A Method for Delivering Dynamic Hyperlinked Streaming Instructional Content Over a Wireless Environment Through the Use of Active Server Pages and XML Technology

Carlos R. Morales
Purdue University

Abstract

There is a tremendous need for delivering video based instruction and supportive documentation in the form of schematics, illustration, and animation to students outside of the confines of their traditional classroom or laboratory environment. Many solutions address the delivery of training materials using Internet technology, but currently, there is no commercial turn-key solution for delivering streaming video content over a wireless IP network. This paper details a method for implementing a system capable of delivering full-screen, full-motion streaming video as well as hyperlinked context sensitive illustrations and schematics to learners over an 11-megabit wireless network. The net result is a system that can serve as an instructional reference to learners while they are engaged in active on-site problem-solving activities. The benefits to the instructor and to the learner are many. It lessens the need for the learner to recall all of the information presented to him, because the materials will be available on-demand for reference at a later time. This minimizes mistakes. It also centralizes instructional and supportive materials into a single system that can be easily updated to reflect changes. Any changes in procedures are immediately reflected in the system and available to learners. Because materials are delivered over the web, the method can be used for distance training.

The author uses traditional video techniques in combination with dynamic web development techniques to develop and deploy dynamic streaming content. Windows Media Encoder is used to create streaming video files of instructional events. The files are edited and indexed by subject and by logical segments within each file. This allows learners to not only look for a video file on a specific subject or procedure, but also for a specific step or sequence within a video file. Descriptions of the files are logged and indexed using an SQL based database and put on the web using Microsoft Internet Information Server. Active Server Pages and XML are used to query the files from the server and present the information to the learner. Supplemental illustrations are constructed using video compositing techniques and encoded to MPEG-4 for delivery. Learners access the materials using a customized web page on a laptop at their point of work. IEEE 802.11 cards are used carry the requested video from the server to worker’s laptop. The net result is a system capable of administrating instructional materials to the learners in an extremely dynamic environment.

I. Introduction

It is extremely beneficial to deliver personalized instructional content in the form of video, schematics, illustration, and animation to learners outside of the confines of the traditional classroom. This paper details a method for presenting personalized instructional materials in a wireless environment through the integration of data bound web technology and traditional video techniques. The result is a system capable of personalizing content for the user in a just-in-time learning environment.

This approach is particularly beneficial because the integration of web technology and streaming video techniques allows the delivery of high-quality personalized content. By encapsulating task specific steps into a database model and then integrating the information derived from that model with data specific to the learner, the proposed method can alter the presentation of the content to be most effective. The use of web technology further strengthens the system by forcing all of the content to be housed in a central location that can be accessed from anywhere. Any changes made to the instructional content are immediately disseminated to all learners. Actions of other learners on the system are also available. This is particularly helpful for group situations where the actions of one student can affect the situation for another student. The immediacy and flexibility of the system makes it an ideal on-site reference tool.

To reach this goal, the author defined a set of requirements and established a framework to deploy the solution. If the primary use for the system was defined as an on-site reference for students, then the system would have 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. Another requirement would be the need to maintain all of the data in a centralized location while allowing access from wireless devices. This immediately discounted the possibility of using CD-ROM or DVD-ROM technology as our primary medium.

Because the system was primarily meant to be used on-site over a wireless environment, it would be primarily accessed using laptop computers or handheld PDA's. This would affect the type of video delivered by the system. While laptops can easily decode a video stream encoded at 640 pixels by 480 pixels, most PDA's would be restricted to 320 pixels by 240 pixels.[1] Another difference would be the method available for delivering illustrations in the form of vector graphics. While laptops would have the capabilities to read standard vector graphics and rasterize to any needed resolution, most PDA's would not have this capability. A third difference would be the use of three-dimensional content directly on the client side. Again, laptops would be able to do this effortlessly, but the majority of PDA's would not.

To account for these discrepancies, the author decided to restrict the system to only work with PDA's based on Windows CE 3.0 such as PocketPC or HandheldPC. This combination would allow the author to use the freely available Microsoft Embedded Visual Tools. This set of tools includes a version of Visual Basic and Visual C++ specifically for developing software for PDA's.[2] The author found it particularly attractive because of its integration with the standard version of MS VB and MS VC++.

For delivering illustrations and diagrams the author decided to use JPEG images. While not as flexible as using resolution-independent vector images, it would allow the author to use the same media on both laptops and WinCE 3.0 PDA's. All WinCE 3.0 devices include a version of Pocket Internet Explorer, which can render web pages as well as GIF and JPEG images.[3] A negative aspect of this approach was that the server would have to house multiple versions of the same JPEG image at different levels of magnification. If the learner asked to see the same diagram but a different level of magnification, he would be served an entirely different JPEG image. This would be easy to implement, but not the most efficient way of implementing this feature.

Differences in video resolution between the two platforms affected the page layout of the web pages. The screen on the majority of PocketPC devices was restricted to a resolution of 320 pixels by 240 pixels, while laptops would have a resolution ranging from 640 pixels by 480 pixels up to 1280 pixels by 1024 pixels. In generating the web pages for both devices the author decided to create separate pages for both devices that would be constructed from the same data. Two separate sets of pages would have to be maintained, but the data that would dictate the content would be centralized.

II. Dynamic structure

In anticipation of having to deliver to a wide variety of devices, the author decided to format the information within each of the database records as XML. This would allow the data to be easily extended through the use of devices such as namespaces, and make the data highly portable. Not only could web servers parse the data, but also standalone applications written in Visual Basic, Macromedia Director, etc. The decision to explicitly write XML into the records instead of translating the recordsets to XML on the server or even using an XML enabled database management engine such as Microsoft's SQL Server 200 was made to insure speed and portability.[4] Translating to XML on the server would have slowed down the searching operation. Using a database that returns XML directly would have restricted development to the most expensive database management system currently available.

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 users 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.

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 the MS XML Parser could be integrated. ADODB allowed the application to access any ODBC compliant database management system such as MS Access, Oracle, DB2, or Informix. The XML parser provided the capability of loading the XML data into the Document Object Model and manipulating at run-time. This was beneficial because it would allow the system to reformat the data for any unforeseeable device used in the future without requiring the reformatting of the data housed in the database. The net effect would be future room to expand without penalty. ASP also provided the mechanism for performing searches and archiving. The net effect was a virtually effortless development of the required web infrastructure (See Figure 1).

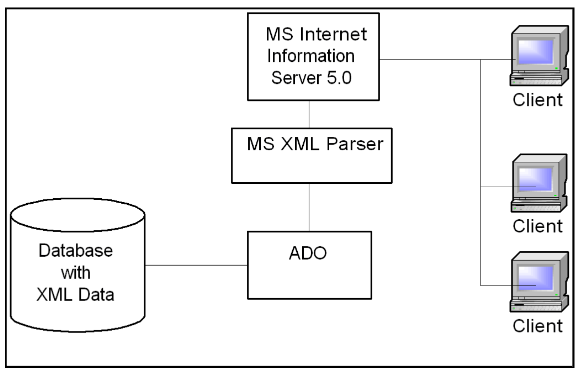

Figure 1. System Overview

III. Video component

To address the video component of the system, the author decided to construct the system around Windows Media Services. 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.[5] It includes components for both encoding live video events as well as streaming video files-on-demand. It can also encrypting 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 author because it integrated extremely well with the IIS server.

Primarily the system would be used to deliver pre-recorded video-on-demand clips of instructional content. To produce the clips, the instructional events would be shot single-camera style, illustrations would be animated using Adobe After Effects, and Sonic Foundry's Vega Video would then be used to integrate the illustrations with the live video. Vegas Video would further be used for indexing the video and encoding in a multi-stream Window Media File.[6] 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. This allowed the author to insert commands into the video sequence at specific points where accompanying illustrations should be presented to the learner. Vegas also allows a single video file to contain multiple streams at varying degrees of compression and target rates. Thus a single file could contain the same content encoded for a 56-kilobit dial-up connection and a 250-megabit version for a broadband connection. Users would access the same file, but receive the stream appropriate to their connection speed. This functionality required the files to be served from a Microsoft Media Server instead of from the Microsoft Internet Information Server.

While the primary intent of the system was the delivery of pre-recorded video content, the author also accounted for the need to stream live video in real-time. This would be beneficial when the instructor wanted to address all of the users of the system simultaneously or when there was some urgent information that needed to be disseminated to the users immediate. For this scenario, instructional events would be shot using multiple cameras that would be fed into a video switcher. The output from the switcher would then be fed to an encoding machine, which housed a Osprey-100 video encoding card. As the event was being shot, the operator of the encoding machine would be able to switch among the video streams (See Figure 2). A single multicast stream would be sent to all of the recipients. This would minimize the bandwidth load on the server, as it would not have to support a separate stream for each user. A negative aspect of this system would be a lack of control of scripting events. In a pre-recorded scenario, the editor would be able to meticulously place events into the script channel of the stream, but during a live event this would also need to occur live. If an event was placed incorrectly, it could not be undone, because the event would have already been transmitted.

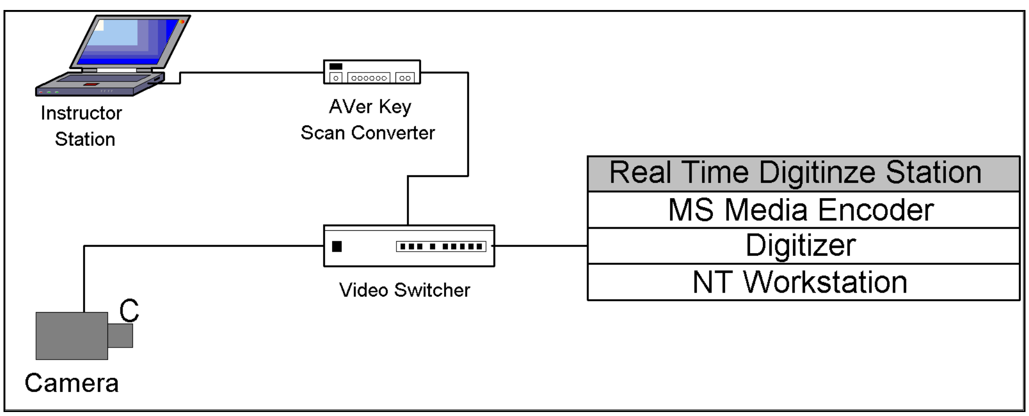

Figure 4. Video Capture Mechanism

The decision to utilize Microsoft's Media Services for video transmission necessitated additional hardware on the server side. Two additional machines were needed. One machine housed a video encoding card and ran Microsoft Media Encoder. This machine was dedicated to digitizing the video and encoding it into MPEG-4 during live video transmission. A second machine ran Windows 2000 Server and Microsoft Media Server. This machine was responsible for multicasting video and for intelligently delivering multi-stream video to the clients.

IV. Data encapsulation

Next, the author focused his attention on the specifics of how to encapsulate the data and how to present it to the users. The author wanted to provide the users all of the pertinent data needed to accomplish his task without overwhelming him. First the author decided to include textual step-by-step instructions. Textual information would be encoded into memo fields in a database. This would allow the information to be easily searchable. Second, the user would be presented with a diagram or schematic of the process. Initially this would be in a JPEG raster format because it could easily be viewed on all devices including wireless PDA's. Third, the page would contain a pre-rendered video/animation of the entire task. Video/animations would be encoded in MPEG-4 at a resolution of 320 pixels by 240 pixels at 500 kilobits per second for laptop users and at 240 by 180 at 100 kilobits per second for PocketPC users. Audio would be included in the video/animation file. Laptop users would have the added option of receiving, when appropriate, a 3D model that could be manipulated in real-time.

The model would contain all of the parts required in the process and provide the user with the ability to view the process in motion. The purpose of providing the real-time renderable model was to clarify any spatial or procedural questions. The three-dimensional real-time content was encoded using Metastream's 3D format for numerous reasons. First, the format provides extremely high quality at relatively low-bandwidth. Second, it reads created in an industry standard packages such as 3D Studio Max. Third, it allows the inclusion of animation information that the user can trigger at run-time while allowing the user to manipulate the camera. This was important because incremental animations could be inserted into the file, which the user could trigger while manipulating his vantage point. Finally, the entire format was built around XML technology. This allowed our system to modify the attributes and content of the files easily though the MS XML parser.[7] Combined these elements would provide the user with a tremendous amount of reference information (see Figure 2). The only drawback would be the danger of overwhelming the user with information.

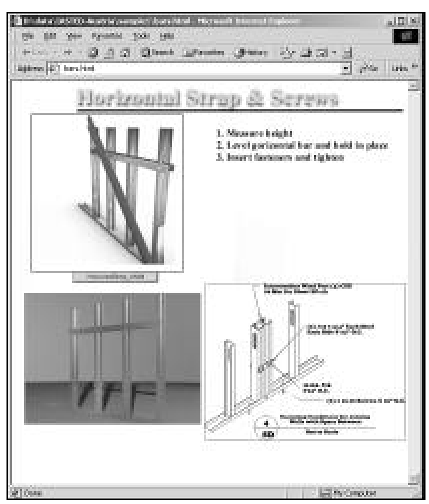

Figure 2.

Sample presentational page

V. Intelligence behind presentation

While the system was designed to allow personalization of content based on user attributes, it became apparent that the definition of which user attributes played a part in deciding content availability would differ in most situations. The author decided to focus on a mechanism that would provide a flexible way of defining those attributes. The most challenging aspect was devising a structure that would allow the instructor to insert an unlimited number of rules into the system. This was accomplished by employing a meta-data table that would describe the rules and assign weight values. Each operation would be defined with an ideal set of conditions and then weights would be defined providing the system with a way to prioritize the requirements. For example, an operation might specify an operator with a grade of A on a certain test and a grade of B on previous lab assignment. By defining minimum level of acceptability and providing the system with a way to weight those levels, the system could be instructed to show specific supplemental materials if the learner was deficient in the overall acceptability level by more than 25 percent. For group situations, the system could identify which group members would best be suited for individual tasks based on past performance.

VI. The Last Mile

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.

VII. Future plans & conclusion

In the end, the system provided the necessary infrastructure and media delivery capabilities to operate as an on-site reference tool for learners, but could be improved in a number of areas. The current method for personalizing user content is based on a simple threshold method. This could be improved through the inclusion of an artificial intelligence engine. The system could also be improved by taking into account users on wireless modem dial-up connections. This would expand the system to be used virtually anywhere. Bandwidth limitations could be overcome by providing a local cache for content on the wireless devices. Cached hits would result in high-quality content, and hits out side of the cache would result in the appropriate content at a lower quality.

Finally the system could be integrated with wireless satellite technology such as StarBand's newly released bi-direction internet access.[8]

In continuing to develop this system, the author plans to focus on integrating wireless bi-directional wireless satellite technology and to conduct usability tests with a variety of students. The end result will be a robust commercial-quality system for delivering training on the field.

Bibliography

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

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

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

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

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

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

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

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

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.