the situation when a remote interpreter connected to a local conference.

- It must be scalable – in the optimal case the solution should work equally well either it is used by a dozen or several hundred clients.

Usual conference related functions are also needed:
- Conference set up, administration, registration – these functions are also required to be implemented on the internet as web applications.
- Centrally managed data store
- Intelligent on-line conference organization – this minimizes the necessary interventions needed from the organizer and the actual chairman of the conference
- Virtual voting machine
- Event recorder – this archives all of the user and system initiated events during the conference, like logins, votes and speaker changes. After the conference these data are presented on a website and can be analyzed and queried.
- Archiver – this component archives the media streams of the conference.

The previously enumerated requirements implicate a centrally managed system. Beside that, the usability can be enormously improved by providing peer to peer interaction possibilities for the participants. These can include:
- Ad hoc created “sub conference” – small groups should be able to communicate during the conference independently from others. The global conference is usually a democratic one with a chairman in charge who gives and takes the right to speak. The sub conferences can use the together in a room like method.
- File sharing

B. Applications and advantages

An IVC system has all the application fields as a normal video conference system. The provided additional benefit is that it allows the communication between parties who have no common language. Important effects of this are the followings:
- Generate information flow that otherwise could be hardly possible
- Widen the applicability of video conferencing itself.
- Integrating remote participants - especially interpreters - into local conferences can reduce the cost of these conferences. .
- Using it in e-learning the interpretation can tremendously increase the size of the audience.

C. Standards

The most important standardized protocols used by the internet based multimedia conferencing applications are the H.323 and the SIP (Session Initiation Protocol). H.323 is an ITU (International Telecommunication Union) umbrella standard for multimedia communication on packet based networks which may not provide a guaranteed Quality of Service.

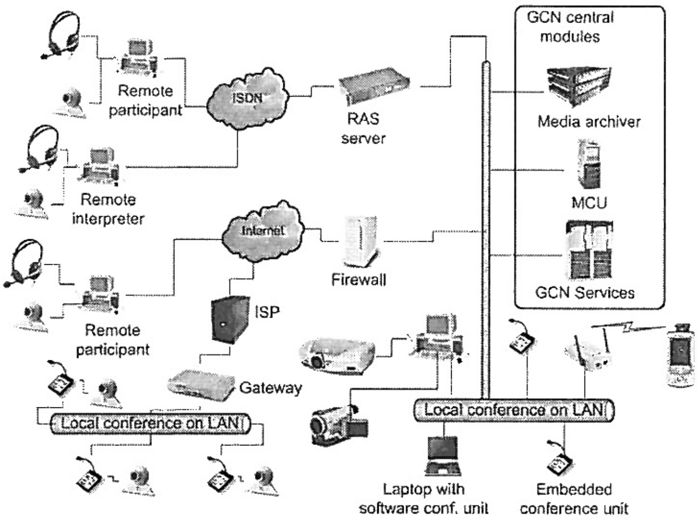

Fig. 1. A typical configuration of the GCN system

H.323 defines a protocol set which has call signalling, call control and media stream protocols. It also defines the following types of entities.
- Terminal is a computer with multimedia application implementing the H.323 protocol stack, or a dedicated hardware solution.
- Gateway is used to translate protocols.
- Gatekeepers are used to manage larger H.323 networks. They are used for admission control, name resolution, bandwidth management.
- Multipoint Control Unit (MCU) makes available the interconnection of more than two terminals in a session. It keeps contact with the terminals individually, and transmits the required stream (transcoded if needed) to them. There are also hardware and software based MCUs.

SIP was created by Internet Engineering Task Force (IETF) and is fully defined in RFC 3261 [1]. It is a lightweight signalling protocol for internet telephony and conferencing. It is based on HTTP and SNMP. SIP is not a vertically integrated communication system like H.323. During the session initiation the communicating parties can rely on Session Description Protocol to negotiate their capabilities and use that media streaming protocols and codec they agreed upon.

The generally used media transport protocol for transmissions of audio and video on packet based networks is RTP (Real-time Transport Protocol, RFC 3550 [2]). It provides end-to end service based on application level framing. RTP offers timing recovery, loss detection, payload and source identification, reception quality feedback, media synchronization, and membership management. It is not a reliable service, but gives that information upon the particular application can decide how to deal with network caused errors. ITU adopted it in as the media transmission protocol in H.323 and applications based on SIP signalling also uses it extensively.

II. SOLUTION

GCN is an internet based distributed system. It has two important roles. During the development, it should be used as a testbed, for different media handling solutions. Later it should work as a production system with the chosen media plug-in. These conditions implicate that it must be modularized with clean and narrow interfaces, robust and scalable. These necessary properties were the guidelines in the architectural design of the system.

A. Components

GCN system has three main functional parts. GCN database (GCNDB) contains all of the information about the conferences organized with the help of that given GCN system. This is an XML database and it is maintained by the GCN Server (GCNS).

GCNS comprise several Java servlet based web applications. It provides web pages to handle whole lifecycle of the conference data, but the emphasis is on the before and after conference functionality. The conference structure is created during the set up process. This includes creating a new conference entity and uploading its specific data, registering participants and assigning roles to them, creating a preliminary schedule and uploading the required distributable materials. During the conference the GCNS helps to review what has happened till that point and acts as a web server which stores the aforementioned background materials. After the conference, the GCNS can be used as an archive viewer. The GCN database contains all the user and system initiated events of the conference and it can be queried and analyzed with the help of GCNS. The structured presentation of the archived media streams is also the task of GCNS.

The third part of the GCN system - the GCN Conference Organizer - (CO) is an extendable set of applications which interact with each other with the floor control messages of GCN. The base set of the applications of CO and their role are the following:

- Conference state manager stores the conference actual state and can be queried by CO application to initialize themselves after login or reconnect.
- Client applications for the different kind of participant roles, like organizer, chairman, interpreter, speaker and observer.
- Admission control authenticates and authorizes the users.
- Vote control manages the voting sessions.
- Media archiver stores the media streams.
- Message archiver captures all the messages of floor control protocol and stores them in GCNDB. From the elementary events represented by these messages the whole conference can be reconstructed, and queried through GCNS.
- Interfaces to external media handling applications. Media streaming, receiving and capturing is solved by external applications. These are integrated in the GCN system with a general interface for the sake of modularity.

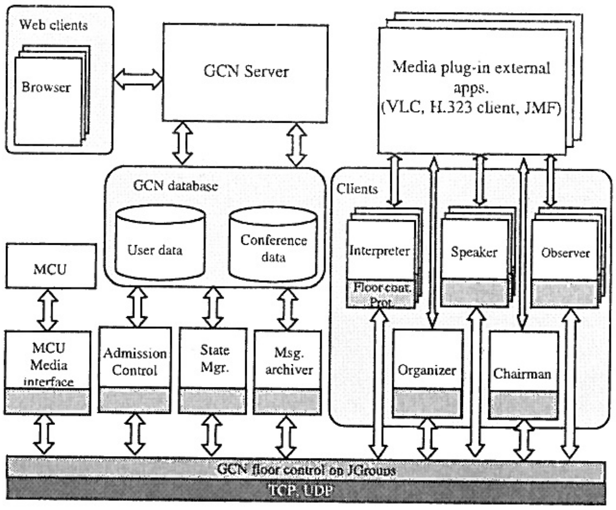

Fig. 2. Relation of the applications of GCN

We wanted to create a portable solution so all of these components are realized on Java platform. This was especially important in the case of the client applications, because Java gives possibility to use user and maintenance friendly distribution methods like applets and Java Web Start. Our decision was strengthen with the hope that the Java Media Framework would be the foundation of the media handling system of GCN. This proved later as an unrealistic belief. Java gave the possibility to choose out of box solutions for the transport layer of the floor control messages, but later caused inconveniences in the integration of non Java applications.

B. Communication infrastructure

The transmission of the floor control protocol messages requires a robust and reliable medium with the ability to traverse firewalls and NAT. It should be able to detect errors and provide encryption capabilities.

Beyond these requirements we also wanted to implement decentralized participant to participant functions, so the evaluation of peer to peer APIs seemed to be evident. We analyzed Suns Java Shared Data Toolkit 2.0 (JSDT), JXTA 2.1 and JGroups 2.2. JSDT was an interesting initiative but was abandoned by Sun in 1999, and lacked of proper documentation. (Meantime the source code has been opened.) JXTA proved a heavyweight solution with excellent feature list, but with large overhead and complicate and error prone configuration requirements. We chose JGroups [3] to use as communication infrastructure. It is a streamlined solution to provide reliable multicast messaging and group membership management. Messages can be sent reliably point to point, or point to group either unicast or multicast. When a message is sent to a group either all of the recipients get it or none of them. The group members get notification about joined, left and crashed group members.

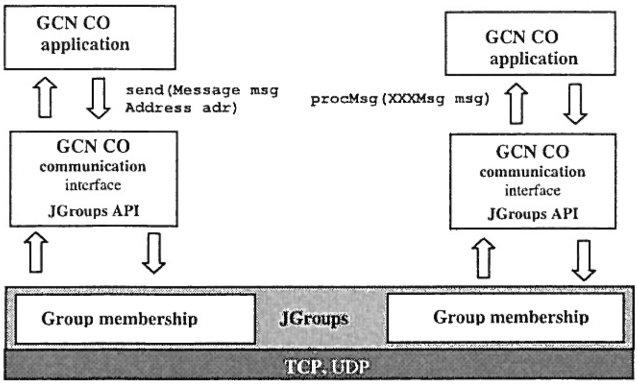

Fig. 3. Communication interface of GCN CO

On the top of JGroups we built a simple communication interface which enables us to easily change JGroups if needed. These includes a send method with a message and an address argument, and a set of overridden procMsg method to process the different kind of messages on the receiver side.

C. Media handling

Media handling tools are loosely integrated into the GCN system. At the beginning of the development neither of the available tools had a distinct advantage over of the others. We did not want to make such an important commitment at the beginning, so we designed GCN clients to be able to include different kind of media handling plug-ins. Each of the client applications can be configured with command line parameters which media plug-in it has to use.

Open H.323 is project aims to create a full featured, interoperable, open source implementation of the H.323 teleconferencing protocol. H.323 and so the Open H.323 project offers a Multipoint Control Unit based solution possibility.

An MCU can have several virtual rooms. The H.323 client programs (eg.:Netmeeting, OpenPhone, GnomeMeeting) can join one of this rooms. The clients who logged in to the same room are connected to each other with a multipoint to multipoint scheme. In a room every participant are equal. To change this behaviour, substantial modification had to be done on the MCU software. The basic idea was that the rooms can be mapped to languages, and a new kind of "room" had to be developed for the interpreters. The interpreter rooms (IR) are special because they "overlap" two other room with different languages, creating contact between them. Beside this the audio and video streams had to be decoupled because the video signal isn't connected to a given room but to the whole conference. Originally a conference in a room was controlled by voice, this had to be changed in order that the control commands could be issued from an other application.

The Open H.323 media plug-in required two kind of interface. The modified MCU had to be controlled according to the floor control messages. These messages are sent by the MCU interface of the H.323 plug-in. Every time a new participant gets the right to speak, the MCU must be reconfigured. The interpreters may want to swap their source and destination rooms, and each participant want to see the current speaker. The other interface which was included in the client controls the start-up and shutdown of the distinct H.323 client application.

Using an MCU leads necessarily to a centralized system. JMF and the later detailed VideoLAN allow a decentralized, media server less solution. The requirement is that the underlying network should be multicast enabled. JMF allowed us to implement a distributed media handling solution, based on the MUMUS [4] system. Originally the MUMUS media handling classes and the client applications was designed to be integrated in one application. During the implementation a beforehand unknown bug in JMF forced us to place the media streaming and displaying modules in a separate application. This application (skeleton in MUMUS's terminology) communicates with the client with messages. The skeleton has to be restarted every time when the source and destination of media streams are changed because JMF otherwise doesn't free the used multicast ports. The mechanism of changing the speakers is the following:

1. The client restarts the skeleton with an appropriate message.
2. The client waits for the message from the skeleton signalling its readiness.
3. The client sends the message to start the streaming and the receiving on the required channels.

The parameters of the transmission, like codec, picture size or bitrate, can be configured with XML files, which can be modified with a graphical editor.

The Open Source VideoLAN [5] project aims to provide video and audio streaming tool for local area networks. One of the result applications of VideoLAN (VLC) is a media processing unit. Its source can be file, media stream or data from a live source like web cam or frame grabber card. The source data can be filtered, transcoded and stored in a file, retransmitted in a stream or displayed. VLC provides several codecs and stream formats. For our application the best suited are the H.263 and MPEG 4 codecs. H.263 is part of the H.323 recommendation and was developed for videoconferencing on low bitrates. MPEG 4 is the most modern standard of MPEG, scales very well, so works also on low bitrates. VLC offers high quality scalable audio codecs too. VLC can stream with RTP either unicast or multicast, as well as through the built in http server, which eases to establish connection through firewalls at least in the receiving direction.

VLC itself a standalone application, but offers several possibility to programmatically access its services. It has a Corba and an http interface and all of its functionality can be accessed through command line. For testing purposes we used the http interface. Unfortunately it doesn't implement all of the documented features; stream transmission can't be started with the http interface.

The VLC plug-in starts an instance of VLC not with graphical UI, but with http interface. The role of this VLC is to display the received streams. The media interface for VLC translates the received commands to http messages. When a given client has to transmit a stream a new UI less VLC is started with the appropriate command line parameters. After the transmission the VLC process is killed programmatically.

III. CONCLUSIONS

The tests proved that the MUMUS based solution is not viable for an IVC application because of the mentioned "port leaking" bug in JMF. The time needed to restart the skeleton is too long to be acceptable. The VLC and the H.323 based solution worked equally well on local area network. VLC can provide much better quality, but it also need more processing power and based on a lower level standard. It gave possibility to implement the sub conference functionality between participants. On an entry level ADSL connection (384/128kbps) the upload bandwidth was a serious bottleneck for VLC. In that direction the connection was not usable. In the download direction the bandwidth was sufficient both the image and the sound quality were satisfactory. The H.323 based solution provided the quality that two Netmeeting clients can achieve on this connection. It is satisfactory for informal discussions, but can cause problems in professional use.

To ensure the required quality, the active participants like speakers, chairmen and interpreters needs at least 256 kbps upload bandwidth with the tools we tested. To further enhance the quality of service they can also dial-in on the GCN central through ISDN and establish a ppp connection. For users only have a passive role, the currently available broadband connections are suitable to participate in an interpreted video conference.

IV. ACKNOWLEDGEMENT

The research project was partly supported by the National Committee for Technological Development (OMFB) under the NKFP Global Conference Network contract.

V. REFERENCES

[1] http://www.rfc-editor.org/rfc/rfc3261.txt
[2] http://www.rfc-editor.org/rfc/rfc3550.txt
[3] http://www.jgroups.org
[4] Balázs Bámer, "Design and operation of a Java based distributed media processing system," in *Proceedings of the 2004 Intelligent System and Design Applications Conference*, submitted.
[5] http://www.videolan.org