CHARLES STURT UNIVERSITY
CRO CSU Research Output
Showcasing CSU Research

http://researchoutput.csu.edu.au

This is the Author's version of the paper published as:

Title: An Efficient Approach for Building Web-based Courseware
Author: X. Huang and W. Lai
Author Address: xhuang@csu.edu.au
Conference Title: Year of Conference: 2007
Conference Location: KunMing, China
Editor: B. H. Hong Liu, Xiangwei Zheng, HuiZhang
Publisher: IEEE
Pages: 260-263
URL: http://www.isitae.org/english.htm
Keywords: Web courseware, on-line learning, user interface, diagram navigation
Abstract: The courseware development has moved to integrate multimedia into the Internet techniques. We think that the courseware design should not only concern about how to publish teaching materials on the Web and to integrate audio/video facilities, but also focus on other important issues, such as user interfaces and software architecture. In this paper, we present a general architecture for building a courseware Web site, identifying the different roles of system programmers, Web developers, and course designers. We emphasize the user interfaces is the crucial part for a courseware system.
CSU ID: CSU279948

An Efficient Approach for Building Web-based Courseware

Wei Lai

Faculty of Information and Communication Technologies

Swinburne University of Technology

PO Box 218 Hawthorn, Vic 3122, Australia

Email: wlai@swin.edu.au

Xiaodi Huang

School of Information Technology

Charles Sturt University

Albury, NSW 2640, Australia

Email: xhuang@csu.edu.au

Abstract – The courseware development has moved to integrate multimedia into the Internet techniques. We think that the courseware design should not only concern about how to publish teaching materials on the Web and to integrate audio/video facilities, but also focus on other important issues, such as user interfaces and software architecture. In this paper, we present a general architecture for building a courseware Web site, identifying the different roles of system programmers, Web developers, and course designers. We emphasize the user interfaces is the crucial part for a courseware system.

Keywords: Web courseware, on-line learning, user interface, diagram navigation.

1. Introduction

Courseware is education software that supports users' learning by computers. With Web techniques, courseware has been built on the Web to support distance education and on-line learning. Web courseware has integrated multimedia techniques to provide the user more impressive presentations combined with text, graphics, animation, audio and video. Several such courseware systems are available [1, 2, 3, 4]. Although they publish teaching materials on-line, there is little focus on effective user interfaces for students' learning.

We think that the courseware design should not only publish teaching materials on the Web and integrate audio/video facilities, but also should focus on other issues, such as

user interfaces and software architecture. That is, what kinds of user interfaces are more effective for students' learning and what kind of architecture is more effective for Web courseware development.

In this paper we present the system architecture for web courseware development and introduce the user interface design for Web courseware.

2. System Architecture

Using the Model-View-Controller design, the system architecture consists of *presentation layer*, *pedagogical control layer*, and *domain knowledge base*. The *presentation layer* specifies in which form to display the teaching materials such as tables, diagrams and multimedia. The *pedagogical control layer* is responsible for how to present the teaching materials in a way that is suitable for the level of a particular student and for specific requirements of knowledge, such as the appropriate pace of presentation, self-study or online group discussion, and example-generalization-excise. The *domain knowledge base* contains the teaching materials to be studies.

Except for storing teaching materials, the database of *domain knowledge* on the server supports

- *on-line operations* (such as storing, updating, and managing course contents),
- *query* (such as extracting required tasks and concepts), and
- *Security* (such as authentication).

From the development team perspective, the system includes *courseware design team multimedia web design team* and *system and tools design team*, as illustrated in Figure 1. Lecturers and professors are responsible for Web courseware development. They are responsible for designing the courses and then present their design and requirements to the multimedia and Web design team, who conducts the web design according to the lectures/professors' course design and requirements. The system and tool design team is responsible for developing a database system for courseware on the web server and software tools for other teams and users.

The software tools provided include two kinds of interfaces:

1. *Designer Interfaces* for the multimedia and Web design team to build Web courseware. These interfaces provide functions such as creating/deleting a course package and its elements.
2. *User Interfaces* for the user to select their course package and set up their learning agenda, and for displaying interactive on-line learning materials to the user.

This architecture is similar to the general approach for developing Web site. The features of our architecture are (1) the separation of tasks for course designers, multimedia and Web designers and system and tools developers, (2) effective interfaces for Web courseware development and for students' learning.

The Web courseware development includes several design issues. In the following we introduce the effective interfaces for students' learning.

3. User Interface

A courseware contains a sequence of concepts. To help students to understand the concepts, the online quiz should be designed for testing the understanding of the concepts. Normally quizzes require students to complete some tasks. For example, a task requires students to create an image on the Web. It enables students to review relevant concepts such as image format, image compassion, and color model.

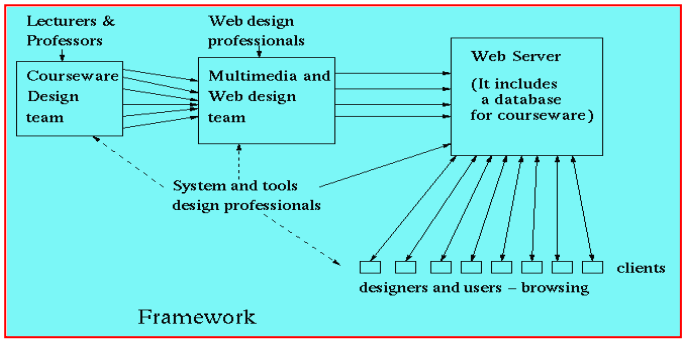

Figure 1: Architecture for Web courseware development

The user interface should allow students to select those concepts/tasks in which they want to learn or review. In the process of students' learning a course, students may not be interested in some concepts which they may have already known or they have learned from another course. Students may like to explore those concepts which are unfamiliar to them and/or which are closely related to the tasks in quizzes or practical exercises. Therefore, we should develop effective user interfaces to support students' choices. Three kinds of interfaces for the students' selection are designed.

1) *Catalogue Interface*. It lists all concepts contained in each course. The user can select them from the catalogue.

2) *Keyword Search Interface*. It supports the user to select the concepts and tasks by specifying keywords. After the user enters keywords, a list of concepts and tasks is shown in decreasing order of relevance to the keywords. The user then chooses some of them.

3) *Diagram Interface*. It uses diagrams to show relationships between courses / concepts / tasks. The diagrams show which one should be learned first and which one should be learned later.

The first two kinds of interfaces are similar to those used in the search engines [5, 6]. We adapted them for Web courseware. After the user's selection, a learning list is generated on the basis of inherent relations

between concepts or tasks. The third kind of the interface is a new approach for Web searching. Guiding students' selection, the diagram interface illustrates the dependency relationships between concepts and/or tasks of a course. Instead of providing a learning list after the user's selection, a diagram learning agenda is shown for helping the user to set up his/her learning plan.

4. Interface Examples

The best way of showing our diagram interface is to use some examples. Figure 2 shows an on-line diagram interface which is converted by parsing and filtering the Web page: www.csu.edu.au. We provide three kinds of modes for the user's interaction: *Editing, Navigation, Show*, which can be switched between by clicking the left button, middle button, right button on the mouse respectively. In the *Editing* mode, the user is able to edit and adjust a diagram layout. For example, the user changes a diagram from a horizontal tree (i.e. *h-tree*) in Figure 2 to a tip-over tree shown in Figure 3.

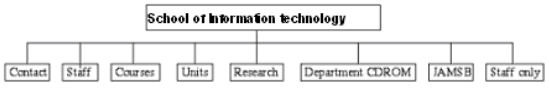

Figure 2: A diagram interface

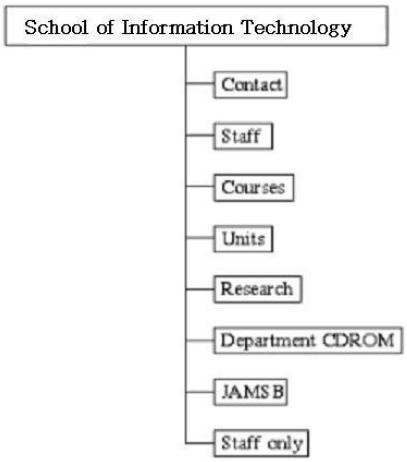

Figure 3: Another alternative layout

In the *Navigation* mode, the user can navigate the Web space by selecting the nodes in the diagram. For example, after the user clicks the node '*Courses*' and the node '*Bachelor Courses*' in Figure 3, another diagram (see the window on the left in Figure 4) is shown based on the user's current focused node '*Bachelor Courses*'.

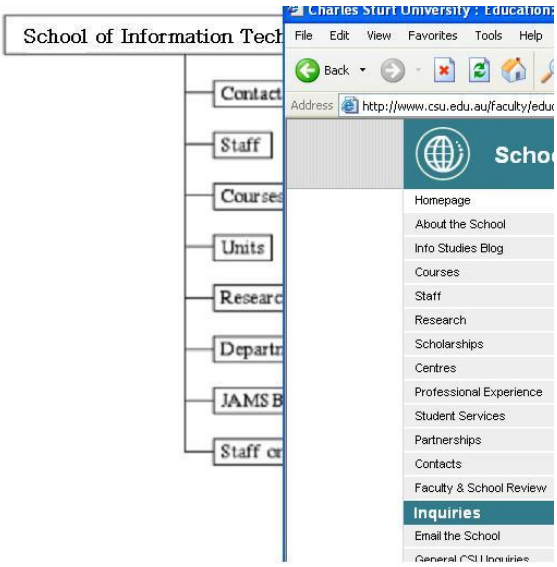

Figure 4: Navigation nodes and their corresponding web page

Our system can support the display of detailed information corresponding to a node in a diagram (after the *Show* mode is set up). For example, after the user selects the node '*DD75-Bachelor of Information Technology*' in the window on the left in Figure 4 in the Show mode, the window on the right in Figure 4 is shown to display the detailed view for this node. The *Show* mode used for the nodes of concepts and tasks is to provide examples for concepts and demonstrations for tasks respectively.

In the process of students' learning for a course, students would concentrate on learning those selected concepts and tasks. For each concept node, an example is provided in the *Show* mode. This is because good examples reinforce concepts to make them understandable to students. For each task node, a demonstration is provided in the *Show* mode. A demonstration enables students to know the irement for the task. In this way, we can stimulate students' enthusiasm for learning.

The window on the left in Figure 5 shows a diagram interface for tasks. It shows the relationships between tasks. Task 1, Task 2 and Task 3 are prerequisite tasks for Task 4. Task 5 is an exploratory task for Task 4. This diagram interface helps students to set up their learning plan. For each task, a demonstration is provided in the *Show* mode. Figure 5 shows a demonstration for Task 4 after the user clicks the node of Task 4 in the *Show* mode. Figure 6 shows a demonstration for Task 5 after the user clicks the node of Task 5 in the *Show* mode.

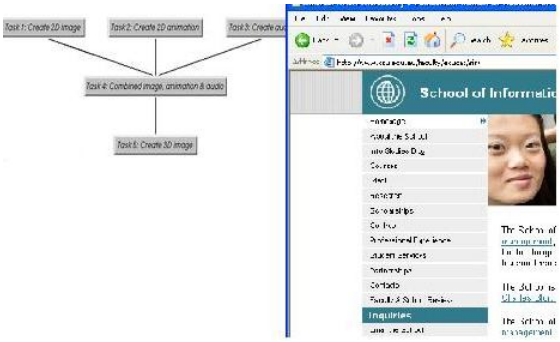

Figure 5: A demonstration for Task 3

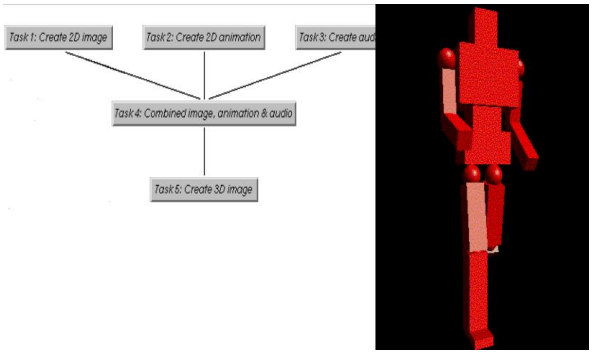

Figure 6: A demonstration for Task 5

5. Navigation and Diagram Layout

The courseware system allows users to explore the knowledge to be studied. One of technique issues is how to display the diagrams of concepts or task during users' exploration. It is required that a number of dynamic diagrams should be displayed in a way that the layouts of two consecutive diagrams have no a big change. The *diagram navigation* techniques used in our system are based on those in [7, 8, 9].Usually it is impossible to visualize a huge diagram in practice. A solution is to use a sequence of subsets of the diagram for Web courseware navigation. A subset is formed based on the user's interaction. The Web subset should be changed dynamically and should follow the user's focus in navigation. This change from one Web subset to another should preserve the user's mental map [11, 12] for diagram navigation. When the user changes the focus (i.e. click a node in the Navigation mode), a new sub-diagram is formed by dropping old nodes and adding new nodes. An example is the diagram navigation from Figure 3 to Figure 4. This is similar to driving a car: new views arrive in the front and old views vanish in the back.

Layout and layout adjustment for diagram editing has been investigated [11, 12, 13]. A change in the combinatorial graph (made either by the user or the application) should induce a change in the layout on the screen. However, the layout should not change so much as to disturb the user's "mental map" of the diagram. Some mathematical models of "mental map" have been defined and some layout adjustment techniques [11, 12, 13] developed for rearranging a diagram while preserving its "mental map". These techniques have been extended for diagram layout and layout adjustment in our Web courseware system.

6. Related Work

Our work is motivated by the growing on-line learning systems, such as [1, 2, 3, 4]. However, we found the interfaces in these systems are not effective for students' learning. Web courseware interfaces should give students more flexibility to select the topics they want to learn and more convenience to establish their learning plans.

This paper presents a new approach of using diagrams as interfaces for Web courseware. Our diagram layout techniques are related to the research on graph drawing algorithms (see [10]). However, these algorithms were designed for drawing the

graph where nodes take a little or no space. The diagrams in practice have node images varying in shapes and size. These nodes are circles, boxes, diamonds and similar shapes, and may contain a considerable amount of text and graphics; thus they take a nontrivial amount of screen space. To handle these diagrams in practice, some techniques have been developed for adjusting a diagram layout and ensuring that there are no overlapping node images in a diagram [11, 12, 13]. These techniques are adopted for the diagram interface of our Web courseware system.

Our diagram-based navigation approach for Web courseware can give the user a whole structure and detail views for concepts and tasks of a course. Traditional web navigation tools, such as Netscape and Microsoft Internet Explorer, cannot provide a contextual overview required for global orientation; instead they only give bookmarks and history lists which are at most a linear list. Some researchers have proposed "Web graph" approach [8, 9]. That is to use a graph for the World Wide Web navigation. Nodes in a graph represent URLs, and edges between nodes represent links between URLs. The whole cyberspace of the WWW is considered as one graph - a huge and dynamic growing graph. The Web graph gives the user a 2-dimensional view for URLs and their interconnections. However, it is impossible to visualise this huge Web graph in practice. The solution is to use a sequence of subsets of the Web graph for WWW navigation. A Web subset is formed based on the user's current focus. The Web subset should be changed smoothly following the user's focus change [8, 9]. We adapt this kind of Web graph navigation techniques for our web courseware system.

The general methods for Web site development and catalogue / keyword searching have been used in many web sites. Examples are [1, 2, 3, 4, 5, 6]. These methods are adapted and fairly well-tuned for our web courseware development.

7. Conclusion

This paper has introduced the system architecture for Web courseware development. We focus on introducing the design for effective user interfaces, especially diagram interfaces.

The feedbacks from users are that they like to use the combination of three kinds of interfaces: catalogue interface, keyword search interface and diagram interface for selecting learning topics and establishing learning agendas. Especially they find the diagram interface is very helpful for navigation and browsing in the learning process.

Further work is needed to improve the diagram interface. In particular, we will continue to investigate diagram layout and diagram navigation techniques that will enhance potential usability of the system. Also, we need to test our interfaces with end-users for further improvement.

References

[1] DCU WWW Database courseware. http://www.compapp.dcu.ie/databases/welcome.html

[2] The CourseWare Company. http://www.courseware.co.uk/

[3] Global University Alliance. http://www.globaluniversityalliance.com/

[4] Next Generation Education. http://www.nexted.com/

[5] Yahoo! http://www.yahoo.com/

[6] Excite. http://www.excite.com/

[7] Lai, W. and M. Danaher. 1996. "Diagram navigation for information browsing". *Proceedings of the First Asia Pacific Conference on Computer Human Interaction*, pp. 339-345, Singapore, June 1996.

[8] Lai, W., M. Huang, Y. Zhang and M. Toleman, M. 1999. "Web Graph Displays by Defining Visible and Invisible Subsets", *Proceedings of the Fifth Australian World Wide Web Conference*, pp. 207-218, Ballina, NSW, Australia, April 1999.

[9] Huang, M., W. Lai and Y. Zhang, Y. 1999. "Mapping and Browsing the Web in a 2D Space", *Proceedings of the 10th International Workshop on Database and Expert Systems Applications*, pp. 248-252, IEEE Computer Society Press, Florence, Italy, September 1999.

[10] Battista, G.. D., P. Eades, R. Tamassia, and I.Tollis. 1998. "Graph drawing: algorithms for the visualization of graphs", Prentice Hall.

[11] Eades, P., W. Lai, K. Misue and K. Sugiyama. "Preserving the Mental Map of a Diagram", *Proceedings of COMPUGRAPHICS 91*, pp. 34-43.

[12] Eades, P., W. Lai, K. Misue and K. Sugiyama. 1995. "Layout Adjustment and the Mental Map", Journal of Visual Languages and Computing, No. 6, pp. 183-210.

[13] Eades, P. and W. Lai. "Algorithms for Disjoint Node Images", *Australian Computer Science Communications*, 14(1):253-265, 1992.