Delivering Instructional Video Anywhere: An Intelligent Wireless Streaming Video Delivery Mechanism for Mobile Asynchronous Distance Learning

Carlos R. Morales, Charles D. Miller
Purdue University, Knoy Hall, Room 363, West Lafayette, IN, 47907

Abstract
During the last two years, Purdue University's Computer Graphics Technology department has been developing an open-architecture distance learning system capable of both synchronous and asynchronous instructional delivery. The system strives to provide the ability to deliver any instructional content, to any learner, under any condition through the integration of intelligent modules that dynamically determine the most appropriate mode of presentation for any particular learner.

This paper details the development of a module for Purdue's Unified Multimedia Delivery System capable of intelligently identifying and delivering asynchronous video content to users on traditional PC's or PDA's over a wireless environment. This provides the system with the capability to deliver traditional video-based instructional and supportive documentation such as schematics, illustration, and animation to students outside of the confines of their traditional classroom or laboratory.

While many commercial solutions address the delivery of training materials using Internet technology, a commercial turnkey solution for delivering streaming video content over a wireless IP network does not currently exist in the distance learning market. Technologically, the system is built around Microsoft's Active Server Pages and Windows Media Services. ASP s used to implement the logic and WMS provides the video streaming capabilities. The system also addresses on-site video delivery through the integration of a intelligent mechanism capable of identifying users accessing the module from wireless PDA's and delivering alternate content through IEEE 802.11b Wireless LAN. Additionally, the system utilizes path-finding methodology and artificial intelligence algorithms to intelligently select the delivered instructional, user-specific content.

Introduction

There is a tremendous benefit in delivering customized instructional content to the learner that not only takes into account the learner's attributes, but also the mode of access of the learner. The Unified Multimedia Delivery System (UMDS) being developed at Purdue University attempts to provide a mechanism for intelligently delivering instructional content to learners in a wide variety of situations. By analyzing the learner attributes, such as past performance and media preference, as well as, the device used by the learner to access the materials, the system is able to select the appropriate media to meet the desired learning outcomes.

Development of the host UMDS System

To gain a better grasp of how the wireless PDA module integrates into the UMDS, it would be beneficial to examine the construction and the capabilities of the UMDS as a whole. The authors' goal was to provide many of the same media delivery capabilities available in a traditional classroom, but in a distance environment. To accomplish this objective, the authors examined the activities that occur within a traditional classroom and developed a set of criteria that dictated the technological construction of the distance learning system.

By organizing the UMDS into the areas familiar to students and instructors the system would be easier to adopt. The authors identified four areas: lecture, homework, tests, and instructor/student conferences. The lecture would contain any activity in which the instructor delivers visual and auditory materials to a group of learners. The homework component would contain any activity in which the learner is expected to retrieve the instructional materials himself/herself, complete an assignment, and turn-in his or her work. The test area encompasses activities that asked the student to demonstrate his or her mastery of the instructional objectives. Finally, the conference area would serve as an area for the instructor and the student to discuss issues privately.

Once the areas of the system had been identified, the authors established a set of criteria that would help to identify the technological components needed to be successful. The authors elected to look at the areas in terms of communication flow, pace control, synchronicity, and presence of the instructor.

First, the authors identified the direction and amount of communication flow in each of the areas. This analysis would serve to guide the authors in allocating the available bandwidth. During lectures, most of communication flowed from the instructor to the students. A smaller amount of information flowed from the students back to the instructor in the form of feedback. This made it necessary for the technological component selected to allocate the majority of the bandwidth for communication from the instructor to the students during lecture. In the other areas, the communication flow would be approximately the same in both directions.

The second area, pace of instructional delivery, would serve to allow the authors to devise a control mechanism for the interactions within each of the identified areas. During the lecture component, the instructor would set the pace. While the students would have the ability to ask questions in the middle of the lecture, the instructor would have the option of answering them immediately or postponing until the end of the lecture. The authors would need to provide a mechanism for the instructor to control all aspects of the video streams during lecture. This would not be the case with the homework, tests or conference.

Finally, the authors considered whether the activities would occur in a scheduled manner, or require the instructor to be present. This would serve to identify areas which required synchronous technology. The instructor would not have to be present during homework assignments. The content for this area could be delivered asynchronously. The lecture component would require all of the students to go through the experience simultaneously. Because of this, a synchronous command mechanism among all of the students' applications was needed.

Lecture	Homework	Test
Comm. flow: Mostly one way Pace control: Instructor Scheduled: Yes Instructor Present: Yes	Comm. flow: Two Way Pace control: Student Scheduled: No Instructor present: No	Comm. flow: Two Way Pace control: Student Scheduled: Yes Instructor present: No

Table 1. Analysis of Classroom Activities

Architecture of the system

The Unified Multimedia Delivery Systems was built using a variety of Web and traditional CD-ROM development components (figure 1). Client applications written in Macromedia Director served as the front-end to the system for both the instructor and the students. These applications were responsible for presenting media to the instructor and/or the student based on the messages they received via TCP/IP. Shockwave Multiuser Server (SMS) was used as a messaging queue to distribute messages among the client applications. Using NetLingo in conjunction with the SMS, the authors were able to establish a virtual message loop among all of the client applications. Event handlers were then written to intercept events sent through this mechanism. This provided the control mechanism that allowed the Director application to behave with the flexibility of a Web application.

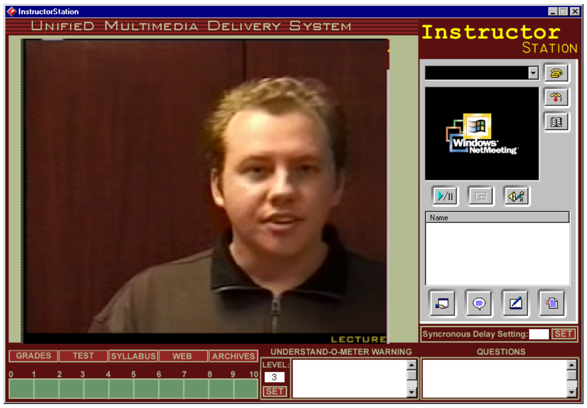

Figure 1. Screenshot of current UMDS (PC)

To deliver video, the authors came up with three alternatives. First, the authors provided a means to deliver extremely high quality video files on demand in low-bandwidth situations. Video files would be delivered to the client ahead of time compressed as MPEG-I, MPEG-II or Microsoft's MPEG-IV. Because the files would be on the client's machine when the presentation would occur, network bandwidth would not be a concern. A message from the instructor's client station forced the MPEG video clips to play on the student's stations. Using this mechanism the instructor could deliver full-screen full-motion video to the student application even under high traffic network conditions. Questions and student feedback were handled via text chat and an embedded NetMeeting Active-X object [10].

In order to use the video-on-demand feature, the authors implemented a mechanism for the system to update its local video files dynamically. Using the virtual messaging loop, the student's client applications reported the content of their local files to the server. The server then compared its content to these reports. If differences were reported, the server instructed the clients to download the latest files from an FTP server. This mechanism insured that most of the clients would have updated video files. An option that allowed either the instructor or the student to force an update was also instituted.

With the mechanism for video-on-demand implemented, the authors turned their attention towards implementing a system for delivering live video from the instructor to the clients and for sending video based feedback from the students to the instructor. To deliver high quality, high bandwidth video from the instructor to the students, the authors implemented a Microsoft Media Server. This system allowed the multicast of up to 3.0 megabit per second multicast video. The multicast option of the system allowed the stream to be sent to a virtually unlimited number of client stations without additional strain on the server. Three machines were needed to implement this system. An NT 4.0 server running Microsoft Media Server, another NT Server running Microsoft Internet Information Server, and a Windows NT workstation running Windows Media Encoder were required (see figure 2) [9]. To decode the multicast video stream, the authors embedded a Microsoft Media Player Active-X component, within the Macromedia Director client applications.

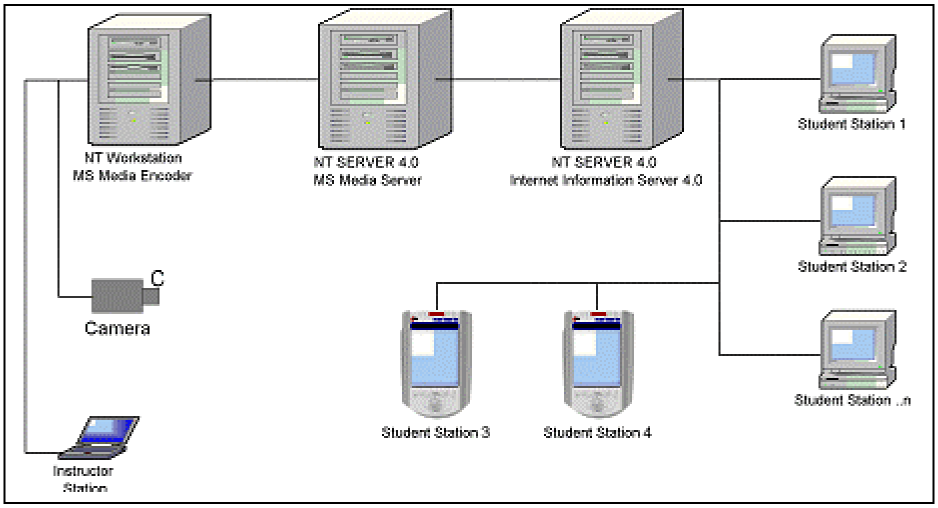

Figure 2. Live Video Streaming Based on MS Media Server

To deliver video feedback from the student station to the instructor, the authors embeded a NetMeeting 3.0 ActiveX object within the student's applications. This would allow the students to send video back to the instructor. The video transmitted by NetMeeting is lower quality and lower bandwidth intensive than the video transmitted by the Microsoft Media Server. Using this mechanism, the authors, in effect, had allocated most of the bandwidth for video from the instructor to the student, but still allowed for the capability of student-originated video.

The authors decided to use an embedded NetMeeting ActiveX component as the central piece in designing the conference and demonstration sections of the UMDS. The teleconferencing features of the component allowed the students and the instructor to communicate in a private environment. This became the mechanism students would use to attend "office hours." The application sharing and collaboration features allowed the instructor to demonstrate software techniques and assist the students with assignments. To allow the students to query their grades, the authors embedded an ActiveX component, which would embed a Microsoft Excel based grade book into a Web page, which could be viewed using the embedded Web renderer. Figure 3 details the internals components of the system.

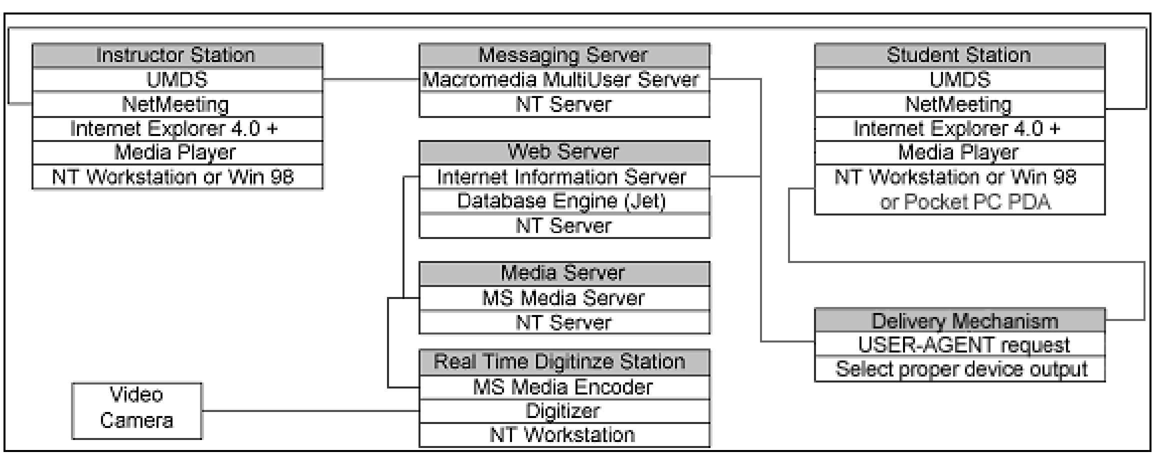

Figure 3. Components of UMDS

In addition, the authors provided a traditional web-based media depository, which would hold Windows Media Files of video content. This would be used for providing database-driven video-on-demand capabilities to the system.

Intelligence Behind the Video

To move towards the goal of an intelligent all-encompassing distance learning solution, the system would need to accurately determine how the user was accessing the media content. To reach this goal, the author defined a set of requirements and established a framework to deploy the solution. Because the primary use for the component was defined as an on-site reference for students, it needed to be rapidly searchable and accessible from multiple devices including laptops, desktop, and PDA's. The querying mechanism would need to provide speed not only on performing the actual search, but also in set-up. The authors decided to implement this component as an ASP web based application.

The web application needed to perform various operations. First, it needed to authenticate users both for security reasons and for content personalization. Based on the user's username and password, the system would decide if the user should be allowed to view any of the content and then decide how to format the content for maximum impact based on the settings for the user, the task, and the user's output device. The decision on how to format the content would need to happen at the authentication phase instead of being explicitly set in the user's database because the formatting would be dependent on rules unknown until run-time. The centralized nature of the database would also make it possible for the instructor to change any settings in the requirements for a task at any time. The user could also elect to access the data from a PDA in one instance and then from a laptop a few minutes later. In any case, the web application would need to keep the user's state and generate content from a generic set of rules queried at run-time, implemented by requesting the USER-AGENT of the student's device.

A relatively facile method of meeting the bulk of the requirements identified by the author was to rely on using Microsoft's Active Server Pages (ASP) for imposing application logic and for retrieving data from the database. An advantage of using ASP was the ease with which other server side components such ActiveX Data Database Objects (ADODB), and Windows Media Server could be integrated. The net effect was a virtually effortless development of the required web infrastructure.

Video component

The authors decided to construct the system around Windows Media Services to address the video aspect of the system. Windows Media is a set of tools freely available from Microsoft for Windows users that allows the streaming of video over the Internet at rates up to 3.0 megabits per second [6]. It includes components for both encoding live video events as well as streaming video files-on-demand. It can also encrypt video streams with a digital right manager, and includes features for delivery video as a unicast or a milticast stream. It was particularly attractive to the authors because it integrated extremely well with Microsoft's IIS server.

Each video file was encoded at three different bandwidths that can address users on 56k/sec dial-up, 128k/sec ISDN, and broadband connections. By employing windows media server, the system can dynamically adjust the stream sent to the user based on real-time available bandwidth. If a student accessed the system from a broadband connection, he would initially receive an extremely high-quality stream. If the connection speed to this student's site decreased due to network traffic, the media server would transparently degrade the quality of the video sent to the user. Thus, by employing Win Media Server, the system gains the ability to react to bandwidth changes in real-time.

To produce the video files, instructional events would be shot single-camera style and edited with Sonic Foundry's Vega Video. Vegas Video's ability to insert indexing markers into the windows media file and to encode the video at multi-stream files made it the only plausible choice for video creation. Window Media allows the inclusion of a script channel, in addition to the standard video and audio channels, which can be used to trigger events such as text display or redirection to URLs [8] . This allowed the author to insert commands into the video sequence at specific points where accompanying illustrations should be presented to the student. Video for laptops and PC's would be encoded at 640 pixels by 480 pixels. The video meant for PocketPC's would be restricted to a maximum resolution of 240x320.

When the user asks for media on a specific subject, the system queries the database for all video files that have the same instructional objectives. Then, it reads the USER-AGENT of the incoming request from the user. With this information the system is able to determine if the user is on a PocketPC, or a desktop computer, along with the browser environment and software version. This information is used to determine the size and bandwidth of the video file to serve to the user. Figure 4 illustrates how the system would present the same video content to users on two different devices.

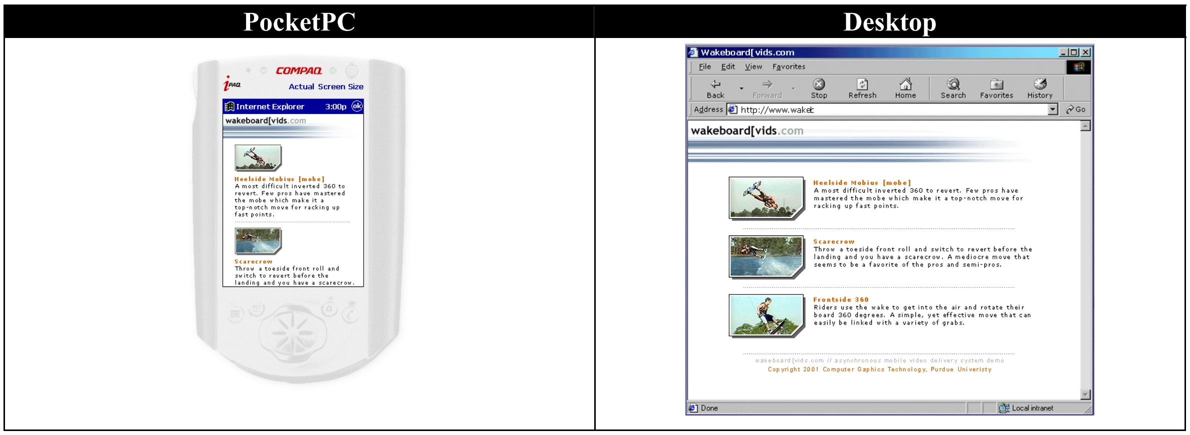

Figure 4. Wakeboard Content on Multiple Devices

Wireless Video Expansion

With a mechanism for retrieving an appropriate video file for the user, the authors turned to implementing a system to allow learners to access the system via wireless transmission using PocketPC's. Delivering the instructional content from the web to the learner's wireless devices was accomplished through the use of IEEE 802.11 wireless networking cards. Each of the testing laptops and PocketPC's was outfitted with one of these cards. A desktop system was outfitted with an IEEE 802.11 access hub and a standard 100-megabit NIC, which connected to the network infrastructure. This machine served a bridge between the Internet and the wireless devices. Each card provides a theoretical bandwidth of 11 Megabits per second, which was more than sufficient for delivering full-screen full-motion video, but only had a range of one hundred fifty feet. Multiple access ports can be placed throughout an organization to provide a wireless access area beyond the 150 feet restriction of a single access hub. Within the confines of the development phase of this project a single access hub was used. It worked as expected. The author found the maximum sustained bandwidth to top out at approximately 7.6 megabits per second.

Path-Finding Methodology

Finally, in order to provide the user with a unique instructional experience tailored to both his/her device and learning preferences, the use of artificial intelligence algorithms has been added to systematically and dynamically select the instructional content. The authors considered using bredth-first, depth-first, and hybrid algorithms such as path-intensive A* Star, which guarantees the shortest path to a correct solution. In a leraning environment, the instruction system attempts to guide the student through the content with the fewest possible communication static and confusion. Therefore, a method such as A* Star can dynamically lead a user down a tailored learning path, avoiding unnecessary instruction and providing the student with proper assistance and content.

An example of this tailored experience deals with instructional media. A student that the system categorizes as a visual learner will be provided mainly with video and graphics-based instruction. On the other end of the spectrum, a user that is having problems with the visual content, will be given sound files and on-screen text. This is determined through functions present within the Active Server Pages that pass variables through the A* Star algorithm and are compiled in a user-specific Microsoft Access database.

Conclusion

While the new component proved successful in expanding the capabilities of the UMDS system to include intelligent delivery of video content to both desktop computers and PocketPC PDA's, the UMDS requires additional development work before it lives up to authors' expectations of delivering any instructional media under any circumstance. As the system currently stands, it can be used for traditional distance learning. It provides mechanisms for tests, lectures, quizzes, live-video, video-on-demand, grading, etc. In short, the system can deliver just about any type of media to just about any device. But, that is not enough.

The next phase in the development of the UMDS will need to be the full integration of a learner model and task model to which artificial intelligence principles will dynamically restructure the system as needed, based upon the initial and average success of the student.

Bibliography

[1] Adaptive Web Navigation for Wireless Devices (2001) [On-Line Available: http://www.cs.washington.edu/ai/proteus/ijcai01.pdf

[2] Microsoft Mobile Devices Developer Tools (2001) [On-Line] Available: http://www.microsoft.com/mobile/developer/BeginnerDev/tools.asp

[3] Microsoft SQL Server 2000 Product Specifications. (2001) [On-Line] Available: http://www.microsoft.com/sql/

[4] Microsoft Win CE 3.0 Hardware Product Specifications. (2001) [On-Line] Available: http://www.microsoft.com/mobile/pocketpc/bguide/default.asp

[5] Microsoft Win CE 3.0 Software Product Specifications. (2001) [On-Line] Available: http://www.microsoft.com/mobile/pocketpc/bguide/features/ppc/pie.asp

[6] Microsoft Window Media Specifications. (2001) [On-Line] Available: http://www.microsoft.com/windows/windowsmedia/en/default.asp

[7] Personalizing Web Sites for Mobile Users (2001) [On-Line Available: http://www.cs.washington.edu/ai/proteus/www10.pdf

[8] Sonic Foundry Vegas Video 2.0 Specifications (2001) [On-Line] Available: http://www.sonicfoundry.com/products/NewShowProduct.asp?PID=280

[9] Starband Services (2001) [On-Line Available: http://www.starband.com/faq/starbandfacts.htm

[10] Viewpoint Metastream Architecture Specifications (2001) [On-Line] Available:http://www.viewpoint.com/create/index.html

CARLOS MORALES
Carlos R. Morales is an assistant professor of computer graphics in the Department of Computer Graphics Technology at Purdue University. He holds a BA in Telecommunications and an MS Ed. in Curriculum and Instruction. Prior to working at Purdue University, Carlos worked as a Technical Director. Some of his clients have included Microsoft, Chicago Bulls Organization, First Alert and Brach's Candies. His research interest includes distance learning, animation, and multimedia development.

CHARLIE MILLER
Charles D. Miller is a graduate student and teaching/research assistant in the Department of Computer Graphics Technology at Purdue University. He holds a BS in Technology with minors in Photography and Organizational Leadership/Supervision. Charles has been developing content management systems for clients such as the Metropolitan Planning Council, the Management Association of Illinois, and the Campaign for Sensible Growth. His research interest includes artificial intelligence, dynamic data visualization, and mobile web advancement.