A Web Services Framework for Collaboration and Videoconferencing

Geoffrey Fox, Wenjun Wu, Ahmet Uyar, Hasan Bulut, Shrideep Pallickara

Community Grids Laboratory, Indiana University

*gcf@indiana.edu, wewu@indiana.edu,
auyar@mailbox.syr.edu, hbulut@indiana.edu, spallick@indina.edu*

Abstract

Conference control has been studied for years but most researches focus on homogenous A/V collaboration. There is no conference control framework for integration of multiple videoconferencing systems such as H.323, SIP and AccessGrid. In this paper, we propose a web-serviced based scalable conference control framework for such heterogeneous collaboration system.

Based on XGSP A/V Web-Services framework, we propose a Global Multimedia Collaboration System (Global-MMCS), which can integrate multiple collaboration systems, and support various collaboration clients and communities. Now the prototype is being developed and deployed across many universities in USA and China.

Keywords

Web-Services, Collaboration, Global-MMCS, XGSP, NaradaBrokering

1. Introduction

Collaboration and videoconferencing systems have become a very important application in the Internet. There are various solutions to such multimedia communication applications, among which H.323 [9], SIP [17], and Access Grid [1] are well-known. At present, most collaboration systems are not designed in the approach of open system and cannot communicate with each other. It will bring substantial benefits to Internet users if we can build an integrated collaboration environment, which combines conferencing, streaming, instant messaging as well as other collaboration applications into a single easy-to-use, intuitive environment. Therefore, it is very important to create a more general framework to cover the wide range of collaboration solutions and allow different users from different communities to collaborate.

In this paper, we define such a common, interoperable framework called XGSP (XML based General Session Protocol) based on Web services technology for creating and controlling videoconferences.

Based on the web-services framework, we are developing Global Multimedia Collaboration System (Global-MMCS), which integrates various services including videoconferencing, instant messaging and streaming, and supports multiple videoconferencing technologies and heterogeneous collaboration environments.

The paper is organized in the following way: Section 2 describes the XGSP conference control framework. Section 3 introduces the implementation of XGSP prototype system – Global-MMCS. Related works are discussed in Section 4. And we give the conclusion and future work in section 5.

2. XGSP Framework

To integrate heterogeneous systems into one collaboration system, we need to reach the following goals:
  (1) Different kinds of application endpoints should join / leave in the same collaboration session.
  (2) Different providers for multipoint A/V and data collaboration should be connected together to build unified A/V and data multipoint channels.
  (3) A common user interface should be present for all the collaboration participants using different A/V and data application endpoints.

The first goal requires a common signaling protocol, which specifies the message exchange procedure between different types of collaboration endpoints and session servers. The conference control framework and multipoint messaging middleware are required for the second goal to integrate various multipoint RTP [18] and data communication servers. Also we need a flexible and common user interface that can contain different A/V application endpoints and aggregate other collaboration endpoints such as whiteboard, shared display.

Web-service seems to be the best candidate for this conference control framework since it can run across various platforms and is easy to be extended and understood. Conference control consists of three parts: user session management, application session management and resource contention management, also known as floor control. Since there are various conference

control protocols for different collaboration technologies, we have to wrap them into web-services and integrate these services in a more general framework. In addition, portlets [8] which have become an increasingly popular concept can be used to describe visual user interfaces to these heterogeneous services. A/V and other data collaboration endpoints can be wrapped into portlets to make them modular, reusable software components. Moreover a XGSP collaboration portal can be built by aggregating these portlets.

2.1 XGSP Conference Control Framework

Figure 1 shows the architecture of XGSP framework. The A/V media and data channel service provides multipoint A/V RTP and data channels for various collaboration applications. This service can be implemented on top of a distributed messaging middleware, NaradaBrokering [4] in order to create a unified, robust and scalable multipoint communication platform over heterogeneous networking environment. Other collaboration applications can also use the middleware for multipoint data delivery. Various collaboration systems including AccessGrid, H.323 and SIP are regarded as Web-services components in XGSP framework. They provide Web-services interface of their conference control protocols to the XGSP collaboration manager servers so that their RTP channels and data channels can be connected with the XGSP A/V media channel service and other data channel services under the control of the manager servers.
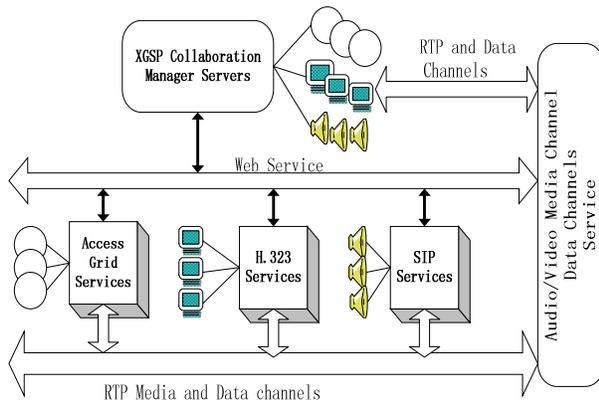
Figure 1 XGSP Framework

Under such a framework, all kinds of A/V application and data collaboration endpoints can communicate with each other whether they are directly connected to XGSP A/V media and data channel service or to different collaboration systems. Different collaboration systems are regarded as XGSP communities having multiple collaboration rooms. A collaboration room is the abstraction of multipoint A/V RTP and data channels. In centralized conferencing, A/V endpoints have to enter the collaboration room to attend the videoconferencing. It is noted that the concept of "room" is widely used in current collaboration systems. Based on these rooms from different communities, XGSP can create a collaboration session for all the endpoints. The XGSP session occurring in a community room is referred to XGSP sub-session. Users have the opportunity to enter either the XGSP session or the XGSP sub-sessions in local communities.

To support such a collaboration model, we have to define a XGSP conference control framework, which should be generic, easy to extend, reliable and scalable. XGSP conference control includes three components: user session management, application session management and floor control. User session management supports user sign-in, user create/terminate/join/leave/invite-into XGSP sessions. XGSP application session management provides the services to A/V and data application endpoints and communities, controlling multipoint A/V RTP and data channels. Floor control manages the access to shared collaboration resources. Although there are various floor control policies for different collaboration applications, XGSP should offer basic floor control mechanisms to support all these policies.

XGSP is a two-level control framework which includes top conference control servers and servers from other communities. Therefore three components can be designed in a hierarchy and distributed model to improve the scalability, which means that the top XGSP servers manages the whole XGSP session and the local servers only control XGSP subsessions. SOAP RPC commands can be used for the communication between the top XGSP servers and the local servers.

XGSP framework provides a collaboration portal to users for the aggregation of different collaboration services. The portal is a container of various collaboration portlets for each collaboration service. Each portlet provides client-side services to the XGSP portal for application session management and floor control. XGSP users can customize their collaboration portals by adding and removing collaboration portlets and changing the layer out of the portals. Therefore it is very easy to integrate various collaboration services such as A/V, whiteboard, shared display in XGSP framework.

Application Session Management

XGSP application session management has two tasks: control the XGSP session over all the media servers and help application endpoints to join and leave a session. In a XGSP conference, there may be different sessions, including A/V session, chat session, whiteboard session and so on. When a user schedules a XGSP conference, he will define which application session will be included in this conference and create profiles for each application session. For example, the profile of A/V application

session specifies the audio/video codec and the list of "rooms" from the communities involved in the session.

When a user wants to join the XGSP conference, the XGSP portal will launch portlets according to the conference profile and the user profile. Since each portlet provides local services to the XGSP portal, the portal can call these services to ask portlets to join or leave their application session.

Since different A/V application endpoints have their own signaling procedures for joining and leaving A/V sessions, we have to define a XGSP signaling protocol for H.225[12], H.245[13] (H.323 signaling protocols) and SIP as well as AccessGrid. The H.323 and SIP gateway transform these protocol specific messages into XGSP signaling messages so that H.323 and SIP A/V endpoints could communicate with the XGSP application session server. AccessGrid tools like VIC and RAT can join in the multicast XGSP subsession without using XGSP signaling procedure. But if they are running in a unicast environment, they have to rely upon XGSP signaling protocols to connect with the XGSP A/V application session server. For those local users, their endpoints can directly connect with the local session servers. This approach can also be applied for non-A/V sessions. For example, there are a lot commercial conferencing endpoints supporting T.120 [10] whiteboard. We have to build a T.120 gateway for integrating these whiteboard tools into XGSP system.

When the XGSP session is activated, the XGSP session server will link all the "rooms" in the session together by connecting multipoint A/V and data channels from different communities to the XGSP A/V Media and Data Channel Services. For H.323 and SIP communities, they connect with the XGSP channel Services by dialing in the H.323, SIP and T.120 gateway. Since a MBONE community like AG, has no signaling procedure, the XGSP servers will launch an AG agent that joins in the multicast A/V groups and forwards the packets between the top XGSP session and the AG multicast groups.

Floor Control

Floor control enables applications or users to gain safe and mutually exclusive or non-exclusive access to the shared object or resource in the conference. There are various floor control policies such as mediator-controlled, first-come-first-served, round-robin. Usually the choice of floor control policy depends upon the requirement of collaboration applications. For example, the shared PowerPoint application uses mediator-controlled policy, while a chess game takes round-robin. Therefore it is the collaboration application to define the floor role system and enforce floor control policy. XGSP should provide:

(1) Floor control primitives, including: request floor, release floor, grant floor, cancel floor, remove floor request. These primitives are exchanged between the conference participants and the conference server. All these floor control policies can be implemented on top of these primitives. A collaboration application can use these primitives to build its own floor control.

(2) mediator-controlled floor control: to support the mediator control policy, it is not appropriate for individual applications to implement it because there should be only one mediator in the conference. Collaboration applications have to define their own roles in the XGSP registration so that the mediator could assign the role of the application to each user. Each user can use XGSP control portlet to compete for conference mediator and request the application roles from current mediator.

2.2 Messaging Middleware for Multimedia Group Communication

Collaboration applications usually need a group communication service for both multipoint data and control information delivery. Different videoconferencing frameworks define their own group communication services based on different implementation strategies. Centralized conferencing systems usually depend upon a single conference server for group communication purpose, while distributed conferencing systems use IP multicast. For example AccessGrid uses Internet2 multicast for audio/video transmission. Both of the services have some limitations. Centralized conferencing systems don't have good scalability. And it is not easy to deploy distributed conferencing systems on current Internet because IP multicast seems to have a long time to become ubiquitously available. So some systems try to take hybrid approach, building group communication middleware over heterogeneous network environments. For example T.120 framework defines T.122 multipoint communication service for data collaboration applications. But T.120 doesn't support audio and video communication which is included in H.323 framework. Obviously it is very natural to integrate A/V group communication with other data collaborations. In our XGSP framework, we use the messaging middleware for group communication over heterogeneous networks.

NaradaBrokering from the Community Grid Labs is adopted as a general event brokering middleware, which supports publish-subscribe messaging model with a dynamic collection of brokers and provides services for TCP, UDP, Multicast, SSL and raw RTP clients. In addition, NaradaBrokering provides the capability of the communication through firewalls and proxies. It can operate either in a client-server mode like JMS or in a completely distributed JXTA-like peer-to-peer mode. We enumerate the advantages of deploying NaradaBrokering for XGSP group communication services:

(1) Covers the heterogeneity of network transportation and provides unified multipoint transportation API

Software multicast – Since it relies on software multicast, entities interested in conferencing with each other need not set up a dedicated multicast group for communications. Problems associated with setting of multiple unique multicast groups are exacerbated in settings with large number of clients.

Communication over firewalls and proxy boundaries – A lot of times two nodes/entities may be in realms separated by firewall and proxy boundaries. Irrespective of how elegant the application channels are, communications would be stopped dead in their tracks. NaradaBrokering incorporates strategies to tunnel through firewalls and authenticating proxies such as Microsoft's ISA and planet's proxy.

Communication over multiple transports – In distributed settings, events may traverse over multiple broker hops. Communication between two nodes may be constrained by the number and type of protocols supported between them. Multi-protocol support increases possibility of communications between two nodes. Furthermore, depending on the state of the network specific transports can be deployed to achieve better performance under changing network conditions.

(2) Provides robust, scalable and high efficient multipoint transportation services

Availability and scalability – There is no single point of failure within the NaradaBrokering messaging system. Additional broker nodes may be added to support large heterogeneous multimedia client configurations. NaradaBrokering's cluster based architecture allows the system to scale. The number of broker nodes may increase geometrically, but the communication *pathlength*s between nodes increase logarithmically.

Efficient routing and bandwidth utilizations – NaradaBrokering computes destinations associated with an event efficiently. The accompanying routing solution deploys links efficiently to reach these computed destinations. The routing solution conserves bandwidth by not overload links with data that should not be routed on them. Under conditions of high loads the benefits accrued from this strategy can be substantial.

3. Global-MMCS Prototype System

We are building a Global-MMCS prototype system across the sites in US and China. In China, we have a partner group called Admire [2] who is developing web-services of Admire system based on our framework. Indiana A/V research group will use the web-services interfaces to integrate Admire with AccessGrid H.323, SIP as well as RealNetwork streaming systems.

The following figure shows the architecture of Global-MMCS prototype system that we are developing. The XGSP Web Server, XGSP naming directory server and XGSP session server implement the web-services framework of Global-MMCS. Through SOAP connection, the XGSP Web Server can invoke web-services provided by other communities, such as Admire and SIP communities. The XGSP Session Server translates the high-level commands from the XGSP Web Server into signaling messages of XGSP, and sends these signaling messages to the NaradaBrokering Broker infrastructure to create a publish/subscribe session over the brokers. These brokers take the tasks of routing and forwarding video/audio events to various communities and collaboration clients.
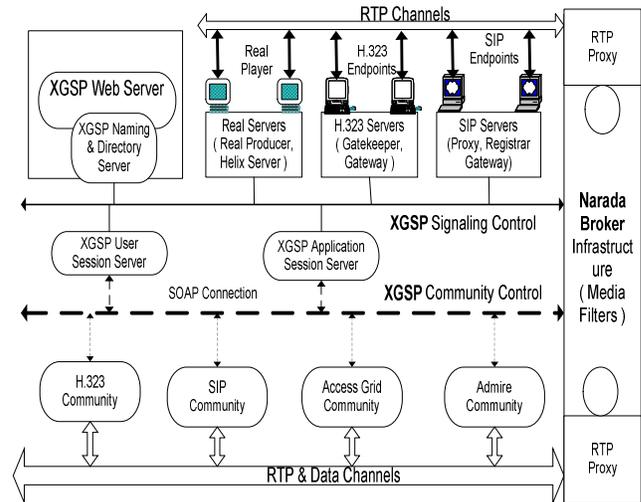
Figure 2 XGSP System Architecture

The H.323 Servers including a H.323 Gatekeeper and H.323 gateway create a new H.323 administration domain for individual H.323 endpoints, translate H.225 and H.245 signaling from these endpoints into XGSP signaling messages, and redirect their RTP channels to the NaradaBrokering infrastructure. The SIP Servers including a SIP Proxy, SIP Registrar and SIP Gateway create a similar SIP domain for SIP terminals and perform SIP translation. In addition, the SIP Proxy and SIP Gateway provide the services of Instant Messaging and Chat room for IM clients such as Windows Messenger. The Real Servers including a Real Producer and a Helix Server provide a streaming service to real-player and windows player. Enhanced with customer input plugin, our Real Producer can receive RTP audio and video packets from network, encode them into Real format and submit them to the Helix Server. Real-players can use RTSP to connect the Helix Server and choose the multimedia streams that they are interested in.

The NaradaBrokering infrastructure provides a scalable distributed messaging platform for RTP communications

in these A/V collaboration applications. Whenever a new session is activated across Global-MMCS, the same "topic" will be created inside NaradaBrokering system by XGSP session server. Any RTP client or server who wants to join in this session, it can "subscribe" to this topic and "publish" its RTP messages through RTP Proxies in the NaradaBrokering system.

3.1 Global-MMCS 1.0

In the first prototype of this system, we use a single XGSP media server for testing, which provides the services of bridging multicast and unicast, video-switching, video-mixing and audio-mixing to H.323, SIP as well as AG endpoints. We also implemented H.323, SIP and Real Servers for A/V clients. XGSP A/V Session Server is built to manage real-time A/V sessions, receiving messages from gateways and the web server, and performing appropriate actions on the media server. The web server provides an easy-to-use web interface for users to join multimedia sessions and for administrators to perform administrative tasks. In addition, users can start some audio and video clients through these web pages such as VIC, RAT and Real Player.

This early prototype can support VIC, RAT, H.323 terminals, SIP clients as well as Realplayer to join in Access Grid sessions. We support Ulaw audio codec, H.261 and H.263 video codec. The following picture shows the scenario of testing this system. We can see mixed video from VIC, Realplayer and Polycom ViaVideo. In background a snapshot of our web page is also seen. Mixed video streams are particularly important for users who can not receive more than one video stream such as a Polycom client. These users get the pictures of four participants in a meeting through one video stream.

Figure 3 Mixed Video in Global-MMCS 1.0
Performance Test

We conducted extensive performance tests on audio and video servers. The performance of the audio server depends on the number of participants and active speakers in a session, and the number of audio sessions at a time. First we test the scalability of our audio server for one audio session, and tested the effects of having multiple sessions at a time. The audio server runs in a 2.5GHz Pentium 4 CPU, 512MB memory, Windows XP machine. When there is only one active speaker, it provides a good quality audio for up to 300 participants. After that it starts dropping packets since it can not process them on time. For the case of two active speakers, it provides good quality audio for up to 275 participants and then it starts dropping packets. We created audio sessions with 50 participants and two active speakers for each session. Our tests show that audio server can support 5 concurrent sessions (250 participants in total) without any packet droppings.

We have tested the performance of video forwarding and video mixing separately. In forwarding case, one user sends a H.263 video stream to the server machine - 1.2GHz Intel Pentium III dual CPU, 1GB MEM, RedHat Linux 7.3, and server forwards it to many clients. The test shows that our video server is capable of supporting 300 clients if there is only one video sender. When the receiver number increases to 400, it starts dropping packets and the transmission latency also becomes pretty high. We should also note the fact that this video stream had an average bandwidth of 600kbps which is much higher than a normal conferencing video source. In mixing test, we run the video mixer in a Linux box like the one in previous experiment. The video mixer mixes four identical H.261 video streams ranging from 100kbps to 200kbps. In this case, it shows that video mixing is a CPU intensive process and this machine can only handle three video mixers at a time.

Through the development of this early prototype, we get a lot of experiences and lessons which have been fully considered in the current development

(1) A single A/V MCU server is only capable of processing medium scale of videoconferences. Distributed A/V MCU architecture has to be introduced to improve the scalability.

(2) Although we build a simple web portal for different A/V clients, it is not easy to enhance it and add more collaboration tools. So we decide to build portlets for different collaboration application tools, and use these portlets to create a powerful collaboration portal.

(3) In the prototype, we also test A/V transmission in NaradaBrokering to see whether NaradaBrokering can support high-performance A/V communication.

After we made some optimizations on the message transmission of NaradaBrokering system, it shows

excellent performance for A/V communication. We have tested the performance of NaradaBrokering in the case of high bandwidth video transmission: A video client sends a video stream to the NaradaBrokering server and 400 receivers receive it. This video stream has an average bandwidth of 600Kbps. So totally it takes up 240Mbps of network bandwidth. We compare the results of NaradaBrokering with the performance of a JMF [21] reflector program written in Java. Figure 4 shows the measurement results. Our messaging systems outpaces JMF reflector in the performance of delay and jitter. It is good enough to be used for large scale videoconferencing applications. And one broker can support more than a thousand audio clients or more than 400 hundred video clients at one time providing a very good quality.

Average delays/packet for 12 (of the 400 total) video-clients.
NaradaBrokering Avg=80.76 ms, JMF Avg=229.23 ms

Average jitter/packet for 12 (of the 400 total) video clients.
NaradaBrokering Avg=13.38 ms, JMF Avg=15.55 ms
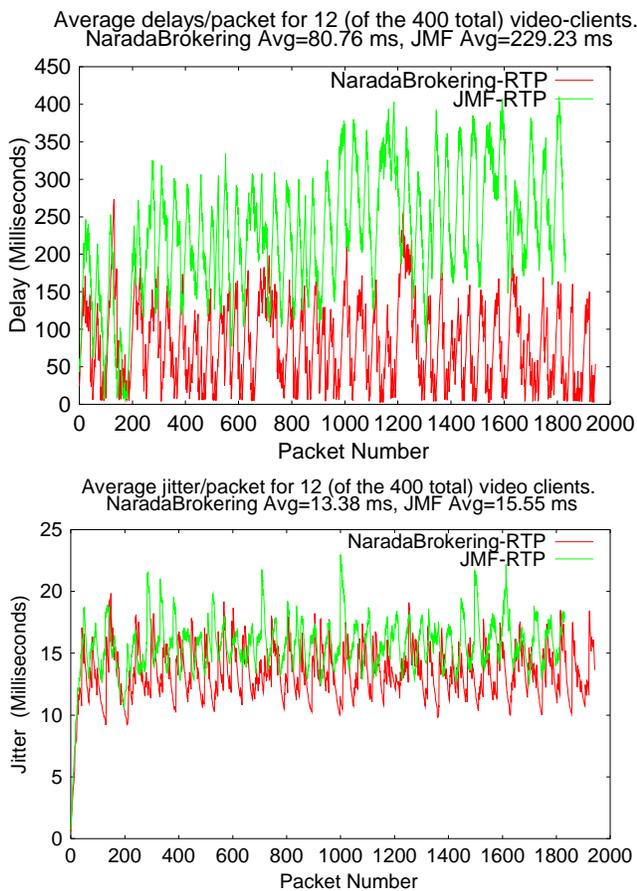
Figure 4 . Comparison between the performance of NaradaBrokering and JMF

3.2 Global-MMCS 2.0

At present, we are working on the second prototype, which will greatly enhance the current prototype in scalability. Based on the XGSP framework and the NaradaBrokering event middleware, we are building an open source protocol independent "MCU" which will scale to an arbitrary number of users and provide integrated collaboration services. And we will deploy it globally and test with thousands of simultaneous users later this year.

The function of the A/V media server will be distributed in NaradaBrokering architecture. Brokers have provided RTP packet forwarding service by building RTP event and RTP "topic". The components of audio mixer and video mixer will be built into separate A/V processing units which can accept RTP events and generate RTP events to new mixer "topic". Other audio and video processing functions such as titling, picture-in-picture can be implemented in RTP modules and embedded into the brokers. Streaming and archiving servers like Helix Server, Voyager Server will also be wrapped into the brokers and provide these services through NaradaBrokering middleware.

Collaboration clients will be built into portlets by creating Java Applet or ActiveX controls for these clients and adding them into HTML pages. A collaboration portlet opens local services for XGSP application session management and floor control. When a new collaboration tools is added into Global-MMCS, we can just upload the portlet into the portlet manager. Users have the right to choose their portlets, customize the rendering of the portlets and make the layer out of the portlets in their browsers. We will use this component based approach to add many collaboration tools into Global-MMCS prototype, including chat, whiteboard, SVG sharing, shared display and so on.

Web-services for Admire will be fully implemented in the new prototype. Global-MMCS will integrate AccessGrid and Admire as well as other H.323 and SIP communities to build a global collaboration platform. Thousands of users across US, Europe and China can test and use this powerful system. Since Global-MMCS is an open source project, we will provide the source code and API to encourage the development of other collaboration tools. We will release our first version late in the year. In addition, portlets to AccessGrid may be available in this summer.

4. Related work and comparison

Problems related to conference control framework have been studied extensively over the years [9, 10, 12, 16, and 22]. However, most of the works discuss only homogenous videoconferencing, including H.323, SIP and MMUSIC [5]. ITU-T developed conference control protocols as a part of the H.323 series of recommendations. It is reported [16] that T.124 [11] has the scalability issue because of the inefficient database replication algorithm. And H.323 Audio/Video collaboration takes the simple protocol in H.243 [10]

rather than T.124. And the IETF's Multi-Party Multimedia (MMUSIC) working group has gave up its own solution and removed conference control from the WG charter. Recently SIP research group begun to develop their framework and produced a few drafts [16, 22]. But SIP work is still in the beginning phase and has not been widely accepted. XGSP tries to define web services framework in which H.323, SIP as well as MMUSIC could be seamlessly integrated into a single, easy-to-use collaboration environment.

Global-MMCS provides opportunities for those who either use H.323 and SIP clients such as polycom, windows messenger or only have unicast network and NAT firewalls. For example, because the multicast link between US and CHINA is not stable, Admire systems which have been widely deployed in CHINA can't communicate with Access Grid. Global-MMCS can play as a powerful bridge between Access Grid and Admire system. Global-MMCS can also provide Access Grid portlets as well as other collaboration portlets which can be integrated into other Grid portals.

VRVS [24] is a pure-software videoconferencing system over a reflector infrastructure, which can provide similar services like Global-MMCS1.0 prototype. It is a mature and dependable system used for some time by many people. But we believe our research focuses on different issues from VRVS though we don't have enough documents to know the details of VRVS:

(1) open source scalable "MCU" based on messaging middleware

VRVS builds its collaboration service on top of reflector infrastructure which is a kind of software multicast. While NaradaBrokering is a distributed messaging middleware, not only supporting secure, high performance and reliable group communication, but also providing publish/subscribe API which is a standard message API for java programmers. Furthermore since both Global-MMCS and NaradaBrokering are open source projects, it will be easy for collaboration developers to build their own collaboration applications on them.

(2) integration with other communities

VRVS seems to be capable of supporting MBONE tools, H.323 terminal and QuickTime player. It doesn't support a H.323 MCU and its connected terminals as a community, and include them in a collaboration session. Through web-services technology, XGSP framework is capable of wrapping several collaboration systems into web-services and creating XGSP conferences over these systems.

(3) portlet for user interface, providing more collaboration tools

VRVS provides a good web interface, but it is not based on portlet approach. Global-MMCS system will provide portlet based web portals to users, which are very flexible for integrating new collaboration functions. In addition, Global-MMCS portlets can be easily embedded into other Grid application portals.

5. Conclusion

In this paper, we have described a web-service based framework XGSP for conference control. Under the XGSP framework, not only various audio/video endpoints but also communities can be integrated into a single A/V collaboration environment. This framework implements user and A/V application session management and floor control function in a scalable structure over heterogeneous collaboration systems.

Based on this framework, we are developing a prototype system, Global-MMCS. Such an integrated collaboration environment greatly benefits those users that want to enter Access Grid world via H.323 and SIP clients, providing the services of videoconferencing, instant messaging and streaming to various clients, and integrating different collaboration communities into a global collaboration platform.

6. References

[1] Access Grid Project, http://www.accessgrid.org.

[2] Admire Project, http://www.nlsde.buaa.edu.cn/projects/Admire

[3] Geoffrey Fox, Wenjun Wu, Ahmet Uyar, and Hasan Bulut, "A Web Services Framework for Collaboration and Audio/Videoconferencing", *the 2002 International Multiconference in Computer Science and Computer Engineering, Internet Computing(IC'02)*, June 2002, Las Vegas

[4] Geoffrey Fox, Wenjun Wu, Ahmet Uyar, Hasan Bulut, Shrideep Pallickara, "Global Multimedia Collaboration System", *1st International Workshop on Middleware for Grid Computing*, June 2003, Rio de Janeiro, Brazil.

[5] Handley, M., Crowcroft, J., Bormann, C., and Ott, J, "Very large conferences on the internet: the internet multimedia conferencing architecture", *Computer Networks 31* (1999).

[6] HearMe Audio conference system, http://www.hearme.com,

[7] Jiann-Min Ho, Jia-Cheng Hu, Peter Steenkiste, "A conference gateway supporting interoperability between SIP and H.323", *Proceedings of the ninth ACM international conference on Multimedia, 2001* , Ottawa, Canada

[8] Jetspeed Overview, Accessed from http://jakarta.apache.org/jetspeed/site/index.html.

[9] International Telecommunication Union, Packet based multimedia communication systems", Recommendation H.323, Geneva, Switzerland, Feb. 1998.

[10] International Telecommunication Union. Data protocols for multimedia conferencing, Recommendation T.120, July 1996.

[11] ITU, Generic conference control, Recommendation T.124, Feb. 1998

[12] ITU, Terminal for low bit-rate multimedia communication, Recommendation H.243, Feb, 1998.

[13] ITU, Packet-base multimedia communications systems. Recommendation H.323, Sep, 1999

[14] ITU, Calling Signaling Protocols and Media Stream Packetization for Packet-based Multimedia Communication Systems", Recommendation H.225.0, Feb, 2000

[15] ITU. Control Protocols for Multimedia Communication", Recommendation H.245, Feb, 2000

[16]Koskelainen P., Schulzrinne H. and Wu X, "A SIP-based Conference Control Framework", *NOSSDAV'02*, May 12-14, 2002, Miami Beach, Florida, USA.

[17] J. Rosenberg et al., "SIP: Session Initiation Protocol", RFC 3261, Internet Engineering Task Force, June 2002, http://www.ietf.org/rfc/rfc3261.txt.

[18]RTP: A Transport Protocol for Real-Time Applications (IETF RFC 1889) http://www.ietf.org/rfc/rfc1889.txt.

[19] K. Singh, G. Nair, and H. Schulzrinne, "Centralized conferencing using SIP"*,* Internet Telephony Workshop 2001, New York, April. 2001.

[20] K. Singh and H. Schulzrinne, Interworking between SIP/SDP and H.323, *In Proceedings of the 1st IP-Telephony Workshop (IPtel 2000)*, Berlin, Germany, Apr. 2000.

[21] Sun Microsystems, Java Media Framework 2.1, http://java.sun.com/products/javamedia/jmf/2.1.1/index.html, 2001

[22] Wu, X., Koskelainen P., Schulzrinne H., Chen C. "Use SIP and SOAP for conference floor control", Internet Draft, Internet Engineering Task Force, Feb. 2002. Work in progress.

[23] Ahmet Uyar, Shrideep Pallickara, Geoffrey Fox, "Towards an Architecture for Audio/Video Conferencing in Distributed Brokering Systems", *To appear in the proceedings of The 2003 International Conference on Communications in Computing*, June 23 - 26, Las Vegas, Nevada, USA.

[24] Virtual Rooms Video Conferencing System, www.vrvs.org

[25] Wenjun Wu, Ahmet Uyar, Hasan Bulut, Geoffrey Fox, "Integration of SIP VoIP and Messaging Systems with AccessGrid and H.323", *To appear in the proceedings of The 2003 International Conference on Web Services (ICWS'03)*, June 2003, Las Vegas, ND, USA.