

# **A Web Services Framework for Collaboration and Audio/Videoconferencing**

*Geoffrey Fox, Wenjun Wu, Ahmet Uyar, Hasan Bulut*  
Community Grid Computing Laboratory, Indiana University  
[gcf@indiana.edu](mailto:gcf@indiana.edu), [wewu@indiana.edu](mailto:wewu@indiana.edu),  
[auyar@mailbox.syr.edu](mailto:auyar@mailbox.syr.edu), [hbulut@indiana.edu](mailto:hbulut@indiana.edu)

Tel: 812-8561245

Postal address: Indiana Univ Research Park, 501 North Morton Street, Suite 222,  
Bloomington, IN47403

Submitted to: PDPTA'02

## **Abstract**

At present, there are various videoconferencing frameworks, such as H.323, SIP, Internet Audio and Access Grid, which usually cannot directly interact with each other. Web Services have been proposed as a new way to produce modular Web-based components. So in this paper, we present a possible Web Service framework for an audio/video collaboration system. Under such framework, we can implement a collaboration system, which can support H.323, SIP, Access Grid in the same audio and video session. We describe our approach in terms of the clients, session servers and communication channels. We introduce an XML description XGSP for collaborative sessions which encompasses existing systems and allows for extensions to define richer environments. This paper illustrates the value of Web Services in providing an interoperability framework as well as providing a new platform for building general collaboration environments. In future papers, we will describe how this approach can be used to incorporate further collaboration models such as those like JXTA from the peer-to-peer realm and emerging powerful messaging infrastructure. It can also be extended to support further functionality like white boards and more general shared applications.

**Keywords:** videoconferencing, web service, H.323, Access Grid, SIP

## **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[6], SIP[8], and Access Grid[1] are well-known. H.323 is an umbrella standard designed by ITU for multimedia conferencing over IP-based networks. It has been widely adopted by the industry of videoconferencing and there are many H.323 based systems, such as Polycom, and VCON. SIP is a standard of IETF, which is an alternative solution to H.323, especially for Voice over IP. Some collaboration systems, such as Hearme[5] use SIP for session initiation. Access Grid is a derivation from MMUSIC conference[4], which uses MBONE tools and can support a large scale audio/videoconference based on a multicast network. At present, all the systems are effective and have their own separate user communities which cannot easily communicate together. Further they have features that sometimes can be compared but often the systems make implicit architecture and implementation assumptions that hamper interoperability and functionality. Therefore it is very important to create a more general framework to cover the wide range of collaboration solutions and enable different users from the different communities to collaborate.

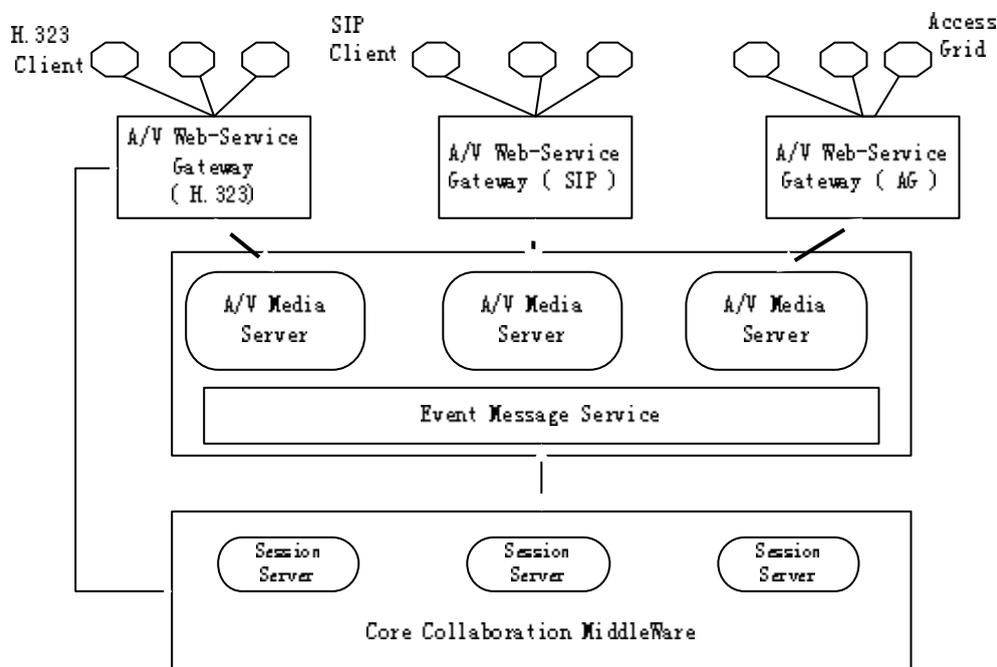
Recently Web Services has become increasingly popular because of their prospect of linking various applications running over the Internet by providing standard interfaces and communication channels. The idea of using Web Services to provide a standard interface to audio/video conferences over the Internet and collaboration services seems very attractive.

An A/V collaboration system consists of three parts: clients, session servers and communication channels. For example, in an H.323 based system, a client refers to the H.323 endpoint that is capable of sending audio and video. A session server refers to the Multipoint Controller that can create multipoint session. A communication channel is the Multipoint Processor that can mix audios and videos from different clients. In an Access Grid system, a client is based on the MBONE audio/video tools such as RAT and VIC. Further there is a venues server in Access Grid, which is responsible for scheduling meeting. Multicast RTP Channels are the communication infrastructure for Access Grid.

Each system has a different implementation for the client and server components and different communication protocols between them. So our idea is to build the web service for each component, and define a general collaboration protocol in XML to describe the interaction between the components. In this way, each component becomes a web service entity that can be described in WSDL[11], and can communicate with each other using XML based protocol, such as SOAP[10]. The advantage of such a framework is obvious: different clients, session servers and communication channels from different system can be transformed into a general web service components and work with each other under the general framework. Note that Web Services allow one to bind communication channels to different protocols; SOAP can be used for control messages without real time constraints. However one needs to bind the time sensitive media channels to high performance protocols like RTP.

This paper is organized as follows: section 2 presents the architecture and communication protocols in this framework, in section 3 we discuss the two examples that develop A/V collaboration systems using such a framework, section 4 and 5 give the future work and conclusion respectively.

## 2. Architecture of Audio video collaboration Web Service



service entities. The second is the Media Server, which is a web service entity for RTP communication channels between the clients. The third entity is the session server, providing the basic services for an A/V session, such as constructing collaboration groups, maintaining the membership, advertising collaboration resources and binding communication channels. The session servers can be termed the core collaboration middleware.

For each of web service entity in our framework, we can use WSDL to define its interface and operation. Each entity can use XML message mechanism to communicate with each other and a session protocol to work together. We define a XML based protocol, XGSP (XML-Based General Session Protocol), which describes the interaction between the components in the same session. The details of the architecture are discussed in the following.

## 2.1 Web Service Interface of collaboration components

### (1) Client

A client can send some session requests to a session server to create or join the session so that it can take part in some meeting. Further some clients can provide their own information and be adjusted with some local configurations for their audio and video system. All these operations can be done through the WSDL interface in the client gateway. A client gateway uses XGSP protocol to communicate with other client gateways as well as the session server.

### (2) Media Server

A media server is a RTP Channel for audio and video communication between clients. It can report to the session server about the channel resources such as RTP port number, media codec type and also accept the commands from the session server to bind the RTP channel of some client.

### (3) Session Server

The session server is the core of the XGSP, which can accept request of various clients and organize the videoconference. It can also control the media server to make RTP channel binding.

## 2.2 XGSP: XML-Based General Session Protocol

XGSP is a XML-based general session protocol. It enables WSDL-based collaborating clients to create dynamic groups and join in the groups to share various collaborative capabilities, such as audio, video, whiteboard and so on. The goal of XGSP is to develop a general session layer so that different clients for the same application can interact with each other and different collaboration application can be integrated into the whole system.

There are three important entities in XGSP: session entity, user entity and media entity. The session entity describes the attributes of the session, including the creator of the session, the schedule time for the session, the URL for the some resources of the session and so on. The user entity is used to define a user at the specified location. Its format is [protocol tag]: user @ hostport url-parameters. The media entity represents a media type that the client can support. It includes some codec name and transportation address.

There are four sets of methods in XGSP.

Registration Method	Session Command Method	Query Method
	Session Channel Binding Method	

### (1) Registration Method

Each user can register itself in a registration server with its alias name and current location. The registration method can be used to identify users and support the mobility of the users. A registration

server can be found by manual configuration or multicast.

There are four messages for registration method: Registration Request, Unregistration Request, Login Request and Registration Response. Registration and Unregistration Request are used by a user to make or delete the registration record in the registration server. When a user logs in into the system, it should send a Login Request to the system to activate its registration record.

### **(2) Session Command Method**

The command for the session can be divided into two categories. One is for the membership of the session. The other is for the session control. Membership Control Commands include: Create Session, Invite Into Session, Join Session, Leave Session, Modify Session, Terminate Session and Session Command Response. Session Control Commands include: Source Select Request, Request/Release Chairman, Request/Release/Grant/Cancel Floor.

There are various styles of session: free seminar, chairman-based, lecture-based. Currently different videoconferencing systems support different style of the session. For example, the meeting of Access Grid is always of the free seminar style. On the other hand an H.323 system can support all the styles. In order to make various sessions compatible, we introduce a hybrid session control mode, which means that each client can choose the video and audio streams it is interested in. Further there are two special channels for chairman and speaker in the session, which can be received by all the participants in the session.

Based on the above mechanism, various style of session can be implemented. In a free seminar, the special channels for chairman and speaker will not be established. When a lecture-based session is created, two channels can be created for the teacher client and students clients. XGSP will allow richer floor control and experimentation with further styles.

### **(3) Query Method**

Clients and the session server can use the query method to discover various properties about the system. For example, a client can discover how many sessions are going on. The session server can discover which kind of RTP channels that media server can support.

### **(4) Session Channel Binding Method**

The session server uses this method to bind the RTP channels of a client into the media server. Further the media server can make codec conversion between different clients. In addition, this method can be used to bind the other collaboration applications such as chat and shared display into the Narada[2] or JMS[7] topics of a publish-subscribe message model.

## **3. Applications of the XGSP Web Service Framework**

In the section, we discuss two examples that how to use the A/V web service framework to build some real applications. The first example is how to enable H.323 and SIP clients to join the session of Access Grid. The second example is how to make Access Grid and HearMe system work together.

### **3.1 Adapting various clients in our Prototype System**

At present we are developing a prototype of A/V web-service system that integrates H.323 clients, SIP clients as well as MBONE clients into the whole collaboration system. The architecture of this system is showed in the figure 2.

It consists of three components, a H.323 and SIP signaling gateway, a service session server and a media server. H.323 and SIP signaling gateway can translate the signaling procedure of H.323 and SIP into XGSP methods in our system. The session server accepts the request from the gateway, performing the task of making registration, creating and maintaining session membership, making service

negotiation. The media server can accept the commands from the session server and create the communicate channels among H.323, SIP and Access Grid clients. The media server can support publish-subscribe model for A/V clients, which means a client can subscribe to the audio and video streams via general signaling procedure and the media gateway can create the filter and transcoder that this client wants according to the commands from the session server.

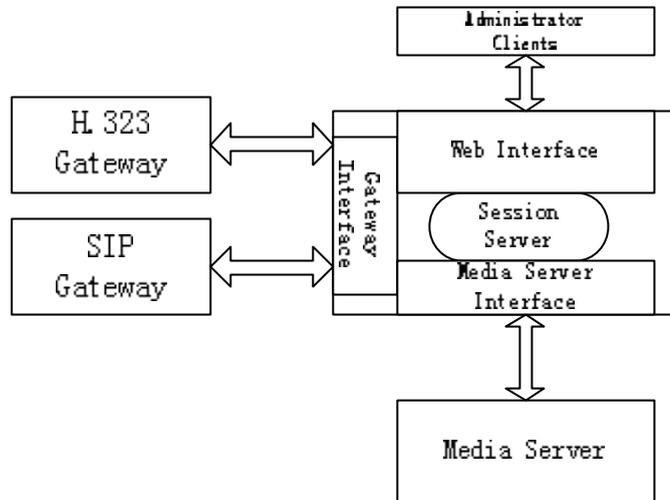
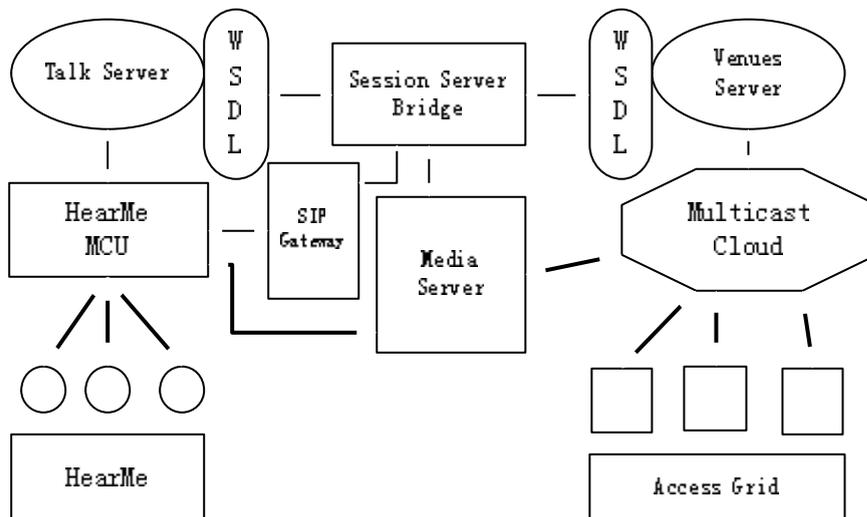


Fig 2: A Prototype which supports H.323, SIP and AG clients

### 3.2 Bridging different collaboration communities

In this example, we discuss how to connect between different collaboration systems using web service technology. HearMe system is an audio system based on SIP. The HearMe Talk Server plays the role of the session server in other systems and the HearMe MCU provides SIP signaling and RTP channels for multipoint meeting. A bridging system could be introduced to connect HearMe and Access Grid together. It is showed in Fig 3.



## 4. Future Work

In addition to audio and video collaboration, there are many other important data sharing tools, such as whiteboard, distributed PowerPoint, shared display and chat. We are planning to integrate these collaborative applications into our prototype. Further as explained in [3], one can design powerful event infrastructure to support communication between different Web Services. This event web service supports routing, filters, and publish-subscribe linkage of clients. As indicated in fig. 1, we will experiment by using messaging services such as JMS or Narada to control the communication channels. The application tools will be built into web-service entities in our system using WSDL. They can use XGSP protocol for session management such as creating, modifying and deleting chat and shared display and whiteboard sessions.

For a large scale heterogeneous conference, there will be an infrastructure of many media servers and here we are planning to use a message system such as Narada to transport audio and video traffic. This system will be optimized for delivering multimedia traffic and Narada supports UDP along with TCP, providing better load balance and audio reliability enhancement over basic communication models.

## 5. Conclusion

In this paper, we present a web service framework for an audio/video collaboration system. In this framework, all the components of videoconferencing system are regarded as web service entities. And they can be coupled together using XML based communication protocol.

Under such a framework, we can implement a more general collaboration system, which can support H.323, SIP, and Access Grid in the same Audio and video collaboration. Thus the framework makes it is easier to organize large scale of collaborations across the different communities based on different collaboration technology.

## 6. References

- [1] Access Grid, <http://www.accessgrid.org>
- [2] Geoffrey C. Fox and Shrideep Pallickara, "The Narada Event Brokering System: Overview and Extensions", To appear in the proceedings of the 2002 International Conference on Parallel and Distributed Processing Techniques and Applications (PDPTA'02)
- [3] Geoffrey Fox, Ozgur Balsoy, Shrideep Pallickara, Ahmet Uyar, Dennis Gannon, and Aleksander Slominski, "Community Grids" invited talk at The 2002 International Conference on Computational Science, April 21 -- 24, 2002 Amsterdam, The Netherlands.
- [4] Handley, M., Crowcroft, J., Bormann, C. and J. Ott, "The Internet Multimedia Conferencing Architecture", Internet Draft, draft-ietf-mmusic-confarch-03.txt, July 2000.
- [5] HearMe Audio conference system , <http://www.hearme.com>
- [6] H.323 ITU Recommendation
- [7] Java Message Service (JMS), <http://java.sun.com/products/jms>
- [8] Real Time Transfer Protocol (RTP), rfc 1889, <http://www.ietf.org/rfc/rfc1889.txt>
- [9] Session Initiation Protocol (SIP), rfc 2543, <http://www.ietf.org/rfc/rfc2543.txt>
- [10] Simple Object Access Protocol (SOAP) 1.1, <http://www.w3.org/TR/SOAP/>
- [11] Web Services Description Language (WSDL) 1.1, <http://www.w3.org/TR/wsdl>