

Converting Synchronous Instruction for Asynchronous Delivery

Raj Boora
Department of Educational
Psychology
University of Alberta
Canada
Raj.Boora@ualberta.ca

Alan R. Davis
Division of Technology in
Education
University of Alberta
Canada
Alan.R.Davis@ualberta.ca

T. Craig Montgomerie
Department of Educational
Psychology
University of Alberta
Canada
Craig.Montgomerie@ualberta.ca

Abstract: Students miss classes for a number of reasons. With emerging technology, providing asynchronous access to that class is possible. Rural Advanced Community of Learners¹ (RACOL) is creating a new standard in delivering distance education to students in rural Alberta. RACOL uses a combination of the Community of Learners Models of Instruction, n-way broadcast quality real-time video classes in which the instructor and all students can be seen by all participants, state of the art visualization using interactive whiteboards, individual computers equipped with workgroup software, and provision of asynchronous access for students who may miss a lesson. This paper describes the system developed to record synchronous communication and to provide asynchronous access to that material by students or teachers.

The Problem

The Fort Vermilion School Division #52 (FVSD) is located in the Northwestern corner of Alberta, a very rural and remote area of Canada. In conjunction with the University of Alberta and a number of other partners, FVSD is implementing the Rural Advanced Community of Learners (RACOL) project to provide state of the art, synchronous instruction to students in K-12 and post-secondary courses (see Montgomerie, King & Dropko, in press). The design of RACOL includes four concurrent streams:

- an MPEG-2 stream that contains the video of the presenter (usually this is the teacher/instructor), or, if the visualizer is being used or a DVD or Videotape is being shown, this will be displayed with an optional picture-in-picture of the teacher/instructor,
- an MPEG-2 stream that contains the combined images of the student locations in a split-screen format,
- a mixed audio stream, and
- an AVI data stream that is the combined image of what is displayed on the SmartBoard™.

Our challenge was to create a Web-based system that would allow students and teachers to access these streams asynchronously.

Specifications and Constraints

The system specifications and constraints included:

- A student would have a similar experience to what they would have had in a face2face classroom, with the added capabilities to pause and review the material. RACOL allows only registered students and authorized instructors and administrators to view recorded sessions using a unique ID and password.
- The entire system must be able to run in an automated manner with minimal intervention.
- The system should be able to recover from interruption and restart unattended. The control of the system is mediated by an external “turn key” system. This requires completely unattended operations.
- Any development should be standards-based. We did not wish to get involved proprietary software issues.
- If possible, Microsoft products should be used. FVSD has personnel who are MCSE certified and prefer to be a single platform operation.

¹ The Rural Advanced Community of Learners is supported by a contract from CANARIE Inc.

™ SmartBoard is a trademark of Smart Technologies.

- The user interface must be “light weight” and able to run on a wide variety of end user machines, ranging from Windows 95 with 28.8k connections to the latest Windows and Mac platforms with broadband access. While accessing the recorded material, the user should have seamless random access to any video stream without any interruption to the audio stream.

Theoretical Basis

A core requirement for a successful learning program is the ability for the instructor to effectively communicate with the student. RACOL attempts to create virtual learning environments for students in the FVSD and is based upon concepts such as the community of learners (Scardemalia & Bereiter, 1996). Without the face2face communication that is almost taken for granted in the traditional classroom, many students who must access course material outside of class are at a disadvantage. Using recorded video can restore much of what is lost to these otherwise disadvantaged students (De Vries, 1996; Heerema & Rogers, 2001; Fulford & Sakaguchi, 2001).

The recorded video sessions can also enable students to “take the classroom with them” when they are unable to attend class. Recreating the classroom, complete with students and instructors, allows for a perception on the part of the student that they can, via computer-mediated communications, make a meaningful contribution to the class by interacting with others. Positive interactions facilitated by this system are bound to lead to improved achievement for the student (Freddolino & Sutherland, 2000). It is important to note that the type and quality, not the quantity of interaction of the interaction is important for achievement (Fulford & Sakaguchi, 2002).

The development of a user interface that will allow a student to reconstruct a synchronously delivered lesson at a later time in a distant location will provide another tool for use in distance education.

Implementation

We examined many alternatives to recording and subsequently providing access to those recorded sessions. These included “off the shelf” commercial solutions, modified commercial solutions, and shareware solutions. It became apparent that, given the constraints and capabilities of RACOL, we would need to develop our own system using existing methods and technologies. The implementation began with two major foci: development of a method to capture the streams and to store them in a way that would be easily accessible, and development of a client tool to access the stored streams. We wanted to take a standards-based approach, and MPEG-4 seemed the appropriate candidate. While other streaming protocols were considered, we finally settled on QuickTime because of its flexibility.

Capturing the Synchronous Information

Apple Computer Ltd. produces a set of streaming server software and tools that allows the capture and streaming of a number of different protocols (including MPEG-4) under the QuickTime protocol. The commercial version is the QuickTime Streaming Server (QTSS) (Apple, 2002) which runs on Apple hardware. An open source version, the Darwin Streaming Server (DSS), is available for use on a number of different platforms including Unix and Microsoft Operating Systems. We tried QTSS on an MacOS 10.2 server and DSS on a Windows 2000 server. While DSS provided the streaming capabilities necessary, the development tools that were provided were not equivalent to the tools available on QTSS. We decided that, for the first implementation, we would use QTSS while keeping the design as compatible with DSS as possible.

Five different video streams need to be provided: the presenter and instructor streams at broadband speed, the same streams at modem speed, and the data stream (e.g., the SmartBoard stream). A timing mechanism also needed to be developed to allow synchronization of these streams.

The output from the central control system consists of the instructor vector stream, the split screen student stream, a single audio stream and a data stream. The two video streams are captured using hardware conversion (Imaging Source DFG1394-1). The audio stream is captured using a USB audio converter (Griffon Technology Corporation - iMic USB interface). Capture of the data stream is undecided at this time. A system similar to the *Whiteboard Session Logging and Post-Production Tools* (University of Washington Department of Computer Science and Engineering, 2002) is being investigated. The video and audio streams are captured via the QuickTime

Sequence Grabber, compressed using MPEG-4 and synchronously streamed using the RTSP protocol, and placed on disk for asynchronous delivery.

The instructor indirectly controls the streaming process as an adjunct to the classroom control system. Access to the instructor and classroom video and audio material is via a secure system with authorization controlled by the school administration.

When we specified the system initially, we thought that we would be able to make a class available to students within one hour of the class ending. Under our current implementation, it appears that students will be able to access any stream within seconds of it being recorded. This may allow a student who is unable to attend a class in person to participate from any location that has relatively high-speed access to the Internet. This is still being investigated.

Viewing the Captured Video Asynchronously

Figure 1 shows Web interface that allows the user to view any of the three video streams while hearing the audio stream. QuickTime provides the ability to start a movie, regardless of type at a preset time using the *starttime* parameter. Using HTML and JavaScript this parameter can be set dynamically for each stream. A Perl script (Wall, 2000) generates database queries that obtain the starting address of any recorded video session with two mouse clicks.

Once connected, each client manages its own video feed using Javascript (Netscape Sun, 1995). Synchronization is based on a timer that starts when a video stream is initialized. Because the audio and video data is packaged together and each of the three streams are themselves recorded synchronously, this single Javascript timer is all that is required to locate the same point in any of the streams. This client side scripting frees the video server from having to manage multiple users and is only required to respond to client demand for a video stream.

The Perl script on the Web server manages the system from the client's point of view. The client first requests access to a script that queries a remote database. This database authenticates the client and passes this information to the video server, ensuring that only authorized individuals can view video. The Web server then uses the information from the remote database to build an HTML page that presents the authorized sessions to the client.

While it was not an initial specification, it became apparent as we were working with the prototype, that students need to bookmark a location where they could return to review. This feature has been added.

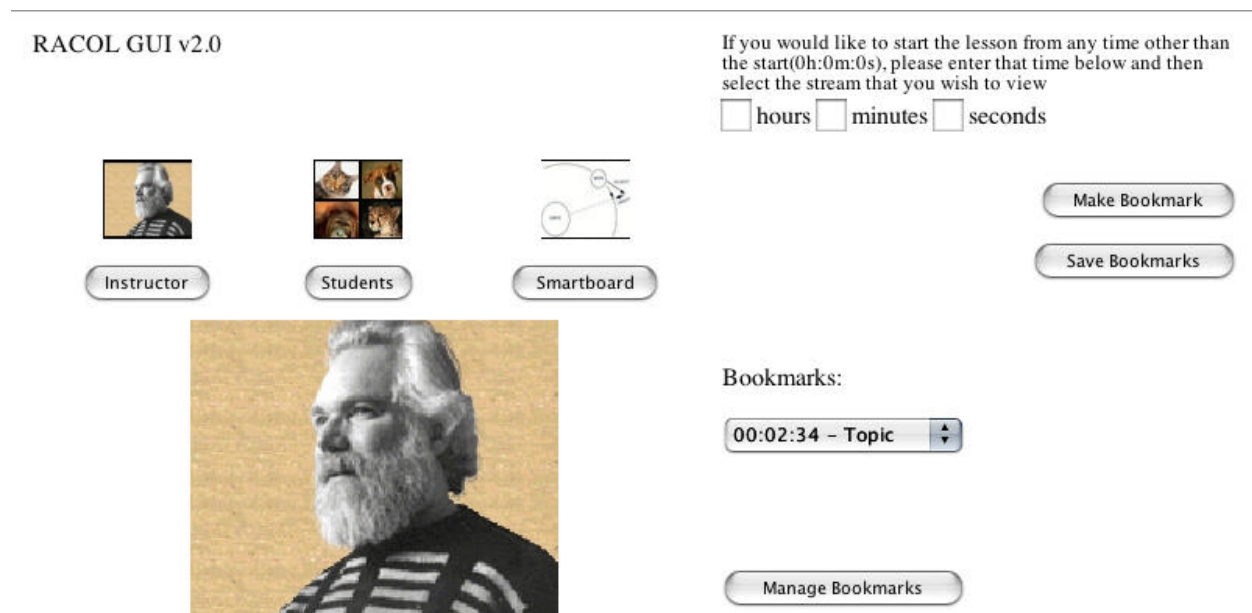


Figure 1: Asynchronous Interface

Conclusion

Providing asynchronous access to “real time” lessons is a challenge. This paper describes a system that captures the different streams, and then provides access to these streams through a secure Web interface. By ensuring that all

the streams (presenter, student, data, and audio) are available to the student as they review a lesson, the student can control their progress through a course, book marking important sections and rewinding to repeat a section while looking at, say, the SmartBoard stream instead of the presenter stream.

References

- Apple Computer Ltd (2003). Quicktime 6.2. Retrieved May 6, 2002, from, <http://www.apple.com/quicktime/>.
- Apple Computer Ltd (2003), Darwin Streaming Server 4.1.3. Retrieved May 6, 2002, from, <http://developer.apple.com/darwin/projects/streaming/>.
- Apple Computer Ltd (2003), Quicktime Streaming Server 4.1.3. Retrieved May 6, 2002, from, <http://www.apple.com/quicktime/products/qtss/>.
- De Vries, Y.E., (1996), The Interactivity Component of Distance Learning Implemented in an Art Studio Course, *Education*, 117(2), 180-185.
- Freddolino, P.P., Sutherland, C. A., (2000), Assessing the Comparability of Classroom Environments in Graduate Social Work Education Delivered via Interactive Instructional Television, *Journal of Social Work Education*, 36(1), 115-130.
- Fulford, C.P., Sakaguchi, G. (2001). Developing a Taxonomy of Interaction Strategies for Two-way Interactive Distance Education Television. *International Journal of Instructional Media*, 28(4), 375-396.
- Fulford, C.P., Sakaguchi, G. (2002). Validating a Taxonomy of Interaction Strategies for Two-way Interactive Distance Education Television. *International Journal of Instructional Media*, 29(1), 79-92.
- Heerema, D.L., Rogers, R.L., (2001), Avoiding the Quality/Quantity Trade-Off in Distance Education, *T H E Journal*, 29(5), 14-19.
- Montgomerie, T. C., King, C. and Dropko, K. (in press). A Needs Assessment and a Design for a Distance Education System: The Rural Advanced Community of Learners (RACOL). In C. McNaught & D. Lassner (Eds.), *Proceedings of ED-MEDIA 2003 – World Conference on Educational Multimedia, Hypermedia and Telecommunications*. Norfolk, VA: Association for the Advancement of Computing in Education.
- Netscape Sun, (December 4, 1995), Javascript, <http://wp.netscape.com/eng/javascript/>.
- Scardamalia, M., & Bereiter, C. (1996). Adaptation and understanding: A case for new cultures of schooling. In S. Vosniadou, E. de Corte, R. Glaser & H. Mandl (Eds.), *International perspectives on the design of technology—Supported learning environments*. (pp. 149-163). Mahwah, NJ: Lawrence Erlbaum Associates.
- University of Washington Department of Computer Science and Engineering, Whiteboard Session Logging and Post-Production Tools. Retrieved December 8, 2002, from http://www.cs.washington.edu/education/dl/tools/wb_logging/.
- Wall, L. (March, 2000), Perl (Practical Extraction and Report Language) v5.6. <http://www.perl.org>.