

ROUGH EDITED COPY

PEN-INTERNATIONAL

"VIDEOCONFERENCING USING OPEN-SOURCE: BUILDING A PROOF-OF-CONCEPT PLATFORM FOR COMMUNICATIONS AND DISTANCE LEARNING"

PRESENTED BY: SIMON TING, ZACH SZAFRAN, CATHY CLARKE, REGINA KIPERMAN-KISELGOF

JUNE 25, 2008

1:00 PM ET

ON-SITE CART PROVIDED BY:  
ALTERNATIVE COMMUNICATION SERVICES, LLC  
PO BOX 278  
LOMBARD, IL 60148  
800-335-0911

\* \* \* \* \*

This is being provided in a rough-draft format. Communication Access Realtime Translation (CART) is provided in order to facilitate communication accessibility and may not be a totally verbatim record of the proceedings

\* \* \* \* \*

>> SIMON TING: I will start the presentation now if everybody is ready. Video conferencing is a business tool. It allows workers from different geographic locations to meet and collaborate using video. It looks like a good educational tool, but the high cost of commercial systems discourages experimentation. So we want to make it possible for people to be able to utilize a low-cost system using open-source software.

We wanted this to work in the web browser utilizing Web 2.0 applications instead of people using desktop software.

So we begin by looking at some of our available options. We have two main categories. One is consumer products, and the other is commercial products.

Commercial products include instant messaging, for example, Yahoo! Messenger, iChat, Skype, and so forth. They allow two people to connect and talk to each other using video chat.

Some of the newer ones allow three or four videos at the same time. But it's hard to accommodate a large group of people using this technology.

Commercial software is very expensive. It will accommodate a larger number of people, but it's very expensive. You may have to pay a monthly subscription fee and buy and setup your own server.

There's a steep learning curve for people who use this conference technology once in awhile. There's a lot to learn. If you want a simpler system, you can just click and start a conference very easily.

I'd like to give you a brief, non-technical description of the project that we've been working on. Building a video conference system is a very technical topic. I would like to talk about the project a little bit in non-technical terms and give you an overall view.

We took a hybrid approach to this system. Video conferencing tools tend to be written using Java programming language. Flash is also a very popular platform for this kind of project, for video conferencing systems.

When we were looking for open-source software components, we used both Flash-based and Java-based programming languages. We want to have a different approach where we are different from other video conferencing

systems that are out there. We take open-source Java, or open-source components and we use HTML, CSS, and JavaScript in order to build the framework to make those components work together. You can see this in the user interface that the participants see.

Also most video conferencing systems are developed for business purposes. What we do is take our course management systems using our IdeaTools, the software that we've developed and we incorporate multimedia and video conferencing together with the open-source software. So with our system we have more integration in our course management system.

Most course management systems already have support functions where the instructor can create design courses, create group projects, et cetera. So by building our conferencing system on IdeaTools, we're able to take advantage of all of these various functionalities.

That's the end of my introduction. Next, Regina will show you how we plan for the project, and how we decided what features that we would support and include in our system.

Cathy will show you a simple way of building a videophone that can work in your web browser. It shows you how we can combine HTML, XML, and Java and so forth. Zach will describe the most technical part of this project. And in the end we will give a demo of the system that we built.

>> REGINA KIPERMAN-KISELGOF: I would like to talk about the traditional way of -- I'm sorry.

I would like to talk briefly about the traditional ways of integrating students and the interaction between students and teachers.

Normally a professor in the traditional model of teaching will stand in the front of the room, and they'll lecture to the students, and the students will be sitting in rows and writing notes while the teacher is lecturing. The teacher may require some teacher aids such as PowerPoints, overhead projections during their discussion with the student. They'll use aids or PowerPoints. They also may be writing things down on a white board to help brainstorm or bring up important points during the lecture, and the students will look at the white board as well as the teacher.

Also, you know, they may be drawing some pictures or charts and graphs and so forth. In this model the interaction between the teacher and the students, the teacher can ask the students questions in the classroom to make sure that they are comprehending the lecture. Or the students may ask questions of the professor.

Or they may break into group discussion. That's the traditional model for teaching.

Now we're setting up a model on the computer where a teacher can use Sign Language, or the teacher can use visual aids if the students don't know Sign Language. The same kind of interaction is possible. The teacher will sign a lecture to the students, and the students can discuss the lecture using Sign Language.

The teacher may need to have some kind of aid such as captioning, realtime captioning is an option. Those are traditional ways of teaching in the classroom with Sign Language.

So we wondered if video conferencing can provide a kind of communication where all of this is possible. Can video conferencing encompass the traditional modes of education that we're used to? Let me hand this over to Cathy so that she can talk more about video conferencing.

>> SIMON TING: Okay. Based on what we just discussed about the traditional classroom, we came up with a list of options that we would like to include. These are features that we wanted to include. They're part of most commercial video conferencing products. With multi-point video we wanted to be able to shoot video using chat. We want students to be able to use Instant Messenger chat utilizing video. Also file sharing is the idea of

people working together need to be able to share files. The white board, that's what teachers use in the classroom, and we want students to be able to use PowerPoints in a video conference, or for teachers to be able to write on a screen their notes, in collaboration with people working together and being able to share files on remote desktops, participants can take turn editing documents on the computer.

Also recording and archiving, after we have a video conference we want to be able to save it so that students can review it at a later date.

As far as support for Deaf participants, we want to incorporate support professionals such as Sign Language interpreters, note-taker, and realtime captioning specialists, although we were only designing a prototype, we wanted the roles of these professionals to be recognized early so that it will be easy for these professionals to support and use our system. For example, sharing files with interpreters and captioning specialists so that they can prepare for their assignments.

We also came up with some system and user management requirements such as scalability which means we need to be able to support a large group of users, multiple levels of security were required. We want the overall system security where the users must log into the system. Once they are inside the system, we want students to be able to have more control to stay up with their courses, but we need to limit the participants to their own roles. If somebody is working on a group project, you want only the participants in that group to be able to access that information.

And then integration with course management systems, we want this to be easier than commercial products. We want to take advantage of all of the functions possible so professors can use multimedia such as homework, quizzes, and handouts during a video conference.

Now Cathy will give you -- oh, no, now Zach. Zach will describe the open-source programs that we use in our system.

>> ZACH SZAFRAN: Red 5 is a free and open-source software package developed in the Java programming language. Its purpose is to support the communication between Adobe Flash applications.

Currently it supports stream broadcasting, audio and video streaming, object sharing, as well as stream recording. The goal of the developers was to create and share free and open-source alternative to the high-priced proprietary software Flash Media Server.

Though Red 5 is currently in the alpha stages, and it does not fully support all of the features of Flash Media Server, we were able to implement its current functionality into a well-working video conference application.

The developers included multiple demo applications in their distributed software package. These two applications are broken down into two groups, a server side group where the application is run on the server, and a client-side group where the applications are run in the browser of the user's computer.

Two of these client-side applications, Publisher and Recorder met our needs perfectly. From the source code we were able to develop a flash -- Publisher and Recorder, and from the source code we developed a Flash component for the video conferencing web application.

What this component does that we create is it takes your own video from your webcam and your own audio from your microphone and it sends it out to anyone currently using that component. And it builds a grid of videos where you can basically see everyone that's currently viewing the component and hear them talk.

The online open-source Flash conference also known as OFLA demo application -- OFLA, was also included as a demo application, and without modifying the source code we were able to implement this as our server application.

The server -- the OFLA application currently acts as the middleman between our client-side applications. What Red 5 does is just makes it possible for any user with a browser and a Adobe Flash plug-in to interact and see video and audio from anyone else that has the same ability.

JW Media Player is a complete and fully functional web-based software component that allows the playing of audio and video. It's also built in Adobe Flash, and it allows the playing of audio and video streams from such services like a Red 5 server.

The author released this under a creative commons license which allows us to utilize it and modify the action script source code to use it for non-commercial.

The author provided a heavily-documented API or application programming interface which allows us to make third-party applications that can actually control and manage the JW Media Player.

Another point of interest the JW Media Player is its excellent support for video captions. This is of interest to us for the support of communication accessibility to video information.

Currently we implement the JW Media Player as a viewer for our archived videos for each remote video conference session.

And TightVNC is a free remote control software package that is derived from the virtual network computing software, also known as VNC. What this basically does is it allows someone to see a remote computer's desktop and control its mouse and its keyboard from your mouse and your keyboard. This software was released as a free GPL license package, and the full source code was included for us to use. We did not fully implement this application into our video conference web application, but we did take advantage of the built-in browser-based viewer that the VNC or TightVNC server had to offer.

Our application is pre-constituent figured to give students and instructors an easy one-click access to this built-in viewer. We have future plans to utilize the source code of TightVNC to more heavily integrate this application into our video conference web application. I think that's it.

>> CATHY CLARKE: I am going to talk briefly about why we chose to go with a web-based system architecture.

The system architecture that we adapted was dictated by our experience as creators of a course management system.

Our expertise is in building web-based applications using ASP or PHP on the server, and then using HTML, CSS, and JavaScript in the browser. So that's why we ended up choosing a browser-based system. And using this web-based framework as the user interface which kind of tied together all of these elements in the browser.

So these were using the open-source components that Zach just described.

So I am going to quickly show you how simple it is to use these open-source components and create a simple videophone application for the web.

So I am going to switch to the laptop. Right now I have on screen just a basic HTML page. I am going to copy some text. There are three links here. It's hard to see. I will describe them. The first is Correct. Basically the first step that you need is to connect to a video server and publish your video streams so that other people can accept it.

The second part is "my video." You want to be able to display your outgoing video.

And the third part is "your video" which is the incoming video that you want to have a conversation with someone else.

So, first, clicking on "connect" starts up the Red 5 server. And I'm going to paste my address from my server and connect to the server.

The next thing I need to do is select a video device which is my webcam and start that.

And then I want to start audio.

And the final thing I want to do is name my stream which is kind of like the phone number which is how someone else will connect to you.

And then publish.

So now my video stream is out there so that someone else can connect to it. I am going to do the same thing on my second computer here. Looks like the laptop hasn't detected my webcam. Let me unplug it again. And plug it in.

Okay. Well, while we try and work that out, I am going to show you -- I would basically do the same thing on that computer, start publishing a stream, and name it. Let's see where my mouse went here.

Okay. And over here I am going to click on "my video." I want to show my outgoing video so that I can see what that looks like. And so this is where I will type in the name of the stream. So this is my outgoing video.

The next step would be to click on "your video" which would be the video on the laptop over here (indicating) and I would type in the name of that stream and start that. So then I would see both videos at the same time.

So I would be able to communicate with the person using the laptop. Or whatever, wherever they are.

So that's just simple HTML. There is nothing applied, so maybe the next step that you would want to add some style sheets to it to give it a more polished look.

So I've already developed a really basic style sheet which I am going to enable now. Start my stream again. So now it just applies a bit more of a refined look to your page.

So you can see how simple it is to use these open-source software and apply them with a browser-based system and setup your own videophone. It's a pretty simple process. And in the future, of course, we would want to use log-ins so that people wouldn't have to go through and manually start streams and things like that. But that basic process can show you how easy it can be to get started.

The next part of our presentation is actually going to be another demonstration of our video conferencing tool that we actually developed and have used.

So we will hopefully get this set up. Ready? Okay. I'm finished. I'll give you the mic. I am going to pass this back to Zach. This is another demonstration.

>> ZACH SZAFRAN: There shouldn't be a lot to read on this. We're going to demonstrate a lot with video here. The first part of our application you see here is the actual content management system, IdeaTools, where our application is actually built on top of.

This is the first form of security that you will come across. This uses RIT security and authentication scheme for the content management system which then applies it to every application that's a part of the content management system.

Here is the second level of security that you will come across where you actually have a list of courses. Each course is controlled by the instructor. They control which students that have logged in can actually access the course. Which courses have which applications enabled or disabled. Go to "configuration."

And in here you can see how our application is built on top of the content management system. Along with other applications that we've built, you have the ability to enable or disable this application for your course. If you scroll down to the very bottom of this you will see our application,

the remote video hosting 4 which is already enabled. Upon enabling this, if you go to the home page you will see a video conference button which allows one-click access for students and instructors to enter our web conference application.

All right. Here it goes. Okay. Here is the general layout of our application. On the side here you see the actual grid of videos from our Flash component that we developed. We currently have two other computers, along with this one connected to the web conference where they are able to see each other and hear each other.

On the very top, we have a few tabs built into our application. They are certain tools that we felt would be necessary for a web conference to have. This first tab that you are on right now, what this does is this gives the instructor access to all of the course materials that they currently have created, any homework assignments, any essays, PowerPoints, anything that they've added to their course they have access to here. By simply selecting something they can send this to all of their students and all of their students will be forced to view whichever the instructor wants.

An example would be if the instructor wants them to see this video, they can just click the video and it will start playing on the students' screens.

Now, one of the -- what we did differently with our web video conference application compared to normal video conference applications is a lot of current conferences out there support a VNC-type connection where everything you currently see is on the instructor's desktop.

If they want to share something with you, they have to open it on their own computer and bring it up on their screen, and they really have very little interaction. They can just kind of see what's going on, and that's about it.

What we did here was made the content go directly to the students. This is their own copy that they're viewing. They can interact, pause, rewind, stop, do whatever they would like.

Another example would be a homework assignment. If we were to push that in a normal conferencing environment, an instructor would simply be able to display the homework assignment where now the homework assignment is accessible to the student, and they can actually take the test, submit it right from this conferencing application.

Okay. Hmmm.

Well, also on the top you have other tabs. They are very similar to other functionality that a conference might have. We have a chat tab where students who currently aren't participating in a video chat can still talk and interact. They can get the attention of an instructor, ask questions, answer questions.

Another thing that we added was the file uploads tab on the top. This allows the instructor to, say, upload something for a demonstration, or a file or a document that might be helpful for what they are currently discussing. The students will have access to that immediately, and they can also upload their own files, submit whatever they would like.

Another tab that we have on the top there is a settings tab which is part of the instructor's view only. On this settings tab, the instructor has a list of all of the current users of the course. And with this list the instructor can constituent figure each user with a particular role. They can be a normal student or, say, the instructor would like to have an interpreter in the room. So an interpreter can be selected, or a captioner can be selected. And based on these roles an interpreter will have a guaranteed spot in the video grid, and an interpreter will have the ability to actually type and the students will have the ability to see what the interpreter has typed.

Okay. And this is the example of the homework assignment that I was referring to earlier. This homework assignment can be submitted by a student right from this screen. They can view the results and everything just right from this application.

And another big part of our system is this application sharing that we integrated using the TightVNC viewer. What this does is it lets us put a remote desktop right on the student's screen where an instructor can give an online demonstration of a program that they'll be using, or they can show a PowerPoint on their computer and have all of the students see it. They could also do a white board on their remote desktop and draw on it as the students can watch and interact along with it.

And this is our example where VNC is displaying this exact PowerPoint that we have on this computer that the students can see.

Here the instructor can actually comment. They can draw on the screen, and it can be displayed just in realtime as the students see it. They can go through the PowerPoint, and it just makes a very nice interaction for displaying live content on the students' screens.

>> SIMON TING: Well, we're finished with our presentation. I'm sorry that we had some technical difficulties. The laptop that had all of the software that we needed wasn't really working. One of our webcams doesn't seem to be working today, and we did hope that it would be working. There was also some other problem with another laptop. So I apologize about that.

So right now if you have any questions, we can open this up to questions.

>> Audience member: Do I need to come to the mic?

Simon, I know that for other applications such as Adobe Connect and ePop users is often a learning curve for other people to get used to the software. Is this fairly user-friendly? Do you expect others to kind of pick up on it and use it because I use the Adobe Connect and ePop software pretty often, and I have a lot of issues with the audio and visual just trying to hook everything up. What do you think the learning curve is for this IdeaTools?

>> SIMON TING: IdeaTools was setup to be an easy-to-use program so that you don't have to setup anything. And also you know who your students are. You know, you can easily click a button and then start your video conference. If you are using the same system everyday, video and audio should not be a problem.

But, you know, we had some problems today, of course. Our webcam is not working. You know, if it's not working permanently we'll have to substitute with another webcam. And that can happen. Does that answer your question?

IdeaTools was designed so that you don't have to do anything. You know, you have things setup for your classroom needs and all you have to do is assign a meeting room or a group, and when those students click on that, then the students automatically go to that site.

I know ePop and Adobe Connect have a steep learning curve. That's why we designed our system to try to alleviate that problem. With this you only have to click on one button.

>> Audience member: From what I'm seeing, you're adding these components to a content management system for your school. Is the entire content management system grown? Are you just adding systematically adding these components to it? And if this is so then are there plans at any point to market or offer this out as open-source or what are your plans for the whole system even not just this part of it?

>> SIMON TING: IdeaTools is home grown and we've developed all of it here. If possible it can be offered to the general public. The video conferencing tools, you know, is not built into IdeaTools because of a lack

of time. That's the only reason. The ability to develop courses and have class lists and so forth, that's already in IdeaTools. If we have time we can build it from scratch and build a separate capability for video conferencing systems.

We hope -- well, my hope is that we can share those tools openly with people who are interested in using them. As Cathy tried to show you today, we work mostly with HTML, CSS, and JavaScript. These are all open-source and can be used by anybody. We tried to use the components of black boxes so that if you connect up to it it will work for you.

The only thing that you need to do is to modify the HTML to build your own interface.

Again, I'm sorry about the technical issues that we had. I hope you were able to learn something today.

>> Okay. Well, thank you so much for coming to the workshop. We really appreciate your time, and we've learned a lot.

I would like to remind you that if you would like to fill out a paper evaluation form you can give that to us today before you leave, or if you prefer to fill one out online during your own time that's fine as well. If you have the opportunity to pick up a copy of this PowerPoint presentation, we do have that available on the chair over in the back of the room by the door. Okay? Thank you so much. Bye-bye.

(Applause)

\* \* \* \* \*

This is being provided in a rough-draft format. Communication Access Realtime Translation (CART) is provided in order to facilitate communication accessibility and may not be a totally verbatim record of the proceedings