December 1998

# ActiveBook: A Niche Browser as an Education Support Tool

Sameer Verma
*Georgia State University*

Mihir Parikh
*Polytechnic University*

# ActiveBook: A Niche Browser as an Education Support Tool

**Sameer Verma**
Decision Sciences
Georgia State University

**Mihir A. Parikh**
Institute for Technology & Enterprise
Polytechnic University

## Abstract

*The Internet has been an engine for innovations in many fields. This paper describes an innovative way to utilize the technologies and capabilities of the Internet to support activities of educational environment. The paper proposes a niche browser which employs a hybrid model of traditional web-based "pull" technology and emerging "push" technology. The browser as a framework goes beyond a simple software at client level. It is a complete solution with student client, instructor client and server level software.*

## Introduction

In the last few years, the Internet has rapidly expanded from its original territory, academia, into the business world. Many people now regularly use the Internet to access ocean of information through World Wide Web, the graphical interface of the Internet, and to communicate with friends and other people with like interests. The Web is based on the "pull" technology where the recipient of the information browse the sources on the Web and *pull* the information.

Pulling information through browsing has many disadvantages. One crucial disadvantage is that the information provider is never sure about whether the recipient has received a critical, time-sensitive information on time or not. An emerging Internet technology called "push" technology has made it easier for the recipients to receive the information they need on regular basis in time. Push technology is used to automatically provide news, latest stock prices, weather information, software updates and much more. In this paper, we propose to use both push and pull technology to develop a niche browser which can be used as an education support tool.

## Education Support

In traditional classroom-based education, three types of activities are performed by the students: Pre-class, in-class, and post-class. Pre-class activities include getting the outline of the next class, assignments for the next class, prepare for the class, and act on any other specific information provided by the instructor. In-class activities include interacting with the instructor and classmates, learn concepts, apply concepts to cases, and fill in the missing links in their knowledge. Post-class activities include review lecture notes, reinforce their knowledge, communicate with the instructor for clarifications, and review instructor feedback. In-class activities are supported and monitored by the instructor, but pre-class and post-class activities require a support structure which the student can utilize remotely, from home and/or office.

Building on the wide spread use of Internet, we have developed a framework of ActiveBook. ActiveBook is a niche Internet-enabled browser based on a hybrid push-pull model. It is designed and developed to support students activities in traditional learning environments. It uses state-of-the-art Internet connectivity and intelligent software that surpass support provided by simple HTML-based Web pages. The framework of ActiveBook is generalized such that specific ActiveBooks developed from the framework can support specific course requirements of different courses.

We have developed and utilized two specific ActiveBooks based on the framework: WebSIMQ to support a simulation gaming course and WebSTAT to support an introductory business statistics course. The simulation gaming course requires heavy interactions (both communication and transfer of files) between the course instructor and students and teams of students, working as competing firms in a business simulation game. The business statistics course requires students to spend a considerable amount of time in pre-class and post-class activities to acquire and reinforce their knowledge in statistics. Although a discussion of the results of our experiments with the systems is beyond the scope of this paper, the initial results are very encouraging.

## ActiveBook: A Niche Browser

Figure 1 shows the architecture of ActiveBook. The architecture is designed to provide remote help in organization of course material, assist communication between the instructor and students, and use an intelligent interface to provide dynamic delivery of course content. As shown in the architecture, ActiveBook uses a (or multiple) server(s) to store protocols for Internet-enabled communication and databases of course information and material. ActiveBook has two types of clients: one used by the students to access lecture notes, assignments, schedule, etc.; and the other used by the instructors to update the information on the server. The specific features provided by ActiveBook are:

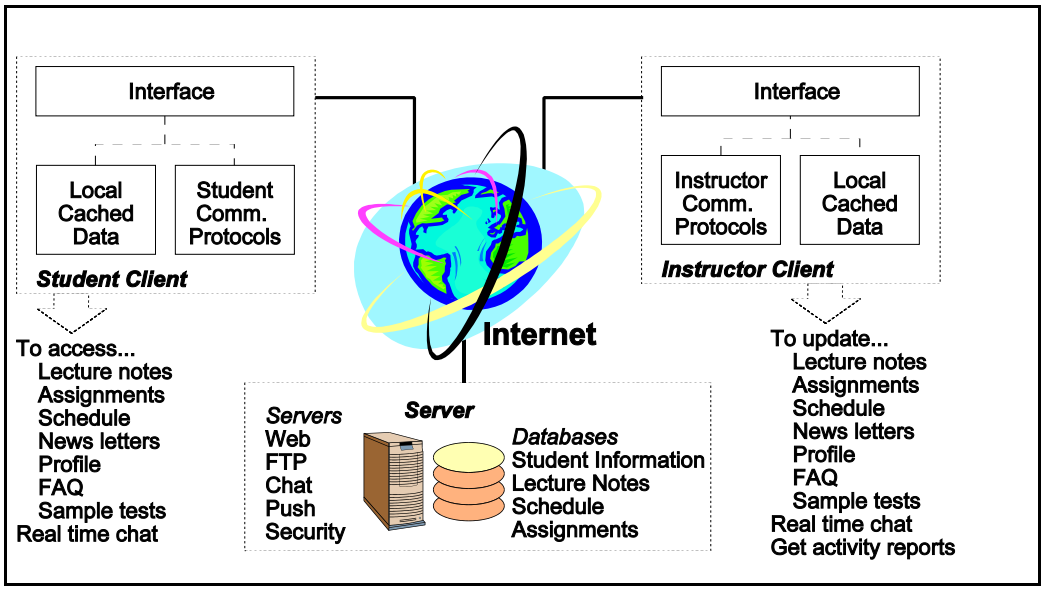


**Figure 1. Architecture of ActiveBook**

### *Interface*

ActiveBook provides a very easy-to-use interface. The opening home screen is shown in Figure 2. This screen provides a direct and quick access to all features of Active Book. The features include Lectures, News Letters, Assignments, E-mail, Frequently Asked Questions, Chat, Help, etc. The features can be easily added or removed to match specific course requirements. For example, in the simulation gaming course, transfer of historical data files, spreadsheet based models, software, and submission of decisions/getting results are specific course requirements. These are handled separately through an add-on module.


**Figure 2. Home Screen**


**Figure 3. Lectures**

### *Class Lectures*

Class Lectures are accessible from the Lecture screen (Figure 3). The lecture notes can be made in Microsoft PowerPoint®, WordPerfect Presentation®, or any other commercial presentation software. The notes can directly run through the software's slideshow viewer or runtime software as required by the vendor. The slideshow employs a full screen display, with transitioned screens and bullet points that roll out individually with visual cues. This provides full freedom to the instructors in choosing the presentation software of their choice. However, for students it provides a one-point-and-click, guaranteed access to all class lecture notes. If a student does not have the runtime software, the required runtime version can be provided to be downloaded and installed on the local computer through ActiveBook.


**Figure 4. NewsLetter**

### *News Letter*

News Letter supports two "newspaper-like" columns (Figure 4): *Message* column and *Flash* column. The class sessions are summarized in the *Messages* column. Agenda for the next class is also published here. Any class related concerns are also addressed through *Messages*. This broadcast helps in setting the students expectations before class, and provides some guidance to the students about the sections to be covered from the book or the cases to be examined. It

also provides a synopsis of lectures *after* the class. The other column, *Flash* column, relates to the software upgrade notices and other ActiveBook administrative issues. *Flash* also displays announcements about changes in classrooms, class topics, schedule, etc.

Other features such as Assignments, Schedule, E-mail, Chat, Frequently Asked Questions, Course Related Links on the Web, etc. are provided in the similar fashion as described in the above sub-sections on Lecture Notes and NewsLetter

## ActiveBook Technologies

ActiveBook utilizes many emerging software technologies at different levels to perform different functionalities. The technologies are summarized below.

| | | |
|---|---|---|
| Networking | Student and instructor clients | Asymetrix ToolBook |
| | Web and FTP servers | Microsoft Windows NT, UNIX, AIX |
| | Chat server | Asymetrix ToolBook |
| | Push server | Asymetrix ToolBook |
| Course Material | Lecture notes | PowerPoint, WordPerfect |
| | News letters | Rich text format |
| | Assignments | Word, WordPerfect, Rich text format |
| | Schedule and syllabus | Asymetrix ToolBook |
| Software Setup | Student and instructor clients | InstallShield guided self-extracting executables |

As shown in the above table, the course material files come in a variety of formats such as ASCII text, Rich Text Format, executables, worksheets, and image formats. ActiveBook can download all these formats equally well. Another advantage is the intelligent access control on client activity. For example, if data files have to be downloaded to a three and a ½-inch floppy disk, it will instruct the student to do so. This feature is particularly helpful because some of the spreadsheets are sensitive to location of data files. The students are oblivious to the format of files, or the applications needed to run them. The students are not required to download any viewers or plug-ins. They do not need to keep track of any Website location address or any Website log-ins. They also do not have to face any browser ambiguity (like choosing from Netscape, Internet Explorer, HotJava, etc.), because ActiveBook itself is a browser specifically designed for education support. ActiveBook's universal interface introduces a sense of familiarity and stability to the application as opposed to Websites that change drastically overnight. The organizational structure of ActiveBook also encourages the instructor to organize material and provide it to the students in a structured way.

## Directions for Future Development

The above described systems are fully functional. However, more functionalities could be added in the future. Here are some suggestions.

The systems currently download new information automatically when the student opens the software. This does not consider bandwidth availability at that time. During peak hours, depending on the size of the new information, the download could take some time. In addition, this may put extra burden on the memory and processing power of the client computer. Although this has not caused any major problems, the new version could be improved to have a function where the new information could be streamed during off-peak hours when net traffic is relatively less or when the user is not using the system.

The systems simply work as an intelligent tool for information delivery. They do not adapt to the knowledge level of the student. In future, the system could have intelligent agents which identify the learning needs of the student. The agents then could search databases on the domain knowledge and extract and present the information that matches the needs.

### *References*

References available from authors upon request.