# Design and Implementation of Audio/Video Collaboration System Based on Publish/subscribe Event Middleware

Geoffrey C. Fox
*Indiana University, Community Grid Computing Laboratory*

Wenjun Wu
*Indiana University, Community Grid Computing Laboratory*

Ahmet Uyar
*Syracuse University, Department of Electrical Engineering and Computer Science ; Indiana University, Community Grid Computing Laboratory*, auyar@syr.edu

Hasan Bulut
*Indiana University*

# Design and Implementation of Audio/Video Collaboration System Based on Publish/subscribe Event Middleware

**Geoffrey C. Fox[1], Wenjun Wu[1], Ahmet Uyar[1,2], Hasan Bulut[1]**
**[1]Community Grid Computing Laboratory, Indiana University**
**[2]Department of Electrical Engineering and Computer Science, Syracuse University**
**gcf@indiana.edu, wewu@indiana.edu, auyar@mailbox.syr.edu, hbulut@indiana.edu,**
**Indiana Univ Research Park, 501 North Morton Street, Suite 222, Bloomington, IN47404**

**Abstract**

In this paper we present our A/V collaboration system based on our XGSP collaboration framework and NaradaBrokering messaging middleware. Using publish/subscribe event model, this system can provide videoconferencing services to heterogeneous endpoints such as H.323, SIP and Access Grid. This paper discusses the common a/v collaboration model shared by all kinds of A/V conferencing clients and introduces the details about how to implement such a model based on publish/subscribe event middleware.

**Keywords**

A/V collaboration, Publish/subscribe, 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 [1], SIP [2], and Access Grid [3] 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 A/V collaboration environment to all kinds of A/V endpoints. In the paper [4], we defined a common, interoperable framework called XGSP (XML based General Session Protocol) based on Web services technology for creating and controlling videoconferences. Based on the framework, we are developing Global Multimedia Collaboration System (Global-MMCS) [5], which integrates various services including videoconferencing, instant messaging and streaming, and supports multiple videoconferencing technologies and heterogeneous collaboration environments.

In this paper, we introduce the work on the A/V collaboration system which can support different endpoints such as H.323, SIP and Access Grid. Although these collaboration clients are designed in quite different frameworks, they have the common A/V conferencing model, that's sharing a group of low-delay audio and video streams. Besides of the different protocols, the major difference among these endpoints is the capability of receiving and rendering these A/V streams. To address the issue and support all these endpoints in the same videoconferencing, publish/subscribe event model seems to be very promising. Because the group of A/V streams may be published and an endpoint can make its own subscription based on its capability. In the development of the A/V collaboration system, NaradaBrokering [6], a general event brokering middleware, is used to provide A/V publish-subscribe messaging.

The paper is organized in the following way: Section 2 introduces the XGSP conference control framework and NaradaBrokering. Section 3 and 4 describes the design and implementation of A/V collaboration system based on XGSP framework and NaradaBrokering middleware. And we give the conclusion and future work in section 5.

## 2. XGSP collaboration Framework

Problems related to conference control have been studied extensively over the years. However, most of the works discuss only homogenous videoconferencing, including H.323, SIP, T.120 and MMUSIC. Our job is to define a web services collaboration framework in which H.323, SIP as well as MMUSIC could be integrated. 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. 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[7], H.245[8] (H.323 signaling protocols) and SIP as well as Access Grid. Floor control manages the access to shared collaboration resources.

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 Access Grid 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 a hybrid approach, building group communication middleware over heterogeneous network environments. For example T.120 [9] framework defines T.122 [10] 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 creates a better architecture to integrate the 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. A/V Collaboration over publish/subscribe Middleware

### 3.1 XGSP A/V Collaboration Model

Although different A/V endpoints have quite different implementation and capabilities, they share a common A/V collaboration pattern. Assume in an A/V session, there are video streams (VS) and audio streams (AS): VS $\{v_1, v_2, \ldots v_m\}$ , AS $\{a_1, a_2, \ldots a_n\}$. And we also have A/V endpoints: $E_1, E_2, \ldots E_n$. Each endpoint may send a single or multiple video streams, but only send an audio stream. In this A/V collaboration session, each endpoint adds new A/V streams into VS and AS by sending audio and video RTP packets, or removes streams by sending RTCP "BYE" packets to leave the session. At the same time, each endpoint gets a subset from VS and AS by receiving RTP streams and rendering them.

Different types of A/V endpoints have different collaboration capabilities. Multicast endpoints are able to receive multiple video and audio streams, display all the video streams in their screens, and mix all the audio streams by themselves. For example, Access Grid endpoints receive all the streams in VS and AS to create the duplications of VS and AS, and allow users to make selection of rending video and mixing audio. On the other hand, unicast endpoints like Polycom ViaVideo can only receive and play a single video and audio stream. So they can't attend the collaboration with Access Grid endpoints. This issue may be addressed by introducing some middleware, which offers the description data of the streams to the users of these endpoints, allows them to switch among the video streams, and mix audio streams for them.

Publish/subscribe middleware is a perfect candidate for this purpose. An A/V RTP stream is regarded as a "topic" and each RTP packet from this stream as an "event" for this topic. Only the sender of this stream can "publish" A/V events to this topic. Other endpoints need to subscribe to this topic in order to receive the stream data. In this way, the Publish/subscribe middleware provides a group communication service for those unicast endpoints. Although there are some arguments about the performance of audio video communication over the publish/subscribe system, our previous work [11] has shows that NaradaBrokering messaging middleware can support a large scale of multimedia communication very well.

Besides of the RTP events for the stream topic, some major events for each stream topic have to be defined to describe the change in this stream. They include five major events: *NewStreamEvent, ByeEvent, TimeOutEvent, Active-to-Passive, Passive-to-Active*.

Table 1 major events of an A/V stream

| **Event Name** | **The change in the status of the stream** |
|---|---|
| NewStreamEvent | This stream has been created |
| ByeEvent | The stream gives a BYE RTCP packet, indicating it has left the session |
| TimeOutEvent | This stream has not send any RTCP packet for a long time, indicating it may have left the session |
| Active-to-Passive | The stream has stopped sending RTP packet |
| Passive-to-Active | The stream resumed sending RTP packet |

Based on these events, Unicast endpoints can maintain the duplications of VS and AS list just like multicast endpoints. Furthermore Active-to-Passive and Passive-to-Active events notify the endpoints whether the streams have new data at that moment, which sometimes are very important to A/V collaboration. For example, since the endpoints only get a mixed audio for all the audio streams, it is very useful for the endpoints to know who is speaking in this mixed audio. These events help the users to get the "focus" of the A/V collaboration.

## 3.2 A/V Collaboration Architecture

Based on this model, we can integrate H.323, SIP, and AG A/V endpoints into one system. The following figure shows the architecture for this A/V collaboration system.

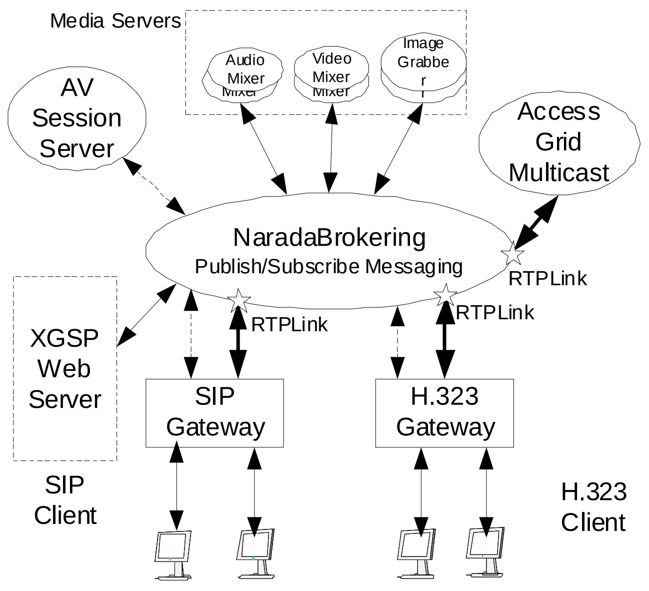


Figure 1 A/V Collaboration system architecture

**(1) NaradaBrokering message middleware**

The NaradaBrokering publish/subscribe middleware provides messenger service to all the A/V components. XGSP control messages can be carried by NaradaBrokering JMS style event. But RTP packets have to be transformed into a special event by RTP links components in border brokers before they can be routed in NaradaBrokers. And the special RTP events will be transformed back into normal RTP packets when they leave for their subscribers. Therefore for every legacy RTP audio or video client, one corresponding RTPLink needs to be set up at a broker within broker network.

Currently there are two types of RTP links: RTPLink and MulticastRTPLink. A RTPLink receives raw RTP packets from a unicast RTP endpoint and publishes them on a given topic after wrapping them in RTP events. Each RTPLink occupies two NaradaBroker topics to publish its RTP and RTCP data. For the performance reason, these topics are integer, for example: RTP topic is an even number: 2000 and RTCP topic is the following odd number: 2001. A MulticastRTPLink works as a reflector between NaradaBrokers and multicast groups, encapsulating raw RTP packets from a multicast IP address to RTP events, publishing these events to NaradaBrokers, and forwarding the data it receives from broker network on the same IP address. Because there are usually multiple RTP streams in the same IP multicast group, MulticastRTPLink has to publishes incoming RTP packets from different streams to different NaradaBroker topics. It assigns the base topic number to the first received stream, and increases the topic numbers when new streams arrive.

**(2) Audio Mixer, Video Mixer, Image Grabber**

Under the support of NaradaBroker middleware, some A/V application components are needed to implement the

XGSP A/V collaboration model and optimize the performance of the collaboration.

Some unicast audio endpoints have to rely upon an extra audio mixer to create a mixed audio stream from all the audio streams in the session. Since the audio mixer reduces the number of the streams that audio endpoints need to receive, audio mixing can relieve the processing overhead in the endpoints and reduce some audio traffic across the network. Video mixing is particularly important for users who can not receive more than one video stream such as a Polycom client. By video mixing, these users get the pictures of multiple participants in a meeting through one video stream. Furthermore a video mixer can also decrease the number of video streams so that the bandwidth consumed by video traffic could be lowered. This optimization seems to be very necessary to support large scale videoconferencing.

Image grabbers capture video streams and save them as static JPEG files. These snapshot pictures can be embedded into the control panel of each endpoint, which visualize the VS set in the session.

All the media processing components can be distributed among the pool of the media servers connected to NaradaBrokering infrastructures. We run a media component manager server, which can initiate an instance of the media processing components and start it up on any available media server.

**(3) H.323, SIP Gateway Servers, A/V Session Server**

These servers deal with the control layer of the A/V collaboration system, which co-ordinate the services of NaradBrokering middleware and A/V application servers. 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 A/V session server. The session server implements session management logics including creating/destroying A/V sessions, allowing endpoints to join/leave session and make audio/video selection, managing A/V application components like audio mixers, video mixers as well as image grabbers.

## 4. Implementation

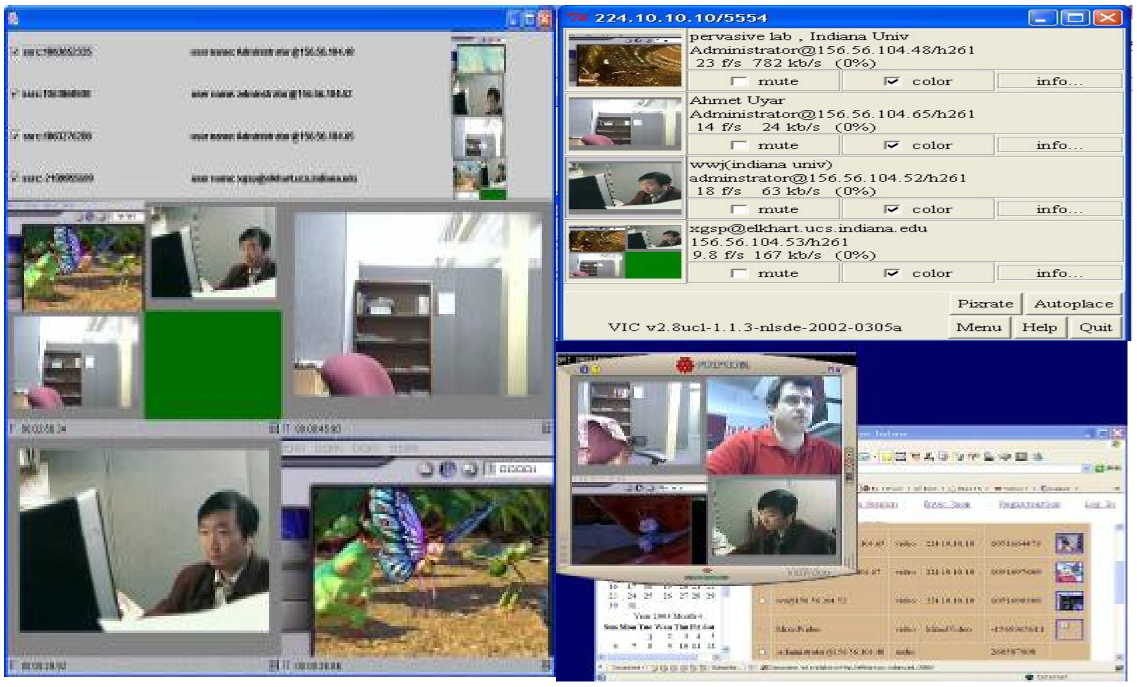
We have implemented an A/V collaboration prototype system based on the NaradaBrokering middleware. All the A/V processing components, including the video mixer, audio mixer as well as the image grabber servers are developed using Java Media Framework [12]. Java Media Framework (JMF) API enables audio, video and other time-based media to be added to Java applications and applets. This optional package, which can capture, playback, stream and transcode multiple media formats, gives multimedia developers a powerful toolkit to develop scalable, cross-platform technology. Based on the codecs provided by JMF, our system can support ULaw audio codec, H.261 and H.263 video codec for all the A/V endpoints. In the development, we also fixed some minor bugs inside JMF and integrate H.261 encoder into it.

To implement the H.323 gateway and SIP gateway, we use the protocol stacks from the open source projects, including OpenH323 [13] and NIST-SIP [14] project. The A/V Session Server is built to manage real-time A/V sessions, receiving messages from gateways and the web server through different control topics on the NaradaBrokering. The XGSP web server based on Apache Tomcat provides an easy-to-use web interface for users to join multimedia sessions and for administrators to perform administrative tasks.

We provide different user interfaces to different A/V endpoints. Access Grid users only need to visit the correct multicast address to attend the collaboration. The system provides two types of consoles for unicast capable endpoints, named Unicast AG portlet and H.323 portlet. Both portlets depend upon XGSP servers to build up their A/V stream list in a videoconference and allow users to subscribe any number of video streams for receiving and rendering. The difference between them is: a H323 portlet only permits a single video selection and doesn't offer the video rendering since H.323 endpoints can render video by themselves. A Unicast AG portlet allows multiple video selections and render them for users.

Let's give an example to show how the system can work. Assume two users A and B want to join in the multicast A/V session in the address: 224.10.10.10/5554 ( video ) and 224.10.10.10/6666 ( audio ). But A stays in unicast network and B has to use Polycom Via Video. So they ask GlobalMMCS administrator to create a conference connecting to the multicast A/V group. In the creation of this conference, XGSP Web Server asks AV session server to create MulticastRTPLink for the multicast groups in NaradaBrokers and initiate three instances of audio mixer, video mixer and image grabber components in the pool of media servers. After the conference is ready, User A starts its Unicast AG portlet to connect to the AV session server and join the conference. The AV session server creates RTPLinks for the audio and video channels of this user. User B then starts its polycom endpoint to dial in the H.323 Gateway, which forwards the request to the AV session server. The server also builds RTPLinks for this H.323 endpoint. When all the procedures are completed, User A and B can select the video streams from the multicast group and themselves, and receive the mixed audio in the conference.

The following picture shows the scenario of the example in testing the system. On the left side, a Unicast AG portlet gets four video streams including a mixed video and renders them on the desktop. On the right side, an Access Grid VIC endpoint in the multicast group receives the same streams and a Polycom Via Video displays a mixed video including four video streams.



```
224.10.10.10/5554
pervasive lab , Indiana Univ
Administrator@156.56.104.48/h261
 23 f/s  782 kb/s  (0%)
Ahmet Uyar
Administrator@156.56.104.65/h261
 14 f/s   24 kb/s  (0%)
wwj(indiana univ)
adminstrator@156.56.104.52/h261
 18 f/s   63 kb/s  (0%)
xgsp@elkhart.ucs.indiana.edu
156.56.104.53/h261
9.8 f/s 167 kb/s  (0%)
VIC v2.8ucl-1.1.3-nlsde-2002-0305a
```

Figure 2 Prototype Demo pictures

## 5. Comparison

Compared to some similar system like VRVS [15], our system tries to take a more open and integrated solution: VRVS builds its collaboration service on top of reflector infrastructure which is a kind of software multicast. In contrast, NaradaBrokering not only supports a secure, scalable and reliable group communication, but also provides a publish/subscribe API which is a standard message API for Java programmers. Furthermore XGSP defines a very general collaboration framework for all kinds of application. Therefore it will be easy for collaboration developers to build their own collaboration applications and integrate these applications into our Global-MMCS system. But VRVS doesn't provide such an open framework for integration of collaboration tools from third party.

## 6. Conclusion

In this paper we have presented our A/V collaboration system which provides services to heterogeneous endpoints. This collaboration system is developed based on our XGSP collaboration framework and NaradaBrokering messaging middleware. Such an integrated collaboration environment greatly benefits those users that want to enter Access Grid world via H.323 and SIP clients.

Further works include: extending the scalability of the system to 10,000 users, adding archiving and replay services.

## 7. Reference

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

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

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

[4] Wenjun Wu, Hasan Bulut, Ahmet Uyar, , Geoffrey Fox, "A Web-Services based Conference Control Framework for Heterogenous A/V collaboration", 7th IASTED International Conference on Internet and multimedia systems and applications, August 13-15, 2003, Honolulu, Hawaii, USA.

[5] 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.

[6] Geoffrey Fox and Shrideep Pallickara, NaradaBrokering: An Event Based Infrastructure for Building Scaleable Durable Peer-to-Peer Grids. Chapter 22 of "Grid Computing: Making the Global Infrastructure a Reality".John Wiley April'03.

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

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

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

[10] International Telecommunication Union. Multipoint Communication Service, Recommendation T.122, February 1998.

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

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

[13] OpenH323 Project, http://www.openh323.org

[14] NIST SIP, http://snad.ncsl.nist.gov/proj/iptel/.

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