第三屆即時與媒體系統研討會
Third Workshop on
Real-Time and Media Systems (RAMS' 97)
台北市南港中央研究院
1997. 年 7. 月 30、31 日

A Dynamic and Flexible Presentation System Based on HTML

Yao-Nan Lien * *and Shanon Jong* **

* Department of Computer Science, ** Department of Advertising
National Chengchi University
Taipei, Taiwan, R.O.C.

Abstract

Traditional presentation systems require considerable resources and preparation time to support presentation facility and material preparation. A high quality low cost personal computer hardware and software significantly reduce the required resources and preparation time. However, popular authorwares still have plenty of room to improve. This paper will describe a more dynamic, more flexible presentation system and an open authorware architecture based on HTML format to facilitate fast slide authoring. The proposed system will be affordable by most presenters in terms of environment, hardware, software, authoring, and slide management.

1. Introduction

Traditional presentation systems require considerable resources and preparation time to support the presentation facility and material preparation. A high quality low cost personal computer hardware and software significantly reduce the required resources and preparation time. For instance, a notebook computer, a portable LCD projector, and a Microsoft's Powerpoint presentation authorware would constitute a personal presentation system. However, popular authorwares such as Powerpoint still have plenty of room to improve. Although Powerpoint has already offered a convenient WYSIWYG authoring environment, a user still needs some training to become skillful. Furthermore, there are some limits with its presentation environment which will be detailed in Section 2.

In this paper, we describe a more dynamic and flexible presentation environment, and an open authorware architecture that supports faster slide authoring based on HTML format. The resources required for environment, hardware, software, authoring, and slide management will be significantly reduced.

1.1 Affortable Presentation Environments

There is a wide variety of presentation environments, ranging from multi-million systems suitable for public addressing to the simple overhead projectors only good for transparency. This paper focuses on the environments that are affordable by most of common users in the near future. A typical system consists of following components:

- a multimedia Pentium class notebook running Microsoft Windows 95;
- a HTML Browser such as Netscape Navigator;
- a track-ball-type wireless mouse;
- an optional network connection;
- a pair of optional external powerless speakers.

Normally, we expect a projector that accepts VGA signals from the notebook and projects its screen image to the wall. An alternative is to have a notebook with an openable back cover and also a LCD screen which can be put on the top of an overhead projector. The overhead projector can project the image on the LCD screen of the notebook to a wall-screen. Such a notebook is referred to as a *projection notebook* in this paper. Such a minimum system can be obtained under NT $100,000, and hence is affordable by many presenters.

The size of the LCD screen on a current notebook is about 8" to 12" in the larger of the two dimensions. The wireless mouse is equipped with a track-ball plus some buttons to emulate a desktop mouse. Usually, it has to be operated single handed. The limited size of the screen and the special single hand manipulation of wireless track-ball-type mouse become the greatest obstacles in designing a comfortable presentation system. More specifically:

1. It is difficult to manipulate a pointer.
2. It is difficult to perform any operation that requires both hands working together on a wireless track-ball mouse. For instance, pulling the scrolling bar or doing a drag-and-drop operation requires both hands to perform.
3. The LCD screen is too small.

1.2 Desired Properties

Most traditional presentations are just sequences of slide shows. Occasionally, some slides that are not in the predefined sequence would be shown in answering on-the-scene questions. An out-of-sequence slide show is quite time consuming and distractive in a presentation,

thus it is always kept in the minimum and is discouraged. In fact, a good presentation is not just presenting all materials in hand sequentially, especially in a classroom where there is an intensive interaction between speakers and audience. A presenter might need to present his/her materials in an unexpected order and might have to access to different sets of materials dynamically. For example, one might have to access to the course materials he/she uses in the Database class, to the students in the Basic Computer Concept class when he/she needs to explain what a database is. It doesn't take a Computer Science Ph.D degree to realize that, a hypermedia system with all course materials stored as a multimedia database in a notebook is a perfect solution to meet this requirement. However, a multimedia database doesn't land itself a good design automatically. It takes efforts to produce a good presentation. Besides, it is impractical to expect a multimedia database installed in a notebook system. Even if it is possible, not every user can be a good DBA (Database Administrator). In fact, handling Windows 95 has already been such a nightmare to most of notebook users. The complexity of multimedia database administration is even more complicated than handling Windows 95 by at least one order of magnitude.

All ideas presented in this paper are all implementable using current technology and the resulting systems are usable by a common user without a long training period. In the rest of this section, we will discuss the properties that an ideal low-cost presentation environment must have.

For simplicity, we assume that all presentation materials are organized into pages of hypermedia slides linked together by hyperlinks. Further, we assume all pages of slides are stored in one notebook although it is very easy to generalize to the entire Internet.

1. The document sources of slides must be open to various viewing and authoring tools. In other words, the source documents of slides must be viewable and editable by generic tools. They must not require any specific software package to do the viewing and authoring tasks.
2. The document and presentation system must be platform independent.
3. Information area in each slide should be as large as possible.
   On each slide, some areas may be reserved for navigation purpose, such as a button labeled as "next" which can be clicked to switch to the next slide in a predefined sequence. The left area can be used for presenting the information.
4. Navigation within the network of slides must be convenient.
   This includes the navigation within the same set or across different sets of slides in any order as quick as possible. In other words, it is desirable to minimize the number of links from any slide to another. Further, it is desirable to minimize those operations that require both hands to manipulate the "track-ball" type mouse. (For instance, a drag-and-drop operation needs both hands to operate the "track-ball" wireless mouse.
5. The multimedia capability of the system must be fully utilized.
   For instance, it must be able to present graphics and images dynamically, it must fully use the colors and sound effect.
6. Authoring must be as easy as possible.
7. The process of design-present must be as short as possible.
   For instance, WYSIWYG is an ideal authoring process. (However, it might not be the best choice when we consider about other design goals.)
8. The collection of slides must be easy to manage.
   A speaker may have to perform many presentations in his/her career. These presentations in some certain degrees are related. One would not be fool enough to author every set of slides from scratch. In the traditional way, one might have to ask his/her assistant(s) to organize all his/her slides and spend at least several hours to reorganize the slides before and after each presentation. After several career changes or reorgs, he/she might have totally lost in boxes of slides.

Some of these desired properties are contradict to each other. For example, one may design an easy-to-navigate presentation software that uses more than 50% of the slide area to lay out navigation buttons and allows only half of that for information presentation.

2. Current State of Arts

Some off-the-shelf products such as Microsoft's Powerpoint seem to be able to fulfill all we need. Unfortunately, it is far from perfect. This section will discuss their drawbacks based on Powerpoint as a reference product.

2.1 Proprietary Document Formats

The slides produced by most presentation systems can only be authored and viewed by a specific authoring tool. It won't be a big problem if a user only uses his/her own computer for presentation. Unfortunately, this is not the case in general. From time to time one might have to use other people's computer to do the presentation.

As described in Section 1.2, document sources had better be open to various viewing and authoring tools. For instance, we can use any editor to edit any ASCII based documents, such as troff, TeX, and HTML. One may argue that users of these document systems must learn another language(s) to process these documents so that

there is no advantage over a proprietary document system. However, the editor is one of the most frequently used tools in one's life. Using one's preferred editor will maximize his/her editing productivity. Furthermore, it is not too difficult to develop some public or personal middleware to relief the need of learning various processing tools. As we can see from latter sections, we will present a middleware that can help eliminate the need to learn a complicated language. Such a middleware can be easily implemented using popular script languages such as shell and perl.

Finally, a proprietary document format and presentation system may not be platform independent.

2.2 Slide Navigation

Most current presentation software tools were designed with an assumption that all slides are mainly presented in sequence so that they are designed to be very convenient to move forward and backward. Some of them may also offer non-sequential navigation capability. However, out-of-sequence moves are not as convenient. One common practice is to let the presenter point to some point in a sliding bar or type in the page number. However, presenters can hardly know exactly the page number they want to "jump" to. It usually takes several try-and-errors to locate the target slide. In this sense, unexpected out-of-sequence presentation is very difficult.

Furthermore, presenting slides in different sets is extremely difficult, if not impossible. One has to launch another Powerpoint session or, in the same session, open another "file" that contains the new set of slides. It usually takes several minutes to do so even on a high speed personal computer. One could seldom tolerate such a delay in most presentations, even in a slow pace classroom.

In summary, most current presentation software products are designed for deterministic presentation sequence.

2.3 Multimedia Capability

Most presentation systems can easily incorporate images or even videos into their "slides". Unfortunately, including dynamic graphics into a presentation is considered a luxury expectation in terms of required authoring efforts. On the other hand, the video and audio information requires little or no programming effort. But their use must be limited because they are huge resource drains.

2.4 Slide Authoring

Even though graphical user interface is quite user-friendly, users still require some training to be skillful in using any of such software tool, especially for advanced authoring. Except for some skillful computer professionals, most speakers still rely on their assistants to prepare most slides. However, lots of future speakers who have no luxury to have assistants are demanding computerized presentation. Thus, there is a need to offer these people an affordable authoring tools to author their own slides without an intensive learning process. It would even be better if they can use their preferred basic editing tools such as word, vi, PE2, etc.

2.5 Slide Management

Management of slides involves slide viewing (to know the content of slides), editing, reorganization, classification, duplication, unification, merging, splitting, etc. There is no much difference from paper management. (After publishing 40 or 50 papers, a university professor would definitely face a paper management problem unless those papers are all written and managed by their graduate students.) If one uses Powerpoint (or similar) to author slides, the only way he/she can manage his/her slides is to use Powerpoint, which is very time-consuming. Viewing the content of a set of slides would take at least several minutes to start. A busy presenter would not have that kind of time. In summary, current methodologies and software environments can only offer a labor intensive slide authoring and management system to its users. A good system must allow a presenter to manage thousands of slides easily. (We assume each professor has to teach three courses per semester, 48 hours per course and at least 5 slides per hour. He/she might have to teach more than five different courses in several years of his/her teaching career. He/she also has to make several research-oriented presentations every year.)

3. HTML Based Presentaiton Systems

Due to its popularity and sophistication, HTML has become a very popular hypermedia document description language. Together with the *http* protocol and various browsers, it is an excellent candidate for presentation system. In this section, some of their characteristics with respects to the desired properties described in Section 1.2 will be discussed.

3.1 Open Document Format

The document of HTML is in ASCII format. As previously mentioned, an ASCII format allows users to choose whichever editing tool to view and edit document sources. This is one of the reasons why huge number of non-computer-professional users can write millions of HTML documents on the Web. (Although there are WYSIWYG tools available on the market, actually they were available after HTML had become popular already. Thus, we can conclude that WYSIWYG tools did not contribute to the popularity of HTML.) Because of its open format, any one can easily write viewers, browers, WYSIWYG authoring tools and simple script-based middlewares. Widely available tools and the free tool choice are the main factors that contribute to the popularity of HTML.

3.2 Navigation

Due to its hyperlink capability, the navigation of HTML documents are fairly easy. There are at least three aspects to be considered:

1. It is easy to embed navigation hints into HTML documents to actively guide the users browsing the entire hypermedia documents.
2. A user can easily navigate him/herself to any component of a hypermedia document network by explicitly specifying the URL of the target objects.
3. Most HTML browsers offer some frequently needed navigation steps. For example, one can freely navigate him/herself back and forth along the sequence of document objects that he/she has "visited" in the current browsing session.

By combining these properties, it is not too difficult to design a presentation system which is easy to navigate within a network of slides, especially when the pointing devices and the scrolling bar are very difficult to manipulate. It is essential for a slide authoring system to help its users overcome these obstacles.

3.3 Slide Authoring

As mentioned in previous sections, the HTML specification language is easy to learn and can be edited by using any editing tool. Further, it is not too difficult for such an ASCII based language to design some middlewares to facilitate user-friendly and fast authoring environments.

3.4 Multimedia Capability

We have known that HTML is a hypermedia document format with multimedia capability. We've also known that dynamic graphics require intensive programming and the video consumes too much resources. One possible compromise is to use the *dynamic images* specified by GIF89a. A dynamic image consists of a set of images and can be displayed by its viewer one by one automatically. It does not require intensive programming such as dynamic graphics or huge amount of resources such as video. This dynamic image can be generated according to the GIF89a standard by using some public domain software tools easily. HTML also allows users to simulate the same effect. (In some sense, dynamic image is actually a video. However from the viewpoint of resource consumption, it is considered different from video.)

3.5 Slide Management

Since each HTML document is an ASCII file, it can be managed as a plain file using the traditional directory tree convention. There is no need to design a complicated and expensive document database to manage these slides. The ASCII format is a critical factor to make this possible. Usually, it is essential to view a document object quickly in managing a file in a directory tree. For instance, if one has to clean up a directory one month after a presentation, he/she needs to view the content of the files quickly to decide whether to keep that file or not. This is very easy to carry out if the documents are in ASCII format and are stored as regular files.

4. Our Design

4.1 Design Objectives

According to the analysis presented above, we have designed a system which includes file format, slide structure, slide appearance, navigation, and authorware. Other than the requirements listed in Section 1.2, we have design the system with the following further objectives:

1. Minimization of non-content related overhead;
   A common HTML document usually contains lots of special HTML marks. Slides could be the worst case since it usually contains little information and a good variety of formats. (The real information content in most of the slides we designed constitutes less than 10% of the entire source.)
2. Separation of information content and appearance description;
   The source of a slide would be difficult to read if its information content and appearance descriptions are mixed together in the same source, especially when the appearance description constitutes most of the slide source.
3. Effective and efficient information delivery;
   Since the most important objective of a presentation is to communicate information to the audience, all information presented on a slide must be easily (effectively and efficiently) cognizable by the audience. The appearance design of our slide is indeed the greatest challenge to our design.

4.2 Slide Structures and File structures

Each slide is stored as a HTML file. Each set of slides is hierarchically organized and is stored as a directory tree. Terminal nodes are slide files. Nonterminal nodes are the sections. Each slide consists of some components including slide title, background, navigation buttons, and information paragraphs. Each paragraph can have sub-paragraphs, sub-sub-paragraph, recursively. A paragraph (sub-paragraph) may be in various formats such as text, image, table, code segment, music segment, etc. These slides can be managed by using conventional file management skills such as 'cd', 'mkdir', 'rmdir', 'mv', 'rm' 'vi', etc.

4.3 In-Slide Scrolling

It is an ideal to restrict each slide within one single page to avoid the need of scrolling. However, there are some occasions that a slide is longer than a screen page. To scroll a slide up and down by using a wireless "track-ball" mouse single-handed is one of the most challenging

operations in a presentation. Besides, the scrolling bar of most presentation systems are on the bottom or on the far right edge of a slide. It may even be pushed out of the wall-screen when projected onto a wall-screen due to some possible misalignment. (Not every presentation room is well equipped.) To overcome this problem, we have designed some scrolling buttons automatically embedded in the headline of each paragraph. A presenter can point to and click on these buttons to perform some scrolling operations such as moving to the next, previous, or the first paragraph.

4.4 Slide Navigation

The goal of navigation design is to minimize the number of steps moving from one slide to another. Obtaining many ideas from our iterative experiments, we have designed a hierarchical hyperlink network within the constraints of a slide area to facilitate convenient slide navigation within the same set of slides as well as across different sets of slides. The goal of this design is to minimize the number of steps, under the slide-area constraint, to reach any slide in the system.

4.4.1 Between the same set of slides

The set of slides under the same directory is called a *section*. All slides in the same section are fully connected so that one can jump from one slide to another by clicking a navigation button in the slide. There are two sets of navigation buttons, one is labeled by their file names and is on the left frame which occupies only a small slide area. The other set of navigation buttons is on the top of each slide, which is labeled by the titles of the slides in the same section. Each slide also has a navigation button that points to the root of the current slide set. From there, one can change to any section in the same slide set. One can reach any slide in the same slide set in three clicks.

4.4.2 Between different sets of slides

The roots of all slide sets are fully connected. Thus, it is easy to jump from any slide to its root (there is a hyperlink in each slide pointing to its root), and to the root of another slide set. Thus, one can reach any slide in the same slide set in four clicks. Since the slide area for navigation buttons is limited, only several sets of slides can be managed at the same time. However, if there are more slide sets to be managed, it is not difficult to extend the model to create a root page for all slide sets.

4.5 Artistic Design to Present Semantic Structure

Within each slide, the information components are assumed all correlated with various semantic distance. These components actually form a complicated semantic structure. A good presentation format must be able to deliver these semantic structures to the audience clearly and effectively. Indentation, font size, and color are typical components for the construction of a format structure to carry structured semantics. Expecting a complicated semantic structure, it is the greatest challenge to design such a format framework under the mentioned constraints.

4.6 Multimedia and Dynamic Image Capability

Since the most popular HTML browsers such as Netscape Navigator can display dynamic images (e.g. in GIF89a format), our design automatically includes this capability. This is an excellent tool in presenting algorithmic processes.

4.7 SimpleSlide Format Middleware and Open Slide Architecture

To allow users using their favorite tools and working environment to create and manage their slides, we propose to use plain text files and regular directory structure as the slide architecture. A user can use the SimpleSlide format to describe the slides he/she wants to create as follows:

```
0 Slide Title
1 1 First Level Paragraph
2 1.1 Second Level Paragraph
2 1.2 Second Level Paragraph
3 1.2.1 Third Level Paragraph
1 2 First Level Paragraph
2 2.1 Second Level Paragraph
3 2.1.1 Third Level Paragraph
3 2.1.2 Third Level Paragraph
```

The single digit at the beginning of each line is the "tag" of SimpleSlide. It denotes the level of the line in the semantic hierarchy. (We assume that the paragraphs in a slide form a tree-like hierarchy.) A middleware will convert a slide in such a format into the HTML format with all navigation buttons and all other artistic designs automatically embedded. The middleware consists of these independent command tools:

1. a tool to convert macros,
2. a tool to process single slide, and
3. a tool to convert the whole set of slides.

These tools are very easy to be constructed by a programmer who has basic shell script programming skills.

4.7.1 Macro processing

To memorize all HTML tags is quite tedious especially for those busy presenters. (Even a good HTML programmer would need several hours to edit a single page of HTML document.) Some good macro definitions to replace tedious HTML tags can greatly enhance one's productivity. Further, the SimpleSlide format presented here is only an example to demonstrate how easy it can be implemented. It is by no means a final proposal. Anyone can design a better one to replace it.

4.7.2 Single slide processing

The SimpleSlide format is very simple, and can be easily implemented by perl or awk scripts.

4.7.3 Slide set processing

The third tool is to convert a set of slides which is stored as a directory structure into a hierarchy of HTML slides. These slides are automatically linked together by HTML hyperlinks. All navigation buttons are automatically generated.

5. Experiments

5.1 Authoring Efficiency and Slide Management

Within several months, we have used the proposed architecture and the developed tools to authore thousands of slides in three courses. Author Lien has also used this system to make several formal presentations in some technical conferences. Without the help of any assistant, he can easily produce informative slides for a conference presentation within limited time frames and efforts (say, one day and single-handed). Further, as we have expected, managing these slides is as easy as regular file management.

5.2 Classroom/Conference Presentation

The presentation system has been used in several conferences and all three classes conducted by author Lien since the Spring semester of 1997. He has been using a projection notebook in these presentations. It requires approximately 5 minutes to set up the entire system at the beginning of a class. After that, the class can be conducted quite smoothly. The navigation of slides within the same set or across different sets are quite easy as we have expected.

5.3 Teaching Effectiveness

The bottom line of the entire system is to enhance the teaching effectiveness or the efficiency of communications between presenters and audience. We have found that, with the dynamic image feature this low cost presentation system can easily outperform other presentation systems. The feedback from the audience is pretty encouraging. We had proceeded with an informal survey to study the teaching effectiveness of this presentation system. Out of 48 responses, 34 students agree and only four students disagree that this presentation system can improve their learning efficiency. Compared to the traditional transparency presentation system, 31 students prefer our system, while 17 students don't.

Major reasons that affect the teaching effectiveness of this presentation system are as follows:

1. The classroom is too dark, thus it requires extra efforts to stimulate those students who have little interests on the presented subject.
   The reason is that, in most of our presentations, we used a projection notebook over a regular overhead projector which is designed for projecting transparencies. The contrast of the images on the wall-screen is too weak thus the light in the classroom must be turned off. This problem can be solved by using a regular projector.
2. Viewing slides over Internet may be too time-consuming as compared to hard-copy handouts.
   We used to distribute hard copy handouts to the students. We thought that offering slides on Internet can eliminate the need of hard-copy distribution. However, this experience shows us that this need cannot be completely eliminated until the Internet can be ubiquitously and cost-effectively accessed.

In summary, these problems are not inherent problems of this presentation system and thus can be corrected without changing the system itself.

6. Concluding Remarks

Traditional presentation systems requires considerable resources and preparation time to support the presentation facility and material preparation. This paper proposes a more dynamic, more flexible presentation system and an open authorware architecture based on HTML format to facilitate fast slide authoring. We have developed a workable middleware system. By using this system, we have produced thousands of slides which are used in several classes with minimum efforts. We believe that this idea can benefit many future presenters who has no luxury to use a labor intensive slide authoring and management system.

Reference

1. Chien Chou, "Designing Hypertext-Based Learning Courseware for Computer Networks", *1997 International Conference on Computer System Technology for Industrial Applications*, April 1997, pp. 140-146.
2. Shyan-Ming Yuan, "A study of Applying Hypertext System to Intelligent Computer Assisted Learning in a Distinct Cooperative Learning Environment", *1997 International Conference on Computer System Technology for Industrial Applications*, April 1997, pp. 158-162.