Syllabus2004

Communication and Collaboration in Online Courses Using Macromedia Flash

Brian Klaas, Senior Systems Designer, Johns Hopkins Bloomberg School of Public Health

Macromedia's Flash technology has evolved over the years from a simple tool for delivering low-bandwidth animations over the Web into a platform for delivering complex applications, rich media content, and multi-user environments. Flash, alas, is still seen by many as the purveyor of annoying "skip intro"-style Web site splash pages. While it is true that Flash remains the dominant delivery mechanism for vector-based animation on the Web, it is also true that Flash has become the delivery mechanism for complex business and educational applications created by some of the largest companies and educational institutions in the world. At the Johns Hopkins Bloomberg School of Public Health (JHSPH), we've used Flash for many years as a delivery tool for educational content. More recently, we've recently made it our platform of choice for delivering communication and collaboration applications that are used by hundreds of students and faculty members around the world.

What Is Flash Good for, Anyway?

Flash is capable of delivering far more than animations or short video clips. With relative ease, for example, you can create complex animations of biochemical processes in the human body with text and audio. At JHSPH, we now deliver entire lectures, including traditional PowerPoint–style animations, using Flash as the universal, cross-platform playback mechanism. More intriguingly, we have begun to deliver quizzes and assessments in Flash (oftentimes in the middle of lecture presentations) and store the results of those assessments in a database for future analysis by course faculty. It's now possible to capture walkthroughs of statistical applications in real-time on the desktop for student playback right in a Web browser. We can even add quizzes or branching to other parts of the walkthrough in this playback.

A whole new class of Flash applications focusing on communication and collaboration has arisen during the past few years. Multi-point audio and video conferencing are now possible with the only client requirement being the Flash plug-in. You'll come across highly specialized applications—for everything from cattle auctions to shared chemistry experiments in virtual labs—that use the Flash player as a key component of the delivery infrastructure. Many educational institutions across the country, including JHSPH, have begun using custom-built communication applications in Flash. These applications might combine text chat, audio and video streaming, shared whiteboards, polling, and browser sharing to allow students and faculty from around the globe to communicate about and collaborate on course-related projects in real time.

The Flash Communication Infrastructure

The Flash Player

Macromedia, the developer of Flash, has created a rather complex set of servers and software that now form the core of the Flash development ecosystem. At the center of Flash development is the Flash player itself: the lightweight, cross-platform, bandwidth-friendly plug-in currently installed on over 97% of computers connected to the Internet.[1] Until recently, if you wanted to develop content for playback in the Flash player, you had to use the Flash authoring environment (currently represented by Macromedia's Flash MX 2004 and Flash MX 2004 Pro products). However, you can now author complex Flash applications—communication, collaboration, or business—in a number of products, including Macromedia's Breeze Presentation, Breeze Live, RoboDemo, Camtasia's Camtasia Studio, Quarbon's ViewletBuilder, and PresentationPro's PowerConverter.

You'll still need to use the Flash authoring environment to create custom communication and collaboration applications delivered using the Flash plug-in. Macromedia's Breeze Live solution is one of the few currently available commercial products which enables real-time, rich communication and collaboration among users via the Flash plug–in. Its price and relative complexity may deter most educational institutions from implementing it as a solution, however. In many cases, building a custom communication or collaboration solution delivered via Flash may be a better and simpler option for many educational institutions. To do this, a few key pieces of infrastructure must be in place.

Servers

At the heart of any communication or collaboration application delivered via Flash will be a server of one of two kinds: Macromedia's own Flash Communication Server or one of a small number of socket servers which can communicate with Flash clients. Socket servers designed to work with Flash include Moock.org's Unity Server and Electrotank.com's ElectroServer. Users at different computers cannot communicate or collaborate with one another via Flash in any kind of efficient manner without using a centralized server or set of servers to manage the communication between different user computers.

Which kind of server you set up to manage communications between users of your Flash application depends on what kind of information you plan to send across the Internet between clients connected to your application. If your application requires only the sending and receiving of text-based messages (most chat, IM, gaming, and basic whiteboarding applications rely only on small text messages to communicate), then a socket server such as Unity or ElectroServer would be appropriate. You can use the Flash Communication Server for these kinds of tasks as well. What sets the Flash Communication Server apart, however, is its ability to deliver audio and video in addition to text messages between connected clients. If you plan on creating a collaborative application using Flash that combines audio or video (or both) with whiteboards, slides, or text, then the Flash Communication Server is your only option.

Bandwidth

Once you've selected a server for your communication or collaboration application, you must also ensure that you have adequate bandwidth available for delivering your application on both the server end and the client end. Applications that use only text for communication require significantly less bandwidth than those that use audio or video in addition to sending text messages between connected clients. This is not to say that bandwidth is not important for text message-based applications. If your application requires rapid delivery of messages between connected clients (for a chemistry lab simulation, for example) so that everyone can see changes in the application at the same time, ample bandwidth and minimal time between sending and receiving messages is a must. Although text messages are generally small in size, students and faculty alike have come to expect near-instantaneous delivery of such messages (particularly in IM applications), so be sure not to place the server on a slow network.

The issue of available bandwidth becomes even more critical for Flash applications that use audio or video. Multi-user communication applications are notorious bandwidth hogs, as they require bandwidth that grows exponentially as each new user signs in to the system. (The basic equation for bandwidth consumption in such applications is: total bandwidth = required bandwidth for one user * the total number of users currently connected to your application.) You can use proxying strategies during development and deployment to help reduce bandwidth consumption, but audio and video collaboration applications may still quickly use up all available bandwidth on a network if not carefully designed to minimize bandwidth consumption.

Databases for Storage and User Authentication

If you need to store data about what goes on in a communication or collaboration application for retrieval at a later time, or if you've established user authentication to gain access to your systems, it is highly likely that you'll need some sort of application server connected to a database server. If, for example, you wish to store the text from a chat session permanently, you'll have to set up connections either from the Flash application client or the server to an application server (such as ColdFusion, ASP, or PHP), which will then take that chat text and save it to a database. There are many ways to do this, but the most popular involve the use of Flash Remoting, XML–based Web services, or ActionScript programming commands built in to Flash, such as sendAndLoad(). While a complete description of Web services and Flash ActionScript commands is beyond the scope of this article, it will be useful to discuss the basics of Flash Remoting as it is frequently used in the development of collaborative Flash applications, particularly those that use the Flash Communication Server.

Flash Remoting is a package of service libraries that runs on both the client (as a standard set of ActionScript commands) and on the server. Flash Remoting is available for ColdFusion, ASP.NET, and Servlets on J2EE (an open-source implementation is also available for PHP). The Flash Remoting gateway reduces the complexity of transferring information between a Flash client and a server. The server may then pass that information on to a database, Active Directory, LDAP, or e-mail server—or do with that information as it sees fit. Information passes in a highly compressed, binary format rather than as text; this reduces the amount of time and bandwidth it takes to pass information between the client and the server. Compared to similar data sent via Web services or an ActionScript command such as sendAndLoad(), Flash Remoting packets are significantly smaller in size and more efficiently transported across networks.

An institution could use Flash Remoting to tie a Flash application to an enterprise directory for authentication, for example. The Flash client would display the login form, and when the user submitted his username and password, Flash Remoting calls would be made to an application server. That application server would then connect to the enterprise directory and return a success or failure message to the Flash client. If successful, the Flash application would log in to the server and display information or tools based on a wide range of information that would be returned from the server via the Remoting gateway.

Flash Remoting is not necessary if you are building applications using socket servers such as Unity or ElectroServer. It is, however, your only option for communication between a Flash Communication Server and application servers. Even if all you want to do is save text from a chat application to a hard disk and you are using the Flash Communication Server as your server product, you must use Flash Remoting. No other options are available. Unity and ElectroServer do not have this limitation.

The Costs of Building Communication Applications in Flash

Building communication and collaboration applications in Flash can be an expensive process. You will, at a minimum, need a communication server and a Flash authoring environment. Once the server is in place, though, different paths may be taken in developing communication and collaboration applications.

For its Flash Communication Server, Macromedia provides a free set of components that even a relatively inexperienced Flash developer can use to get communication applications up and running in very little time. By simply dragging and dropping a few of the components into a Flash movie, setting a few configuration parameters, and creating an folder on the Flash Communication server, a developer can get a two- or four-way videoconference application up and running in less than half an hour. Macromedia has included Whiteboard, audio conferencing, and text chat components in this free set of components, making basic application building quite simple.

Both Unity and Electroserver provide sample applications and Flash libraries that a developer can use to quickly build communication and collaboration applications in Flash. While the focus for both socket servers is chat applications, you'll also find some components for creating simple games. The breadth of pre-built components for socket servers is much less than those provided for the Flash Communication Server, though that is really a reflection of these products' focus on text-based messaging, rather than full audio/video/text streaming.

Moving beyond the functionality provided in the basic set of Flash Communication Server components or developing for a socket server is a non-trivial enterprise. To truly create rich learning environments using Flash, you will need the talents of experienced Flash and ActionScript developers. While part-time students can build some simpler applications, you'll find it imperative to hire full-time developers to create enterprise-class applications (or even applications used by more than a handful of courses) with full support.

The potential cost of building rich collaborative applications in Flash may be so large that IT staff may need to consider evaluating commercial products such as Macromedia's Breeze Live or Convoq's ASAP. While these products do not come cheap even to educational institutions, the savings (both financial and temporal) gained by having a ready-made, fully supported solution may far outweigh the benefits (and avoid the pitfalls) of homegrown solutions.

Resources:

Macromedia Flash Communication Server MX
http://www.macromedia.com/software/flashcom/

Unity
http://www.moock.org/unity/

ElectroServer
http://www.electrotank.com/electroserver/

Macromedia Breeze Live
http://www.macromedia.com/software/breeze/

Convoq ASAP
http://www.convoq.com/

Camtasia Studio
http://www.techsmith.com/products/studio/

Viewlet Builder
http://www.qarbon.com/

Programming Flash Communication Server
http://flash-communications.net/

[1] http://www.macromedia.com/software/player_census/flashplayer/version_penetration.html