Videoconferencing Using Open-Source Software:
Building a Proof-of-Concept Platform for Communications and Distance Learning

Simon Ting, Instructional Developer, NTID
Cathy Clarke, Digital Media Specialist, NTID
Zach Szafran, Student Programmer, NTID
Regina Kiperman-Kiselgof, Employment Counselor, NTID

This paper is intended to accompany a show-and-tell presentation describing a project to create a proof-of-concept videoconferencing system using open source software. The notes provide background information for the presentation. As the information is general in nature, we have not provided article citations. If you need more information, you can find it online using a search engine.

Introduction

As we begin, we invite you to consider what usually happens in the lecture classroom of today. Here's a typical scenario:

> *The professor stands in front of the class and teaches a group of students who take notes. The classroom may be wired for sound and video, with a podium or laptop computer allowing the professor to project a PowerPoint presentation. The professor will probably use an overhead projector to project article excerpts, photos or diagrams, on which the professor may jot notes or sketch diagrams during the lecture. In the course of the lecture, the professor may quiz students to test their understanding of the main points of the lecture. Students may be invited to discuss some points, demonstrate a project, or do a show-and-tell. While the student is talking, the professor or other students may offer suggestions or comments. Sometimes the class may be videotaped so that students can review the lecture on video after class. In the context of teaching deaf learners, the instructor may be signing. If not, there would a sign language interpreter or teacher's aide in the classroom to facilitate communication with deaf learners in the classroom. A note taker may be present. In rare cases, the professor might even request real-time captioning support.*

In this presentation we will describe a project in which we explored the feasibility of building a low-cost videoconferencing system using free open source software components and low-cost off-the-shelf hardware. As we shall report, even though we did not achieve all our goals, we did succeed in building a web-based videoconferencing system capable of supporting the online equivalent of above classroom scenario.

What is Videoconferencing?

Traditionally, videoconferencing refers to the use of video and audio to facilitate communication and interactions between individuals in different geographical locations. A good example: videophones are now widely used by deaf people in this country to communicate with each other and with hearing people through video relay services. In the context of the classroom scenario we just described, videoconferencing involves more than showing talking head videos of the participants.

Besides seeing and hearing one another, teachers and learners must have show-and-tell capabilities, using PowerPoint slides to present a lecture and a whiteboard for brainstorming, jotting down ideas and points, drawing sketches, diagrams, charts, etc. Students working on joint projects must be able to share project files and edit documents and projects together, just as they would in a classroom or lab, all of which requires the ability to share computers and software used to create projects in today's increasingly electronic and digital educational environment. What this means is that when we refer to videoconferencing, we are talking about a platform, or an all-in-one solution, made up of multiple tools offering different ways for teachers and learners in different geographical locations to communicate, interact and work with each other online.

When we hear the word "online" nowadays, we immediately think of the world of Web browsers, such as Internet Explorer and Firefox. This adds another plank to our conceptual videoconferencing platform. All the tools must work in Internet Explorer, Firefox, and other popular Web browsers, such as Safari. There should be no annoying wait times for software to download. Instead, teachers and learners must be able to join a live online class anywhere and anytime, using any computer with a web browser. This is the reason why more and more commercial videoconferencing vendors are offering web-based solutions to their customers.

Required Tools and Capabilities

Based on the above understanding, we developed a wish list of tools and features for our conceptual videoconferencing

platform. In the world of mainstream business and education, there is a growing consensus on what consistutes must-have conferencing capabilities. The consensus list includes:

- Live video/audio
- Text chat (aka IM or Instant Messaging)
- Remote presentation/annotation (PowerPoint, whiteboard)
- Remote collaboration (sharing computer screens, software, documents/projects)
- Meeting setup/management
- Video recording/archiving

We have two requirements of our own not normally considered necessary in business, which is the sphere that videoconferencing tools were originally meant to serve:

- CMS integration
- Built-in accommodation for sign language interpreters, note takers, and captioning specialist

We'll now discuss the desired tools/capabilities in greater detail.

Live video/audio. The live video/audio tool will need to be configured to support multiple video displays, showing the video streams from the cameras focused respectively on the:

- Professor
- On-site students
- Off-site students (with turn-taking protocol so a group of geographically dispersed students can take turns sharing a single on-screen video display)
- Sign language interpreter (if present)

It may not be ideal to expect geographically dispersed off-site students to take turns sharing a single on-screen video display. However, the speed and computing power of currently available Internet connections and desktop PCs is such that video quality degrades noticeably when more than four video streams are playing simultaneously, making the nuances of signing hard to read.

Text chat/Instant Messaging. Since current technology will not allow all off-site student videos to be displayed simultaneously, we will need text chat or Instant Messaging for students to ask questions or to catch the attention of the instructor who can then assign one of the on-screen videos to display the student's webcam under some turn-taking protocol. Students, of course, can also use IM for private or group chat to supplement video/audio chat.

Remote presentation/annotation. Minimally, the professor should be able to show a PowerPoint presentation to be viewed by all off-site students. Since college professors are increasingly using course management systems (CMS) to upload and manage their course materials online, there should be convenient access to the professor's online learning assets, such as PowerPoint slides, quizzes and surveys, during a remote presentation. A shared whiteboard capability is another essential feature, allowing instructors to scribble and doodle freehand, much as they would on a whiteboard in the classroom.

Remote collaboration. This refers to the ability to take control and work on a computer remotely. In the classroom context, it allows instructors to take control of a student's computer, if needed, in order to provide hands-on assistance. Students may also share computer screens with each other while working on joint projects or editing a collaborative paper during a class. Instructors may also want to give selected students control of the instructor's computer for the purpose of performing some hands-on activities or presenting a class project.

Meeting setup/management. Instructors need the ability to set up different meeting rooms, schedule meeting times, and give permissions for students to join a live class online. During the class, instructors need to view who is present in the meeting room and what the student's role is (viewing or holding turn to present).

Video recording/archiving. A desired feature is the ability to automatically record and save all videos to a central archive, where students can access them conveniently for review purposes after class.

CMS integration. Colleges are increasingly expecting instructors to use online CMS or course management systems for managing course content and learning assets. Therefore, it makes sense for a classroom-based videoconferencing platform to work integrally with a CMS. It makes it easier to set up and manage meeting rooms and meetings. All courses in the CMS can be equipped with an online meeting room by default. Then, since the CMS already knows all members of a class, students can login to the course and click a button to join a live online class. Those with a webcam would immediately have their

webcams activated and recognized. An additional benefit of CMS integration is scalability or the ability for the same application to be used by many people, in this case, students and instructors in other courses in the CMS. Yet another benefit is security; the same security measures used to control access to most CMS will also govern access to the integral videoconferencing tools.

Built-in accommodation for sign language interpreter and/or captioning specialist. This is not usually a consideration for conferencing systems targeting mainstream users, but it would not take much to include the functionality needed to set up a meeting room to facilitate sign-on by sign language interpreters and captioning specialists, with tools to make it easier for them to perform their functions. The benefit of such an accommodation is in terms of supporting the teaching of deaf learners.

Open Source Software Components

We built the videoconferencing platform that you just saw from a few open-source applications. We will touch on these applications here. While we did modify some of the software functionality, the modifications we actually made were minimal. Instead, most of the work we did went into (1) integration, i.e., getting the components to work with each other, and (2) interface design, i.e., building the user interface for accessing and using the tools in the system. The list of open source applications we chose to use is as follows:

- Red 5 Server
- JW Media Player
- TightVNC

We will discuss each of these applications in a little more depth.

Red 5 Server. Red5 is a free, open source software package written in the Java programming language. Its purpose is to support communication between Adobe flash applications. It has support for live stream publishing, audio and video streaming, object sharing as well as the recording of streams. The goal of the Red5 developers was to create and share a free and open source alternative to Adobe's high-priced and propretary Flash Media Server. Though Red5 is still in alpha testing stages and does not fully support all of the features of the Flash Media Server, we were still able to use it to implement our a working prototype videoconferencing application that works in the Web browser. Particularly useful is Red5's support for users to broadcast a video stream live from a webcam for other users to view immediately or for the server to record for users to view later. Red5 supports RTMP (Real Time Messaging Protocol), a communications protocol developed by Adobe Systems for streaming audio, video and data over the Internet between a Flash player and a Flash Media Server (FMS).

Red5 server comes with a suite of server-side and client-side demo applications. This include the OFLA (Online Open Source Flash Conference) application which serves our needs admirably and it became one of the server-side components of our application without almost no modification. We utilize the source code two client-side applications (Publisher and Recorder) to create our browse-side component for display our video streams. This component builds and displays a grid of webcam videos from up to four users during a videoconference. This application can also record the video for archival purposes.

JW Media Player. This is a complete and fully functional Flash component for viewing video and playing audio. Implemented as an Flash .swf file that can be embedded in a Web page, it provides an easy way to add video and audio to the page. The author Jeroen Wijering released the player under a Creative Commons License, allowing users to utilize the player and modify the ActionScript source code for noncommercial purposes. The author also provided a rich JavaScript API, making it easy to control the functionality of the player from a Web page using JavaScript. Of particular interest to us is the ability to use JW Media Player to view RTMP streaming video served by Red5. Of interest also is JW Media Player's excellent support of video captions because of the vital role of captioning in providing deaf users access to video information.

TightVNC. TightVNC is a free remote control program derived from the popular VNC (Virtual Network Computing) program. TightVNC lets the user see the desktop of a remote computer and control it with the user's own mouse and keyboard. As host, one has the option of sharing the entire desktop of one's computer, an area of the desktop, or a specific program running on the desktop. TightVNC is free and cross-platform, with all its source code available if one is interested in modifying its functionality to suit one's needs. TightVNC has a Java viewer plug-in that allows users to view and control a remote computer through the Web browser. This was used to implement the remote desktop and application sharing tools of the overall videoconferencing platform we just demonstrated.

Web-based System Architecture

We set out to design a videoconferencing system that we hope can be built with minimal financial resources, using low-cost off-the-shelf hardware and free open source software as building blocks of the system. A related goal is the ability to put together the system by ourselves. The team consists of an instructional developer (Simon Ting), a Web designer (Cathy Clarke), and a student programmer (Zach Szafran). Before we began the project, none of us had any experience with

videoconferencing beyond a passing acquaintance with using embedded video on Web pages. The key concept is to work with video in a format that allows them to be displayed on Web pages using a browser plug-in or ActiveX components. This separates the system development into two parts: developing the plug-in components, which requires hardcore coding skills, and designing the system in which users can access the components on Web pages built with HTML, JavaScript and CSS (Cascading Style Sheet) elements.

As a student programmer, Zach is not expected to be a permanent fixture of the team. The system architecture we adopted allowed Zach to employ his coding skills to build modular system tools and test them in standalone mode. The other members of the team can then leverage their system integration and Web design expertise to assemble the overall platform using Zach's plug-in tools without worrying about their internal workings. This architecture also allows for future system enhancement and expansion. If a more functional plug-in to replace an existing plug-in becomes available, we can replace the old plug-in with the new. If a plug-in delivering a new functionality becomes available, we can employ that by adding a new Web page with the embedded plug-in.

Demonstration of Videoconferencing Platform

- **Logon to IdeaTools:** Taking advantage of CMS integration to set up meeting rooms, add and manage participants.
- **Start the Meeting:** Single-button convenience in starting/joining a live online class meeting.
- **Videoconferencing Tool:** Participate in a live video/audio chat.
- **Text Chat.** Catching instructor's attention. Asking questions. Turn-taking protocol. Private messaging between participants.
- **Remote Presentation Tools.** Using PowerPoint and whiteboard tools to support a live online lecture.
- **Remote Collaboration Tools**: Sharing computer screens, software. Edit shared documents and project files.
- **Sign Language/Captioning Modules.** Tools to facilitate the work of sign language interpreters and captioning specialists.
- **Recording/Archiving Tools**. Record, archive, and post videos automatically.

After Word

Before ending this presentation, we wish to say a few words about the free IM or Instant Messaging services offered by Microsoft, Yahoo!, and Google, among other services too many to mention. For several years now, audio/video chat has been an integral part of most IM clients, software that you download and install on your computer. Many IM clients also support an add-in or plug-in architecture that allows you to add remote desktop/application control and whiteboard capabilities. However, IM services are typically designed for two-party personal chat. Even though some newer services, such as iChat AV (another service, ooVoo, offers multi-party audio/video but does not offer remote desktop/application sharing and whiteboarding), they have no meeting room functionality. As a result, the hassle factor involved in setting up a meeting for more a limited number of attendees is simply too great for ad hoc live online classes.