AVATS: Audio-Video and Textual Synchronization

Siddharth Maini, Joshua Rosen, Marlon E. Pierce, Geoffrey C. Fox
Community Grids Laboratory
Indiana University, Bloomington, IN 47404, USA
{ *smaini@indiana.edu, jjrosen@indiana.edu, mpierce@cs.indiana.edu, gcf@indiana.edu* }

ABSTRACT

In this paper, we explain the architecture and provide implementation details of a synchronous collaborative tool we created using a lightweight Web 2.0 mash-up development methodology. This approach allowed us to put together a client interface with reasonable capabilities, scalability, and robustness rapidly. The rapid development of this tool also allowed us to perform thorough testing and incorporation of the system in a live, production environment in a relatively short amount of time.

KEYWORDS: Synchronous, Collaboration, Web 2.0, Discussion, Blogs, Audio-Video.

1. INTRODUCTION

Web 2.0 methodologies [1] are driving a new trend of development of applications such as mash-ups, gadgets, and social networking applications. This has revolutionized the use of the Internet by enabling collaboration and building social networks that can scale into millions of users while reducing the gap between developers and users [2]. Web 2.0 technologies and methodologies are now the common state of practice. One of Web 2.0's principle objectives is to let the users publish their own data, share it, and control it in meaningful ways [1].

Applications based on Web 2.0 methods encourage open and social participation and take full advantage of the vast reach of the Internet amongst millions of global users [3]. Architecturally, Web 2.0 is built from open architecture network services that provide typically REST-style interfaces and often use RSS and ATOM as content syndication formats in a manner analogous to WSDL and SOAP in heavier weight enterprise developments [4][5]. Web 2.0 websites or applications often feature a rich user interface based on AJAX, Flex and other rich media [1] for interacting with these backend services. Social networking is one of the most common applications of Web 2.0, creating shared content among well-defined user groups. The open programming interfaces and service infrastructure make it simple to build these and other Web 2.0 sites out of reusable pieces, known as "mash-ups".

The concept of Web 2.0 makes it very attractive for the research and academic community [3]. Much research and development work is being done to extend social networking and Web 2.0 architectures to achieve synchronous collaboration. Popular examples of asynchronous collaboration include YouTube, Twitter, Connotea, Delicious, CiteUlike etc [6][7][8][9][10].

Social networking and more generally Web 2.0 focus on asynchronous collaboration. This is driven in large part by the simplicity of the most common Web protocols and the well-known difficulties of building a reliable synchronous, real-time collaboration system [11]. The focus of this paper is on using Web 2.0 for synchronous collaboration rather than asynchronous collaboration. Our specific research question is to determine if it is possible to build a robust, collaborative "mash-up" out of reusable client parts and standard services.

Currently, the asynchronous collaboration paradigm dominates most Web 2.0 applications. Among other reasons, this is partly due to the human usage pattern, which would make it difficult to have all the participants online simultaneously. This is most evident in international collaborative efforts, as the time difference between the various participants could pose further complications for synchronization. Examples of this asynchronous collaboration concept include social networking websites such as Facebook, MySpace, and LinkedIn.

Although asynchronous collaboration still dominates for the most part, there have been significant efforts in integrating some synchronous components, such as Facebook's Online Friends and Chat feature [12]. There

are also many purely synchronous applications available such as AOL Instant Messenger, IRC, Skype, and NetMeeting. These applications support a text-based form of synchronous collaboration. A few of these have a video chatting capability but rely on using their own integrated video streaming solution whereas our mash-up tool can be used to stream videos from any source producing a flash video stream. In this paper we have focused on using Web 2.0 mash-up techniques for building new user interfaces. We have assumed that the infrastructure to support the audio-video synchronous collaboration is available for our use. It is out of scope of this paper to describe various algorithms or inner workings of how audio-video stream are managed. The outcome of our methodology was a website which supports Audio/Video and textual forms of synchronous collaboration.

Audio-Video and Textual Synchronization (AVATS) is a synchronous collaboration tool created by the developers at the Pervasive Technology Institute [13] initially in order to support the 4th IEEE eScience Conference [14][15]. The main goal of the tool was to be able to stream the entire conference live and allowing online viewers to discuss the individual events and ask direct questions to the presenters during on-going presentations. This has allowed us to achieve a different method of synchronous collaboration, which at present, is confined only to hosting live conferencing events.

In the following sections, we will explain the architecture and implementation of AVATS, and include an evaluation of its performance after the conference. We will also discuss other synchronous collaborative tools, such as SCIVEE, WebEx, and ORLIVE.com. Finally, in the conclusion, we will discuss the future of AVATS and how it can be scaled for other uses, such as distance collaborative learning.

2. RELATED WORK

Collaboration on the Web, both synchronous and asynchronous, is not new. Early efforts in synchronous collaboration include Tango [16], the Access Grid [17], GlobalMMCS [18], and others. These efforts were typically marked to be sophisticated, end-to-end solutions for difficult problems in collaboration such as buffering, time ordering of events, multicast networking, application software-level stream control, and similar research issues.

Many primarily asynchronous collaboration systems have also been developed. Open source examples include Sakai, Drupal, and Joomla. These are not purely asynchronous, as they may include chats, but they tend to focus on asynchronous content management. Learning management systems are an important related case [19][20]. In contrast to both of these efforts, we are investigating a much lighter-weight solution. We focus only on the client, built as a mash-up.

The concept of Web 2.0 for collaboration has been implemented for a long time. Historically, a lot research work has been done on Audio-Video systems including EVO (Enabling Virtual Organizations), VRVS (Virtual Room Videoconferencing System), TANGO and similar [21][22][16]. Zhai *et.al* designed eSports which was a collaborative and a synchronous video annotation system based on grid computing platform [18]. The system was capable of playing and archiving live video, taking snapshots of the video with an option to annotate the video stream using a whiteboard and replay it with annotations [18]. eSports was meant to support Computer Supported Cooperative Work (CSCW) using a distance learning approach. In addition, synchronous collaborative projects such as the Global-MMCS were aimed to build a collaboration system and provide services including video-conferencing, instant messaging and streaming support while integrating the Access Grid [23]. It should be important to point out that these systems supported end-to-end synchronous collaborations and were a result of a huge amount of research effort in the field of networking. We developed AVATS as a client-end interface in a time-period of three months from start to end which included a testing and evaluation period without worrying about the underlying networking infrastructure.

2.1. Collaborative Environments

Collaborative environments can be divided into three categories: *Synchronous, Asynchronous, and full-function collaborative environments* [24].

Synchronous Collaboration tools enable users to interact in a virtual environment regardless of physical location in real-time, such as AIM, NetMeeting, and IRC. Asynchronous Collaboration enables users to interact with each other regardless of physical location or time. Examples: Facebook, MySpace, WebCT, and Sakai. Full-function Collaborative Environments meld the concepts of synchronous and asynchronous collaboration. Facebook with Online Friends and chat is an example.

2.2. Evaluation of some Web 2.0 Collaborative Tools

SciVee claims to be the first Web 2.0 website to support research collaboration between the researchers in the academic world [25]. SciVee is meant to support research collaboration using recorded video and

presentations. While it allowed a user to view a video presentation, it is still an asynchronous collaborative tool, as these videos are pre-recorded, and no actual interaction is possible. WebEx has a comprehensive video conferencing solution, but it is geared towards meeting business requirements in a private environment.

A recent example of synchronous collaboration can be cited on the day of President's Obama Inauguration between CNN and Facebook. Several FB users were able to comment live on the event while the inauguration was being streamed.

AVATS is a different from the above tools because of its ability to stream live and archived videos, as well has its live discussion capability. This discussion capability allows the online viewers to interact not only with each other, but with the presenter in real-time. This allows the users to interact with each other and ask questions to the presenter either during the presentation or after it. It should be noted that the questions were conveyed to the presenters by a chat monitor in the presentation room. AVATS also includes a blog interface which will give the viewers some background information before the presentation starts. This essentially makes AVATS a mash-up tool where the video, chat, and the blog components are using together while having different data sources.

CNN.com's Live with Facebook interface allowed for a live chat through Facebook status updates. ORLive.com only allows asynchronous interaction between the doctors in the operating room and the users interested in asking questions either before or after, but not during the live webcast. WebEx includes many synchronous features, but is designed for business use. The presentation features are designed for private meetings, instead of open, public presentations.

Table 1. A Comparison of Web 2.0 Synchronous Collaboration Systems

Tech	**Description**	**Comparison**
SciVee	Combines a scientific article with a video presentation [25].	SciVee only has recorded videos and does not have live chat. Both provide video presentations combined with a relevant information section.
WebEx	Creates web conferencing and webinars.	Both allow live and archive video presentation with live chat available.
ORLive	Streams live video from operating rooms combined with chat [26].	Both use live video presentation combined with text chat to allow synchronous collaboration
CNN's Live	Shows a live stream with interaction with others over Facebook [27].	Both have a live video feed w/ chat available for viewer discussion.

3. AVATS ARCHITECTURE

The goal of the AVATS project was to use common Web 2.0 technologies and reusable client libraries create a user interface mash up that would allow online viewers to participate in the 4th IEEE eScience Conference. The interface is shown in Figue1. Based on the conference organizers' requirements, the user interface has three capabilities. First, the interface had to be able to both stream the presentation live, and have an archive available for later use. Second, a section was required to display the author's blog, which contained the paper abstract and any other updated information the authors wished to provide. Lastly, we provided a real-time discussion feature to facilitate communication among the users and the presenter.

The live video stream and archives were provided by the video conferencing staff at Indiana University. The chat section utilized the ActiveMQ message broker and the STOMP protocol. Each presentation had a distinct chat session associated with it. The blog section was designed to import and view any RSS feed. Our system supported five live video streams simultaneously and was designed to limit the number of online viewers for any individual presentation. This limit was set to a maximum of a thousand users per presentation.

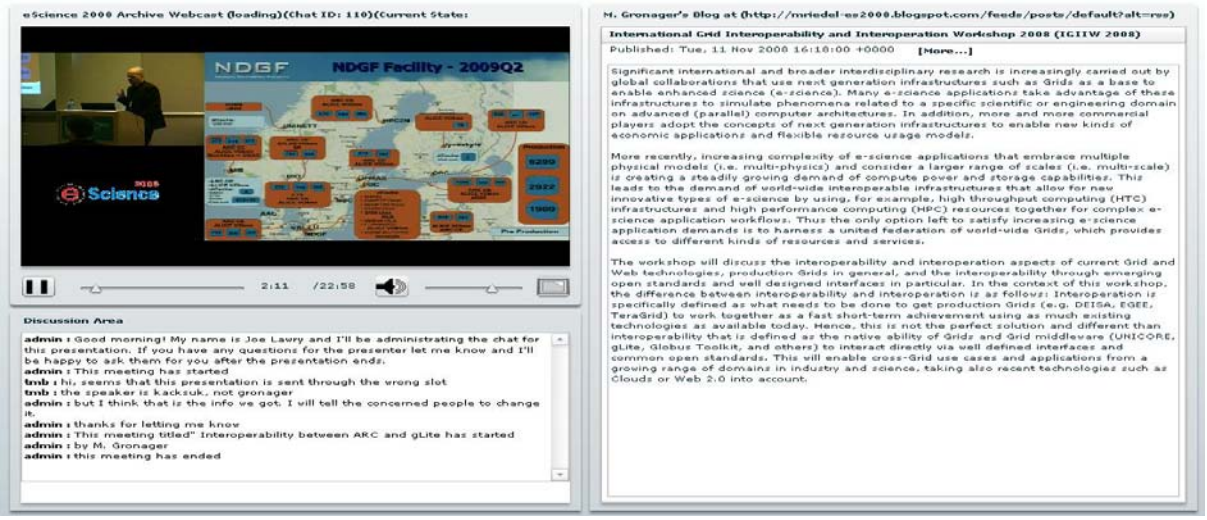

Figure 1. AVATS User Interface showing Video (upper left), Discussion Area (lower left), and Blog (right). The Video Stream includes both the presenter and the slides

3.1. Technologies Used

We summarize the technologies used to build AVATS in Table 2.

Table 2. AVATS Technology Summary

Technology	Purpose
Tandberg Edge 95	Capture a live/video stream combining it with a presentation video and sending it over in H.239 format
Flash Media Encoder and Flash Media Server	Server streams a live video stream encoded by the Flash Media Encoder
Tandberg Codian IPVCR 2200	Takes in the H.239 communication and records it in Codian format.
Codian Converter	Converts the Codian format into FLV, MPEG, and/or WMV format
PHP/MySQL Database	Store the information for each presentation and created RSS/XML documents based on that information
BlogSpot	Used to create blogs for each presentation and export it in RSS format. Around eighty blogs were created.
Adobe Flex	Used to make the entire front-end for AVATS
ActiveMQ	Messaging Server used to relay chat communications

4. AVATS IMPLEMENTATION

4.1. Application Interface Components

Adobe Flex was used to create the user interface due to its ability to interact with Web 2.0 technologies (like RSS/XML) and its modular nature. This allows each portion to be developed individually as components, which allowed them to be upgraded and debugged individually. We used Adobe Flex to developed web applications based on Flash. Flex is designed for use by programmers interested in developing flash applications, rather than designers.

Figure 3 below shows the diagram of how the application interface works. As mentioned above, the interface was divided into three different components namely a *videoplayer component, a chat component, and a blog component.* The main flex application contains an HTTP Service which receives an XML document created using PHP containing the information about the presentations as shown in Figure 3.

The main application grabs the meeting ID from the URL for each presentation and makes an HTTP Service call to retrieve the presentation information as an XML (Figure 3). It passes the information to its components. The videoplayer component uses either the "*streamURL*" or the "*archiveURL*" elements depending on the "*status*" (Table 3). We describe "*status*" numbers in Table 1. The chat component uses the meeting ID to create a connection to the ActiveMQ message broker service which creates a unique chat topic depending upon the meetingID. It also calls an HTTP Service to retrieve the existing chat messages. The blog component

has a HTTP Service that uses the *"blogURL"* element to import, parse, and display the RSS Feed for that particular meeting. The main application includes a timer running in the background that updates the status depending on the schedule start and end times of individual presentations and the availability of the archive videos. The blog component is a simple accordion component [28] that shows a specified number of entries from the related blog.

Lastly, the chat component allows a user to log in with any username, as long as it is not being used, and discuss the presentation while it is live. After the conclusion of the presentation, the discussion is still visible as a log. The videoplayer component allows the users to change the volume, pauses the presentation, and use the slider to browse through the archive video. It also features a Zoom-In button which allows the video to be viewed in a larger window while removing the blog component.

Table 3. Table Describing Live/Archive Status Numbers passed to the Video Component in Figure 3

Status	Explanation
1	Presentation has not started yet
2	Presentation is currently live. Show *streamURL*
3	Presentation has ended and no archive is available for that meeting ID
4	Presentation archive is available. Show *archiveURL*

```
<results>
    <meeting>
        <ID>10</ID>
        <name>KEYNOTE by Rich Wolski</name>
        <speaker>Rich Wolski</speaker>
        <blogURL>http://rwolski-es2008.blogspot.com/feeds/posts/default?alt=rss</blogURL>
        <streamURL>rtmp://flashstream.indiana.edu/live/escience5</streamURL>
        <status>4</status>
        <timer>0</timer>
        <date>2008/12/10</date>
        <from>0900</from>
        <till>1000</till>
        <archiveURL>http://stream.escience2008.iu.edu/archives/Archive10.flv</archiveURL>
        <slidesURL></slidesURL>
        <presentationURL>http://live.escience2008.iu.edu/slides/slides10.ppt</presentationURL>
    </meeting>
```

Figure 2. Example XML containing Presentation Information

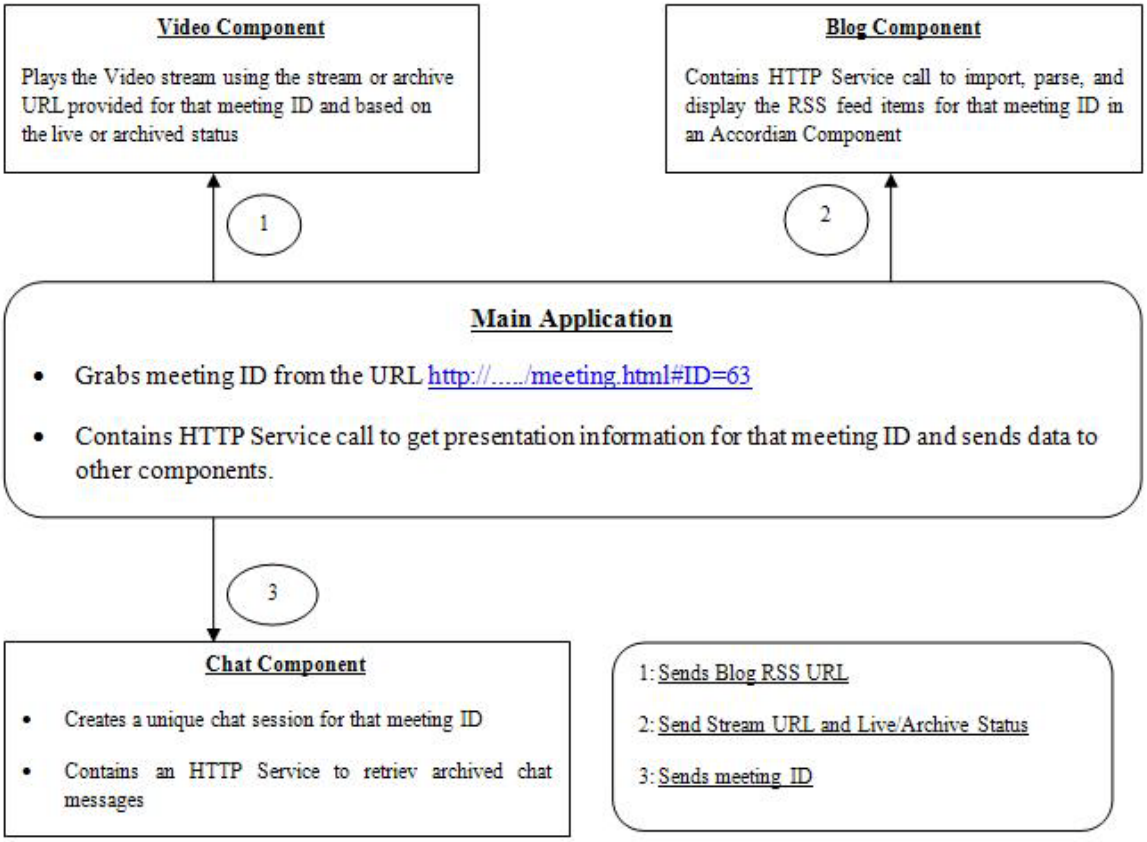

Figure 3. AVATS Application Architecture

In the backend, information about each presentation was stored in a MySQL database. The '*Meeting*' database table (Table 4) contains the information about each of the presentations, including scheduled times, presentation name, and the name of the presenters. It also includes the URLs for the blog and archive video (if available). Lastly it references the *room* table (Table 5), which stores the live stream URL. The name of the presentation, name of the presenter, and times are stored, as well as the room (which provided the live stream URL), and blog URL. Using PHP, XML files are created to be sent to the Flex front end.

Table 4. The 'Meeting' Table in the MySQL Database. This is the general metadata about each presentation.

Field	Description
ID	The ID number of this presentation
Name	The Name of this presentation
RoomID	The ID from the room table for
Start	Scheduled Start Time
End	Scheduled End Time
Speaker	Name of the presenter
BlogLink	URL of the blog RSS feed
ArchiveURL	URL of the flash archive file
SlidesURL	URL of the presenter's slides

Table 5. The 'Room' Table in the MySQL Database. This is stream-specific Information.

Field	Description
ID	The ID number of this room
Name	The name of this room
streamURL	URL of the live stream coming from

Each talk has a separate presentation interface (Figure 1) in our system. To organize these, particularly for a large meeting, we developed **a calendar interface** (Figure 4), also using Adobe Flex, to show the scheduled presentations on a day to day basis, with links to the meeting interface for that presentation, and buttons to

add that meeting's URL to Google Calendar. The calendar interface used an RSS feed of the meetings to display the calendar of events.

An **administration application interface** (Figure 5), developed using Adobe Flex, allowed the administrators to monitor and administrate the discussion, so they could pass along any questions to the presenter. They were given additional chat functionality, such as a banning feature. This interface also provided the various administrators to interact with each other, a window with the streaming video, and the ability to change the scheduled start and end times.

4.2. Video Streaming and Recording Process

According to James McGookey from Indiana University, we used Tandberg Edge 95 endpoints to send an Audio/Video H.239 Dual Stream to a Tandberg/Codian Multipoint Control Unit (MCU). From there a stream was sent to another videoconferencing endpoint connected to a PC with audio/video capture cards. Using Adobe's Flash Media Encoder on the PC and Flash Media Server, we were able to send the live stream to our viewers. The MCU also sent a stream to a Codian/Tandberg IPVCR conference recorder. This device is what allowed us to create derivative transcodes for later, on-demand use. In the presentation room, a Tandberg system was connected to a camera showing the presenter and a set of connections for audio and video from a laptop.

Though this process (as shown in Figure 6) worked well for the eScience Conference, this configuration is not required for the AVATS system. A variety of other processes could be used, as long as a flash stream is available for the live presentation, and a flash format archive file is used.

5. EVALUATION

Following our general "mash-up" strategy, we were able to develop the user interface and backing components in approximately 10 weeks. This period included several live tests before the e-Science conference. These tests exposed several problems in the basic system, which we describe below. Following the conference, we also took a voluntary survey of the attendees.

5.1. Testing

There were several rounds of testing performed. Initially, each component was tested individually. Once the separate components appeared to be working correctly, testing of the integrated system began. Test presentations were performed weekly using the AVATS system for the entire month of October and November 2008. Bugs were reported by the test users via the chat system. We used the same video streaming process that was set up for the 4th IEEE eScience conference.

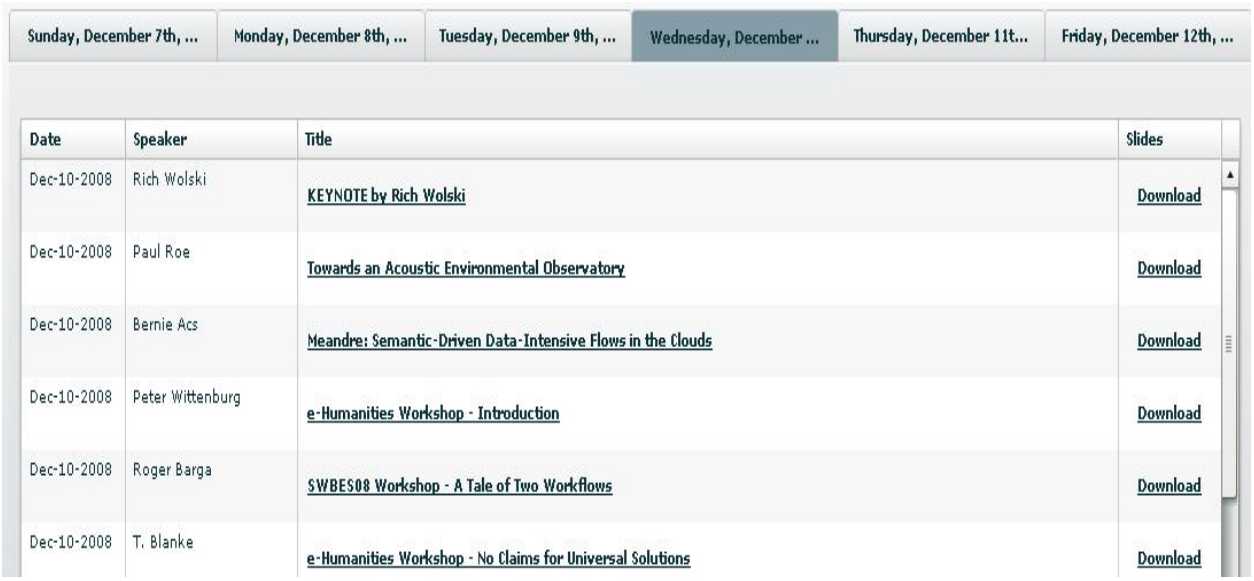

Figure 4. Calendar Interface displaying Schedule of Events. Clicking the "Title" link takes the User to the Session Interface (Figure 4). The "Slides" link is a simple download link for presentation.

Figure 5. Administration Interface for a Session. See text for a full description.

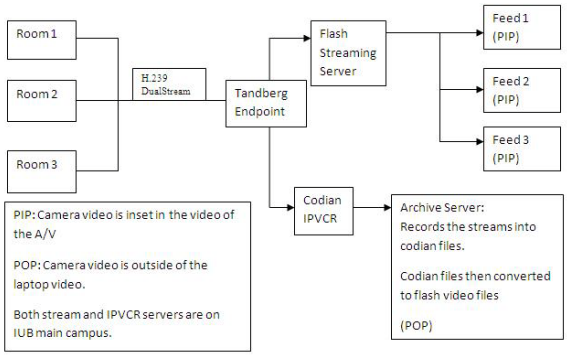

Figure 6. Video Streaming and Recording Process

Table 6. Bugs Found in Testing and Their Resolutions.

Bug	Resolution
The chat component was receiving chats from all presentations.	The ID of the presentation was appended to the chat messages, so that all other chats were ignored.
Chat was subject to denial of service: a very large (megabyte) cut-and-paste froze the system.	A limit was placed on how many characters could be typed into the chat.
The video became fuzzy and an echo in the audio came up as the number of viewers increased. This bug seemed to have occurred mostly in Internet Explorer browsers.	This could not be resolved as this was caused by the flash streaming server. The prevalence of this issue may differ depending on server, number of users, or the transfer protocol used.
If there was no blog associated with a presentation, it gave an error.	A default blog was set up.
The user would get logged out of the discussion session when switching between zoom-in and normal modes	The zoom-in functionality was reworked so that the issue was resolved.
We were unable to find an upper bound on the number of users.	We placed adjustable participant limits (or something)

5.2. Survey Results

After the conference a survey about the conference was created using SurveyMonkey and emailed to all the attendees. The survey included several questions related to the Visual Clarity, Appropriate Functionality, and Usability of the AVATS. In total we received twenty-seven responses.

The calendar interface was considered somewhat/very easy to read by 62.5% of the respondents. 72.2% of the respondents seemed to be moderately/very satisfied with the functionality of the calendar with one respondent commenting about the Google Calendar button being a nice feature.

65% of the respondents found the video interface very/somewhat easy o to use with one person commenting about having problems with the live video stream. 50% of the respondents were very/moderately satisfied with the video interface with 45% of them having a neutral opinion about it. 66% of the respondents did not have any problems with the quality of the video and sound. However, one comment mentioned the prevalent echo problem while another mentioned the Picture In Picture (PIP) feature, which tended to block certain portions of the presentation. Also, 83.3% of the respondents did not find any problems with the usability of the video streaming interface. 78.6% of the respondents did not have any problems with using the chat/discussion area.

Overall, we found that 60.8% of the respondents were moderately/very satisfied with the system's visual clarity. In addition, 84.2% of the respondents did not find it confusing to use with 88.2% mentioning that they liked AVATS. It was worth mentioning that many respondents thought it was important to have archived videos.

5.3. Post-Conference Site Analysis

After the conference concluded, Google Analytics was used to analyze the site traffic information for AVATS. Google Analytics is a free tool used to gather information about website traffic. It gives website owners an ability to track and show various visitor-related trends in addition to other site-related information [29]. Figure 7 contains a snapshot of geographical Map Overlay of the site.

826 visits came from 32 countries/territories

Site Usage

Visits	Pages/Visit	Avg. Time on Site	% New Visits	Bounce Rate
826 % of Site Total: 100.00%	2.60 Site Avg: 2.60 (0.00%)	00:03:36 Site Avg: 00:03:36 (0.00%)	50.36% Site Avg: 50.36% (0.00%)	61.50% Site Avg: 61.50% (0.00%)

Country/Territory	Visits	Visits
United States	385	46.61%
United Kingdom	103	12.47%
Germany	66	7.99%
South Korea	36	4.36%
Australia	33	4.00%
Brazil	31	3.75%
Belgium	22	2.66%
China	21	2.54%
Austria	18	2.18%
Netherlands	18	2.18%

Visits

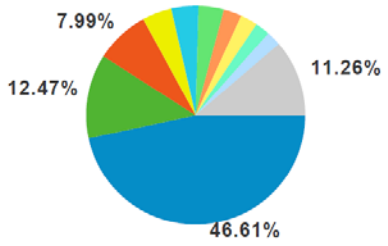

Figure 7. Map Overlay Using Google Analytics

Overall we found that there were a total of 826 visits coming from 32 countries with 418 unique visitors. Average time on the site was approximately 4 minutes with almost 47% of hits coming from within the United States of America. In addition, we found that only 23.85% of the traffic was direct and 76.03% of the total traffic was through referring websites (please refer to Figure 8). It was also interesting to observe that almost 60% of the visitors were using Firefox as their web browser.

Sources	Visits	% visits
escience2008.iu.edu (referral)	521	63.08%
(direct) ((none))	197	23.85%
biocep-distrib.r-forge.r-	34	4.12%
pegasus.isi.edu (referral)	15	1.82%
goc.pragma-grid.net (referral)	10	1.21%

Figure 8: Top Traffic Sources

6. CONCLUSIONS AND FUTURE WORK

AVATS is a synchronous collaboration tool which can be built within a short span of time as long as you have the underlying services or infrastructure required to support its various capabilities. It was important for us to do a lot of user testing to make sure that the hardware infrastructure would be able to support live streaming video and chatting services. Along this process, we tried to fix as many bugs as possible. We used the tool for the first time during the 4th IEEE e-Science Conference while receiving a positive feedback. AVATS can be classified as a mash-up tool because the data is imported into each of the components from different sources. This was an interesting idea to implement for a start. One can use the system for many different purposes after upgrading it by adding more features etc.

Further upgrades to the AVATS system such as an administration interface are planned. This interface will allow the organizers for any conference to integrate the schedule for their conferences automatically. There are also plans to develop a subset of AVATS into an iPhone application.

AVATS may also be implemented in a distant educational setting. Courses could be streamed live to the students who can interact with each other and the lecturer, and be archived for later use. Much more comprehensive instant messaging features can be added to AVATS depending on the user requirements.

REFERENCES

[1] "Web 2.0," Wikipedia, the Free Encyclopedia.

[2] M. Pierce, G. Fox, J. Choi, Z. Guo, X. Gao, and Y. Ma, "Using Web 2.0 for Scientific Applications and Scientific Communities."

[3] C. Ullrich, K. Borau, H. Luo, X. Tan, L. Shen, and R. Shen, “Why Web 2.0 is Good for Learning and for Research: Principles and Prototypes.”

[4] “Representational State Transfer,” Wikipedia, the free encyclopedia.

[5] “SOAP,” Wikipedia, The Free Encyclopedia.

[6] “Delicious,” Delicious.

[7] “CiteULike: Everyone's library,” CiteULike: Everyone's library.

[8] “Connotea: free online reference management for clinicians and scientists,” Connotea: free online reference management for clinicians and scientists.

[9] “Twitter: What are you doing?,” Twitter: What are you doing?

[10] “YouTube - Broadcast Yourself.,” YouTube - Broadcast Yourself.

[11] G. Fox, W. Wu, A. Uyar, H. Bulut, and S. Pallickara, “A Web Services Framework for Collaboration and Videoconferencing.”

[12] J. Wiseman, “Facebook Chat: Now We're Talking | Facebook,” Apr. 2008.

[13] “Pervasive Technology Labs at Indiana University.”

[14] “4th IEEE International Conference on e-Science, 2008: Indiana University.”

[15] “4th IEEE eScience Conference: Event Calendar,” 4th IEEE eScience Conference.

[16] Lukasz Beca, Gang Cheng, Geoffrey C. Fox, et al., "TANGO - a Collaborative Environment for the World-Wide Web," April 15, 2001

[17] M. Wang, G. Fox, and S. Pallickara, “A Demonstration of Collaborative Web Services and Peer-to-Peer Grids,” Proceedings of the International Conference on Information Technology: Coding and Computing (ITCC'04) Volume 2 - Volume 2, IEEE Computer Society, 2004, p. 62.

[18] G. Zhai, G. Fox, M. Pierce, W. Wu, and H. Bulut, “eSports: collaborative and synchronous video annotation system in grid computing environment,” Dec. 2005.

[19] T. Martin-Blas and A. Serrano-Fernandez, “The Role of New Technologies in the Learning Process: Moodle as a Teaching Tool in Physics,” Computers & Education, vol. 52, Jan. 2009, pp. 44, 35.

[20] B. Costello, R. Lenholt, and J. Stryker, “Using Blackboard in Library Instruction: Addressing the Learning Styles of Generations X and Y,” The Journal of Academic Librarianship, vol. 30, Nov. 2004, pp. 460, 452.

[21] P. Galvez, H. Newman, "EVO (Enabling Virtual Organizations), the Next Generation Grid-enable Collaborative", In proceedings of Computing for High Energy Physics, Victoria BC, Canada, September 2-7, 2007

[22] P. Galvez, H. Newman, "From VRVS to EVO, the Next Generation Grid-enable Collaborative System", In proceedings from Computing for High Energy Physics, Mumbai, India, February 13-17, 2006

[23] T. Huang, S. Pallickara, and G. Fox, A Distributed Collaboration Framework for Stream Annotation, 2009.

[24] C. Qu and W. Nejdl, “Constructing a web-based asynchronous and synchronous collaboration environment using WebDAV and Lotus Sametime,” Oct. 2001.

[25] “About | SciVee,” Scivee. http://www.scivee.tv/about

[26] “About OR-Live, Inc.,” OR-Live: Online Surgical and Healthcare Video and Webcasts.

[27] “CNN, Facebook win presidential inauguration live-streaming contest (kinda) » VentureBeat.”

[28] “Accordion - Adobe® Flex™ 2 Language Reference.”

[29] “Google Analytics,” Wikipedia, the free encyclopedia.

[30] S. Colter and G. Phipps, “A Case Study for Evaluating a New Technology: Selecting a Synchronous Collaboration Tool at the University of Tennessee | EDUCAUSE CONNECT,” Jun. 2006.

[31] Bray, Tim; Jean Paoli, C. M. Sperberg-McQueen, Eve Maler, François Yergeau, "Extensible Markup Language (XML) 1.0 (Fourth Edition)-Origin and Goals," September 2006

[32] “About Us. The company, founders, Broadcasters. John Ham, Brad Hunstable, Dr. Gyula Feher. Investors, Venture Partners, DCM, Western Technology,” Ustream.TV.

[33] “Asynchronous Learning,” Wikipedia, The Free Encyclopedia.

[34] L. Guterman, “Specially Made for Science: Researchers Develop Online Tools for Collaborations - Chronicle.com,” The Chronicle of Higher Learnin, vol. 54, Aug. 1922, p. A9.

[35] D. Edelson, D. Gordin, and B. Clark, “The Global Warming Visualizer: A Scientific Visualization Environment for Education,” Jul. 1996.

[36] “Top 10 Web 2.0 Winners,” Wisdump.

[37] “Web 2.0, mashups and social networking - what is it all about?,” TamingTheBeast.net.