

# Designing Effective User Interfaces for Web Courseware

**Wei Lai**

Department of Mathematics and Computing  
University of Southern Queensland  
Toowoomba, QLD 4350, Australia

**Xiaohua Jia**

Department of Computer Science  
City University of Hong Kong,  
Kowloon, Hong Kong

**Dongguang Li**

School of Computer and Information Science  
Edith Cowan University  
Perth, WA, Australia

**Jiro Tanaka**

Institute of Information Sciences and  
Electronics  
University of Tsukuba  
Tsukuba, Ibaraki 305-8573, Japan

## Abstract

Current courseware development has moved to integrate multimedia, the WWW and Internet techniques. We think that the courseware design should not only put teaching materials on the Web and integrate audio/video facilities, but also should focus on other issues, such as user interfaces and software architecture. This paper presents an architecture for Web courseware development and introduces the design for effective user interfaces which can help students' learning.

*Keywords:* Web courseware, user interface, diagram navigation, diagram layout.

## 1. Introduction

Courseware is education software that supports the user's learning on computer. With recent Internet and Web techniques, courseware has been built on the Web to support distance education and on-line learning. Current Web courseware has integrated multimedia techniques to provide the user more impressive presentations combined with text, graphics, animation, audio and video.

However, we think that the courseware design should not only put 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 an architecture for web courseware development and introduce our user interface design for Web courseware.

## 2. Architecture

The architecture for Web courseware development is illustrated in Figure 1. It includes three parts: *courseware design team*, *multimedia web design team*, and *system and tools design team*. Lecturers and professors are in the courseware design team. They are responsible for designing their courses and then pass their design to the multimedia and web design team. The multimedia and Web design team, using the end-user multimedia tools (e.g. FrontPage, PhotoShop, and Director) and tools provided by the system and tools design team, is responsible for converting lecturers/professors' course design to Web based presentations. 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 database for Web courseware on the server supports

- *on-line operations* (such as storing,

- updating, and managing course contents),
- *information retrieval* (such as extracting required tasks and concepts), and
- *information security* (such as checking valid accesses).

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.

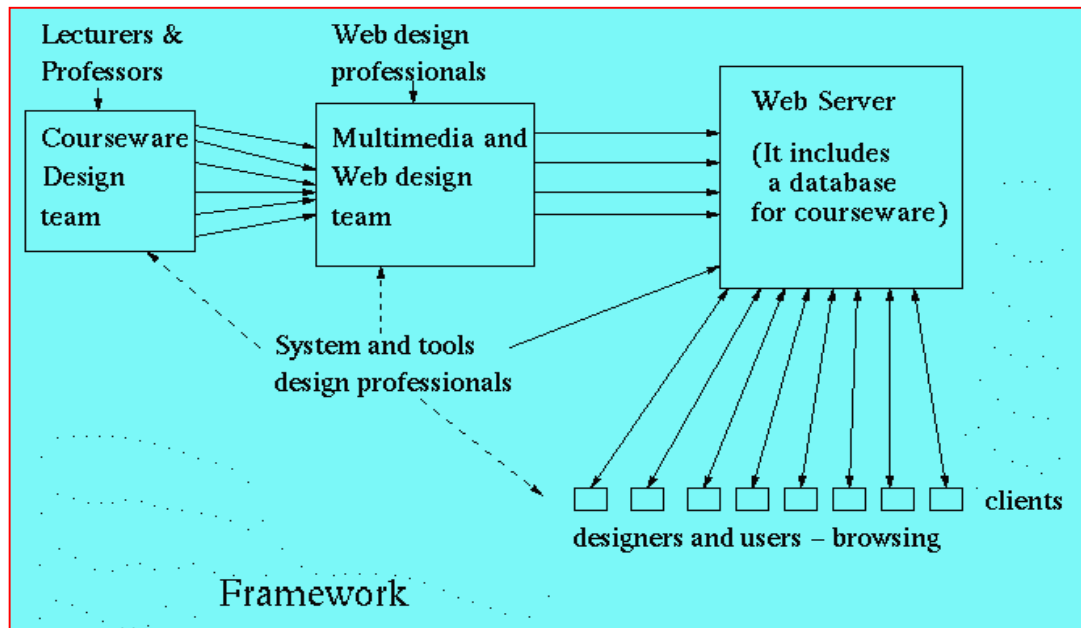


Figure 1: Architecture for Web courseware development

This architecture is similar with those for other Web sites development. 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 focus on introducing the effective interfaces for students' learning.

### 3. User Interface Design

A course package is a sequence of

concepts. To help students to understand the concepts, assignments should be designed for the concept review. Normally assignments require students to complete some tasks. For example, a task requires students to create an image on the Web, it enables students to review concepts such as image format, image composition, and color model.

The user interface should support students to select those concepts/tasks that they want to learn or review. In the process of students' learning for a course, students may not be interested in some concepts which they may have already know or they may have learned from another course.

Students may like to learn those concepts which they do not know and/or which are close related to the tasks in assignments or practical exercises. Therefore, we should develop effective user interfaces for students' learning. We designed three kinds of interfaces for the students' selection.

- (1) *Catalogue Interface*. It lists each course that includes its concepts and tasks. The user can select them from the catalogue.
- (2) *Keyword Search Interface*. It supports the user to select the concepts and tasks by typing keyword(s). After the user enters keyword(s), a list of relevant concepts and tasks is shown up. The user can choose those that he/she wants in the list.
- (3) *Diagram Interface*. It uses diagrams to show relationships between courses / concepts / tasks. That is which one should be learned first and which one should be learned later.

The first two kinds of interfaces are similar with those used in current search engines [1, 2]. We adapted them for Web courseware. After the user's selection, a learning list is established. The third interface is a new approach for Web searching. The diagram interface can show

relationships between concepts and/or tasks of a course for guiding students' selection. Instead of providing a learning list after the user's selection, a diagram learning agenda is shown up for helping the user to set up his/her learning plan.

#### 4. Examples of Diagram Displays

The best way to show 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.sci.usq.edu.au](http://www.sci.usq.edu.au). We provide three kinds of modes for the user's interaction: *Editing*, *Navigation*, *Show*, which can be set up by clicking the left button, middle button, right button on the mouse respectively. In the Editing mode, the user can edit and adjust a diagram layout. For example, the user can change a diagram from a horizontal tree (i.e. *h-tree*) in Figure 2 to a tip-over tree shown in Figure 3.

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 up based on the user's current focused node 'Bachelor Courses'.

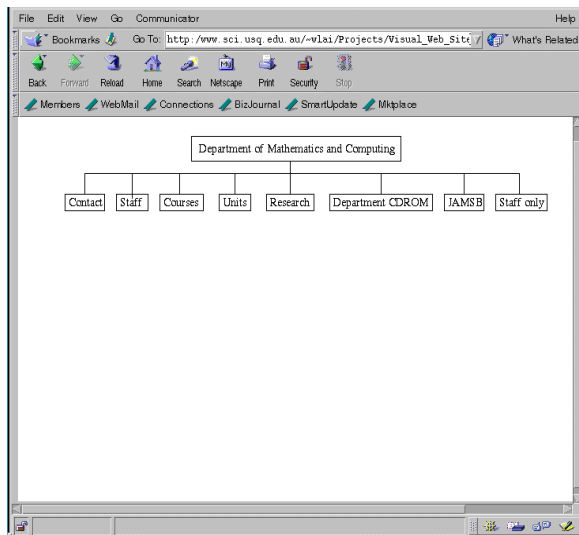


Figure 2: A diagram interface

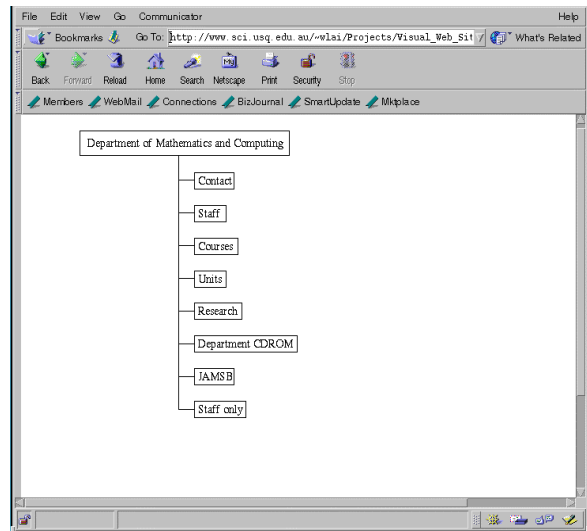


Figure 3: Another alternative layout

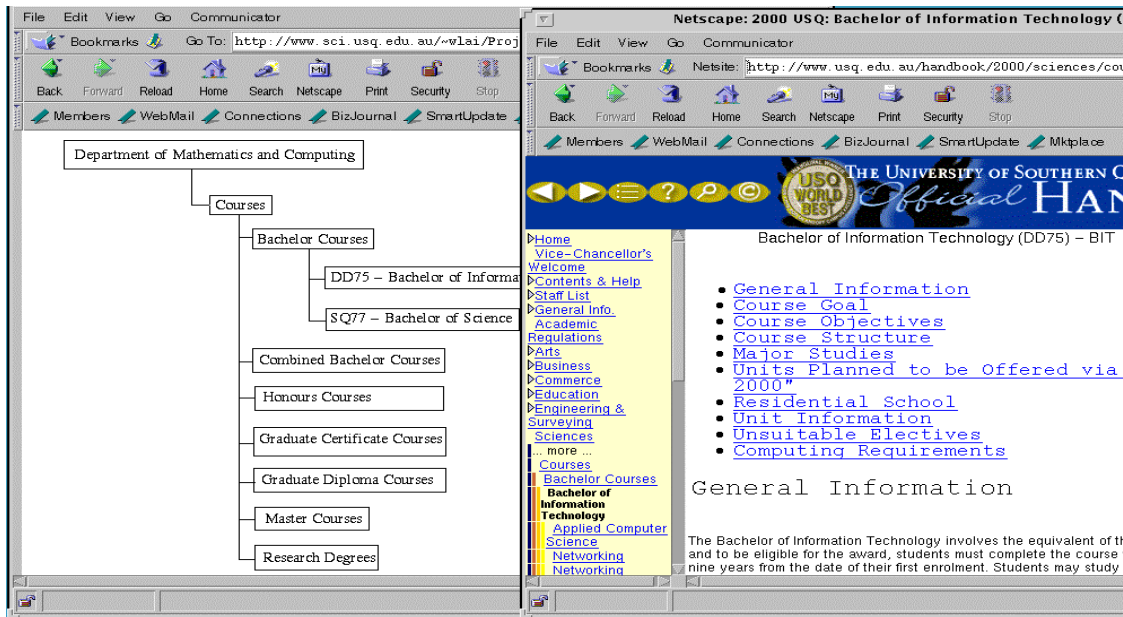


Figure 4: Navigation to another sub-diagram

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 up 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. Because good examples can reinforce concepts to make them understandable to students. For each

task node, a demonstration is provided in the Show mode. A demonstration enable students really know the requirement 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 exploratory task for Task 4. This diagram interface can help 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 click the node of Task 4 in the Show mode. Figure 6 shows a demonstration for Task 5 after the user click the node of Task 5 in the Show mode.

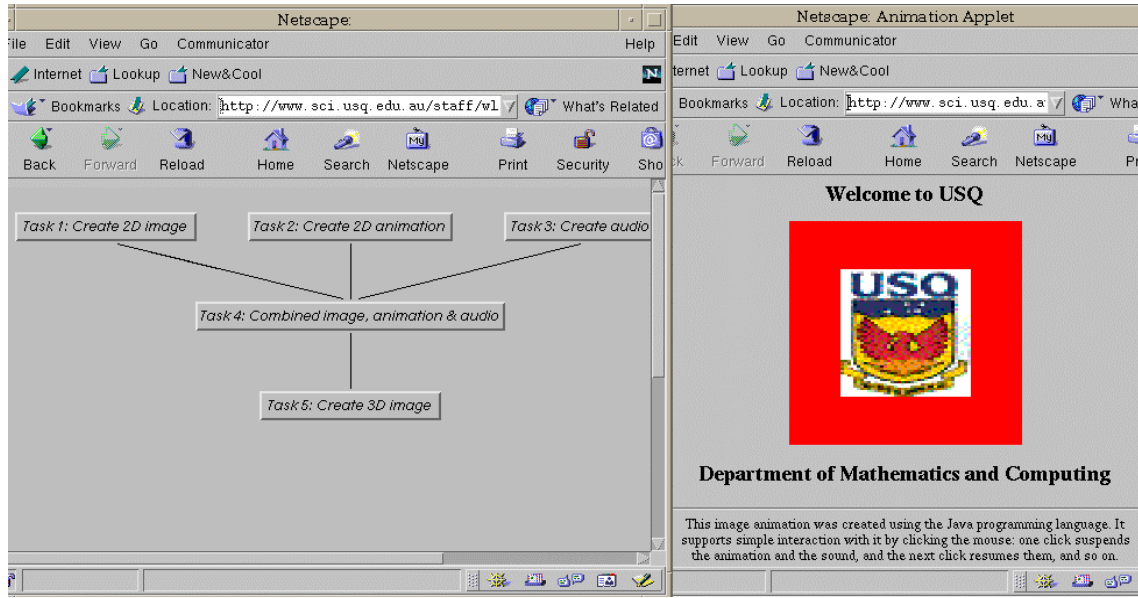


Figure 5: A demonstration for Task 3

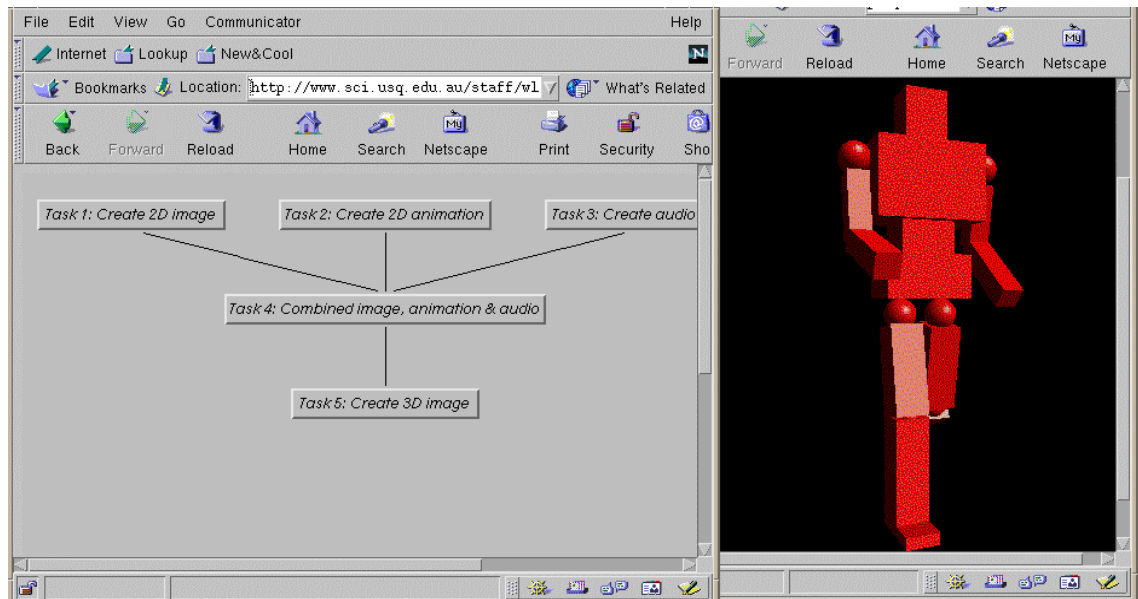


Figure 6: A demonstration for Task 5

## 5. Diagram Navigation and Layout

The *diagram navigation* techniques used in our system are based on those in [3, 4, 5]. Because 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 [7, 8] 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 [6, 7, 8, 9]. 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 [7, 8, 9] 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.

## 5. Conclusion

This paper introduces an architecture for Web courseware development. The architecture covers several design issues. We focus on introducing the design for effective user interfaces, especially diagram interfaces. 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.

## References

1. Yahoo! <http://www.yahoo.com/>
2. Excite. <http://www.excite.com/>
3. Wei Lai and Maurice Danaher. "Diagram navigation for information browsing". *Proceedings of the First Asia Pacific Conference on Computer Human Interaction*, pp. 339-345, Singapore, June 1996.
4. W. Lai, M. Huang, Y. Zhang and M. Toleman, M. "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.
5. M. Huang, W. Lai and Y. Zhang, Y. "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.
6. G. Di Battista, P. Eades, R. Tamassia, and I.Tollis. "Algorithms for drawing graphs: an annotated bibliography", *Computational Geometry: Theory and Applications*, 4(5), 1994.
7. P. Eades, W. Lai, K. Misue and K. Sugiyama. "Preserving the Mental Map of a Diagram", *Proceedings of COMPUGRAPHICS 91*, pp. 34-43, 1991.
8. P. Eades, W. Lai, K. Misue and K. Sugiyama. "Layout Adjustment and the Mental Map", *Journal of Visual Languages and Computing*, No. 6, pp. 183-210, 1995.
9. P. Eades and Wei Lai. "Algorithms for Disjoint Node Images", *Australian Computer Science Communications*, 14(1):253-265, 1992.