A Web Design Modeling System to Improve Communication between Web Designers and Programmers

Lee, Ji-Hyun and Kuo-Hui Lu*

*National Yunlin University of Science & Technology, Graduate School of Computational Design
jihyun@yuntech.edu.tw; g9234706@yuntech.edu.tw

Abstract: Web development has multidisciplinary features involving diverse team members. Owing to the diverse backgrounds, however, there may be communication problems among web team members, especially between designers and programmers. Programmers often use a graphical modeling notation to communicate their designs. The graphical modeling concepts can also be helpful to designers, but most designers generally find graphical modeling notations hard to understand. We, therefore, propose an interactive computer-aided collaborative web design system including revised web graphical model notations to improve communication between web designers and programmers. We have designed and implemented a computational prototype to validate the approach and illustrate how the prototype works using concrete examples.

Key words: Improving Communication, Web Modeling, Collaborative System

1. Introduction

Web site development has multidisciplinary features. The development project usually involves diverse team members, such as a project manager/producer, system analysts, programmers, designers, information architects, etc., because the scale of web applications has increased and the sites have to handle different kinds of information.

However, owing to their diverse backgrounds, there may be communication problems among web team members, especially between the designers and programmers. This is because web designers and programmers principally have different expectations about what constitutes a successful collaboration. Web designers create the look and feel of a site. To apply these principles properly, they use visualization methods such as sketches, drawings, renderings and animations to illustrate their projects[1]. On the other hand, programmers are more concerned about data storage and functional deployment, and they often use a

Corresponding author: Tel: +886 (5) 5342601 #6511; Fax: +886 (5) 5312169.

[1] Newman, M. and J. Landay (2000) 'Sitemaps, storyboards, and specifications: A sketch of web site design practice', in *Proceedings of an International Conference on Designing Interactive Systems*, August 17-19, New York, USA, pp. 263-274.

● Received : Oct. 23. 2006 ; Accepted : Feb. 23. 2007

graphical modeling notation to communicate their designs[2]. We believe that graphical modeling concepts can also be helpful to designers, because a good model can effectively indicate a clear web structure that often can help web designers in analyzing their tasks correctly, to refine each page of a site easily, and to improve accessibility. However, most designers generally find graphical modeling notations hard to understand so that they hardly use them. To meet the challenges facing web modeling, an interactive computer-aided collaborative web design system including revised web modeling notations is conceived to assist a web site development team to attain their work successfully.

In this paper, therefore, we propose an interactive computer-aided collaborative environment to improve communication between web designers and programmers. To do this, we first explore the web design process, identify the design activities of both web designers and programmers, and analyze the communication problems between them. Based on these activities and analyses, we determine the functions to add to as well as to remove from the web graphical modeling notations related to web design to satisfy the requirements. Finally, we design and implement an interactive computational prototype to validate this approach and illustrate how the prototype works using concrete examples.

2. Background

2.1 Web Site and Application Development Process

A well-defined process is useful for web project development because that not only can help web team focus their time and energy on clear goal, but lets customer know what they can expect from web team and what web team need from them to build a web site that meets their expectations[3]. According to Douglas et al. (2003), the development process of a web site can be divided into seven phases: discovery, exploration, refinement, production, implementation, launch and maintenance (Fig.1). The early four phases can be generally called web design process in the whole web site development process.

There are many software development methodologies and modeling languages proposed to help programmers understand how information systems work. For the web development domain these include the Object-Oriented Hypermedia Design Model (OOHDM), Web Modeling Language (WebML), and OPEN space Framework and Relationship Management Methodology (RMM). Most of these tools have specific modeling notations for the navigation, semantics or architecture of system. The web modeling notations of these methodologies use simple shapes, lines or arrows to constitute clear diagrams, which are intuitive and useful for presenting web site functions. Fig.2 shows an example of web application development process using WebML.

---

2 Burdman, J. (1999) *Collaborative Web Development – Strategies and Best Practices for Web Teams.* Upper Saddle River: Addison-Wesley.

3 Douglas, K., J. Landay and J. Hong (2003) 'Processes for developing customer-centered sites', in *The Design of Sites: Patterns, Principles, and Processes for Crafting a Customer-Centered Web Experience.* Boston: Addison-Wesley, pp. 87-105.

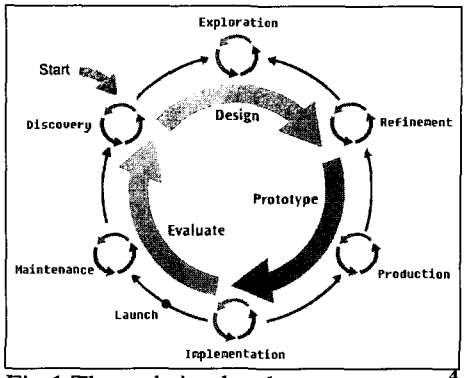


Fig.1 The website development process[4]

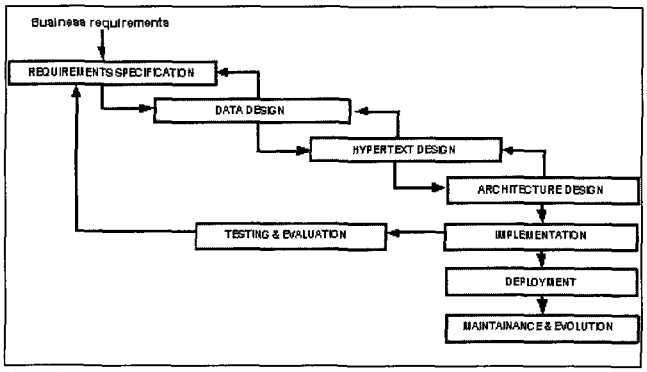


Fig.2 Data-intensive web application development process[5]

There are eight phases in the process: *requirements specification*, *data design*, *hypertext design*, *architecture design*, *implementation*, *testing & evaluation*, *deployment*, and *maintenance & evolution*. The whole process is divided into two parts: from phase 1 to 3 are the ***design*** phases and the other phases are the ***implementation*** phases of the web application development process[6].

2.2 Web Team

According to McDonald and Welland (2001), software engineers and creative designers are the most important staff in a web team. Therefore, communication problems between web designers and programmers are a critical issue.

Web designers generally have to possess several skills to construct site UIs and they are usually concerned about not only the look and feel of the site, but also the navigation behavior of users. Through the study of web site design practice, several researchers observed that web designers use different levels of

---

[4] Douglas, K., J. Landay and J. Hong (2003) 'Processes for developing customer-centered sites', in *The Design of Sites: Patterns, Principles, and Processes for Crafting a Customer-Centered Web Experience*. Boston: Addison-Wesley, pp. 87-105.

[5] Ceri, S., P. Fraternali, A. Bongio, M. Bambilla, S. Comai and M. Matera (2003) *Designing Data-Intensive Web Applications*. San Francisco: Morgan Kaufmann.

[6] Ceri, S., P. Fraternali, A. Bongio, M. Bambilla, S. Comai and M. Matera (2003) *Designing Data-Intensive Web Applications*. San Francisco: Morgan Kaufmann.

refinement: site map, storyboard, and individual page/schematics[7] [8]. Designers also need to check and decide the layout and content that appear on individual pages. Lin et al. (2000) developed a tool called DENIM, as shown in Fig.3. Designers can use this tool to sketch web site diagrams using the three levels of refinement mentioned above. However, DENIM just presents low-fidelity diagrams that only are available for the communication with web designers because it cannot be compatible with the modeling notations used by programmers.

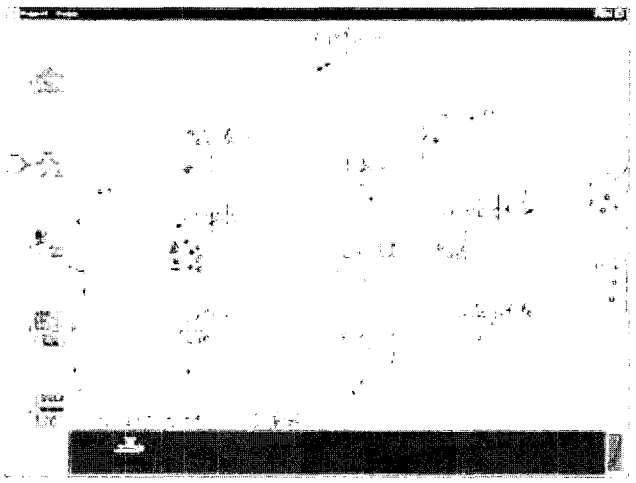


Fig.3 A screenshot of DENIM[9]

On the other hand, most programmers come from the Information Technology (IT)/Computer Science (CS) domain and have trained in administering software engineering methodologies to solve problems. They build applications that include server-side scripts, database applications and other functions for web projects[10]. Programmers use software development to achieve these goals. They may gain knowledge regarding UI design during their training, but they mostly focus on how to make interfaces work technically, without ensuring whether the interfaces are understandable by the designers.

Current web development methodologies and models may not be acceptable for designers, because some elements of the graphical modeling notations are related to data flow or function deployment and therefore do not involve design concepts. So that graphical modeling notations can be the communication medium between designers and programmers, they need to be modified, with a greater focus on the

---

7 Douglas, K., J. Landay and J. Hong (2003) 'Processes for developing customer-centered sites', in *The Design of Sites: Patterns, Principles, and Processes for Crafting a Customer-Centered Web Experience*. Boston: Addison-Wesley, pp. 87-105.

8 Lin, J., M. Newman, J. Hong and J. Landay (2000) 'DENIM: finding a tighter fit between tools and practice for web site design', in *Proceedings of the CHI 2000 Conference on Human Factors in Computing Systems*, April 1-6, The Hague, Netherlands, pp. 510-517.

9 Lin, J., M. Newman, J. Hong and J. Landay (2000) 'DENIM: finding a tighter fit between tools and practice for web site design', in *Proceedings of the CHI 2000 Conference on Human Factors in Computing Systems*, April 1-6, The Hague, Netherlands, pp. 510-517.

10 Burdman, J. (1999) *Collaborative Web Development – Strategies and Best Practices for Web Teams*. Upper Saddle River: Addison-Wesley.

concerns of both, thus becoming an efficient tool between the two domains.

2.3 Communication Problems and Tools

Collaboration benefits the process of complex design like web site design. According to Burdman (1999), however, there are 11 factors that cause poor communication: (1) people have backgrounds in different disciplines; (2) a lack of mutual understanding of terminology; (3) personality; (4) hidden agendas; (5) ineffective meetings; (6) proximity; (7) assumptions; (8) poor infrastructure and support; (9) being an expert; (10) fear; and (11) lack of a good communication structure/system.

Computer supported cooperative work (CSCW) researchers propose a new type of software called CSCW system or groupware[11], which is software for teamwork and can aid communication in an organization and helps staff work together on joint tasks. From basic components or ideas within groupware, more complex forms can be created[12]. Groupware can be defined as software that facilitates work within groups and provides three key group functions: communication, collaboration and coordination[13].

3. Methodology

3.1 Mapping web site design process to web application design process

In Fig.4, we show both the web site development process and web application development process in parallel and identify the development phases that can have similarities to be matched each other.

The products of each phase of processes have certain similarities. For example, in Fig.4, the both products of corresponding phases, 'Discovery' and 'Requirements Specification', concern about requirements gathering from the customers by making scenarios and use cases to make a list of tasks. Another corresponding phase, 'Refinement' and 'Hypertext Design', produces high-fidelity sitemap, storyboards and schematics and site view schema to present the website structure and deploy visual and data components. From the fact we mentioned above, we realize the integration of the products from those two design phases of both web development processes can be a good communication media because the integration is able to be compatible with both sides of their works and processes so that the web designers and programmers can have benefits out of the environment.

---

[11] Ellis, C. and J. Wainer (1994) 'A Conceptual Model of Groupware', in *Proceedings of the ACM conference on Computer supported cooperative work* (CSCW '94) (Smith, Smith and Malone eds., New York: ACM Press), North Carolina, USA, pp. 79-88.

[12] LePoire, D.J. (1999) 'Groupware Diffusion/Adoption: Literature Review and Early Case Studies', in *Proceedings of the 2nd Annual CTI Research Symposium*, November 4, Chicago, USA, http://facweb.cs.depaul.edu/ctiphd/ctirs99/online/lapiore.html (last visited on Oct 17, 2006).

[13] Chaffey, D. (1998) *Groupware, Workflow and Intranets: Reengineering the Enterprise with Collaborative Software.* Boston: Digital Press.

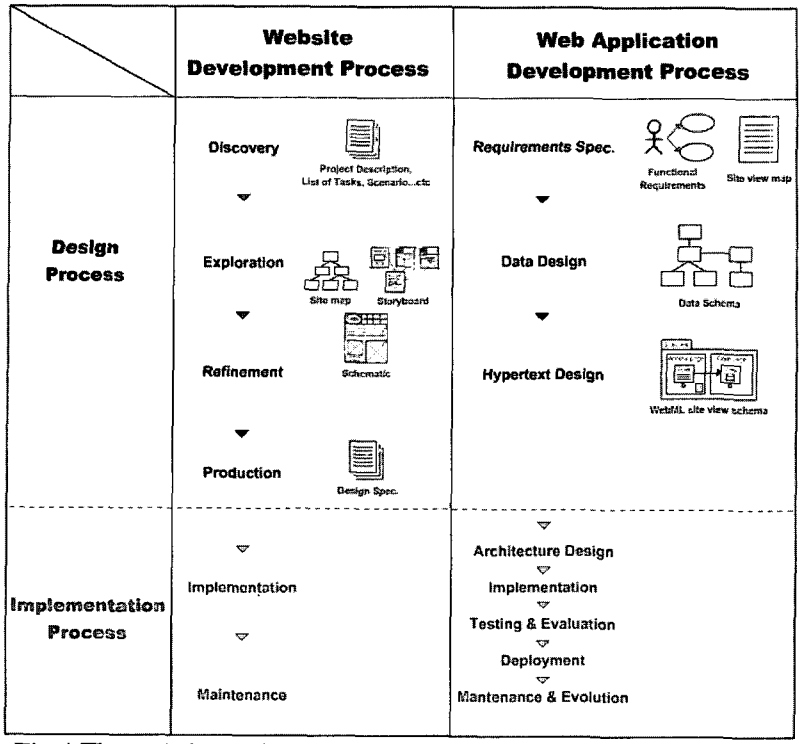


Fig.4 The website design process vs. web application design process

3.2 Revision of web modeling notations

Among several methodologies, we chose the graphical modeling notations from WebML because WebML has intuitive graphical representations and can be easily supported by Computer Aided Software Engineering (CASE) tools. It is also compatible with classical notations like the E-R model, object-oriented model and UML class diagram. Some elements for data flow and functions were removed from WebML to obtain simplified modeling notations to communicate with designers. To achieve this, we first checked all the graphical elements of WebML and analyzed their usage and implementation. We then listed and rearranged all elements and their descriptions. Table 1 shows part of the rearranged WebML notations used in our system and their descriptions.

After rearranging WebML notations, we create two more modeling notations for web designers to help them and to make the concept of the model much clearer (Table 2).

Then we created a page preview function in the modeling notation to simulate notations similar to a storyboard diagram and to make it more useful for designers. At the same time, we take out the notations related to data and functions, which are not necessary for designers (Fig.5). Finally, we obtained the revised notations for a web model, which illustrates a simple and basic model structure. Using the revised modeling notations to represent a web site is not only useful for designers, but also represents a good communication medium.

Table 1 Partial WebML notations that display simple structure[14]

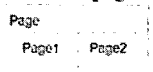
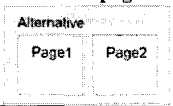
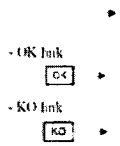
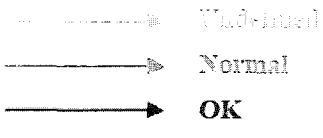
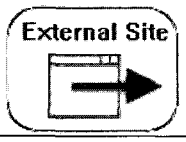
Elements of WebML	Descriptions
Area 	An area is a container of pages or, recursively, other sub-areas, which can be used to give a hierarchical organization to the hypertext.
Page 	A page represents the actual interface browsed by the user. It contains units and/or sub-pages and its attributes include name, landmark, content: units, OR/AND sub pages.
AND Sub-pages 	AND sub-pages are used to divide the page contained in a screen into portions. AND sub-pages are contained in a page or in a sub-page. This describes nested pages and is not related to data.
OR Sub-pages 	OR sub-pages specify that certain portions of the screen may contain alternative pieces of content, each modeled as a distinct page. OR sub-pages are contained in a page or sub-page. Its attributes include nested pages and a default nested page.
Entry Unit 	An entry unit supports form-based data entry. Its only attribute is name, but the user can decide each field's attributes, such as name, type, initial value (optional), modifiability or validity.
SendMail Unit 	The send mail unit provides the capability of sending e-mail messages. Its parameters include: sender, recipients, subject, body and attachments.
Links 	A link is an oriented connection between two units or pages. Links for exiting operations are classified as: – OK link: followed if the operation is successful – KO link: followed if the operation fails

Table 2 New web modeling notations

New Notations	Description
Links 	We add 'undefined' and 'normal' attributes with using different colors to original WebML links notation to present state of links clearer.
External Site Unit 	There are many advertisements, banners, related links or network source that would lead customer to external web site from an internal web site. Web designers can use 'external site unit' notation to avoid misunderstanding.

[14] Chaffey, D. (1998) *Groupware, Workflow and Intranets: Reengineering the Enterprise with Collaborative Software.* Boston: Digital Press.

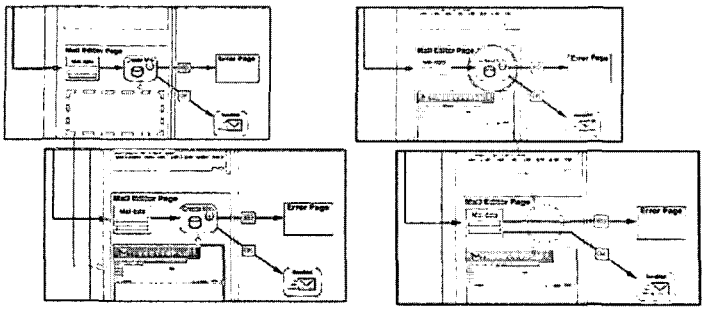


Fig.5 Adding page preview function and omitting the notations related to data or functions

3.3 Integrating Collaborative Modules with Modeling System

To meet the requirements of the three functions of CSCW groupware as we mentioned in section 2.3, we propose a multi- functional collaborative application module, which supports three functions inside: collaborative modeling, whiteboard and video conferencing.

Collaborative modeling function allows users to convert a structured web model from their original set of .html files. Since we create a page review function, the designers are able to get the prototyping view rapidly with preview images and feel friendly toward the modeling notations because they look like a story board diagram.

Whiteboard function allows two or more users to view and draw graphical sketches on a shared modeling tool immediately from different locations. The function can be used during text-based chat, which is also implemented within the function, where each user can work collaboratively on a visual problem.

Video conferencing function allows users to call two- or multi-ways with live visual component. If the bandwidth is enough to make the function in real-time, the amount of information for sending and receiving cannot compare with text-based methods. The function is also useful when visual information is required.

4. System Implementation

4.1 System Architecture

The system that we propose includes three core modules: a *HyperText Markup Language (HTML) page editor*, a *modeler*, and a *conferencing* (Fig.6). We described in section 4.3 the modules in detail.

4.2 System development tools

Our system is developed and programmed by Borland JBuilder 2005 foundational version. The system is based on several packages of open source projects as follows:

JGraph v5.4.7 and **JGraphpad v5.4.4**: constructing and manipulating the web modeler module;

Ekit v1.0: implementing the HTML page editor module; and

jSummit v1.0b: implementing conferencing module.

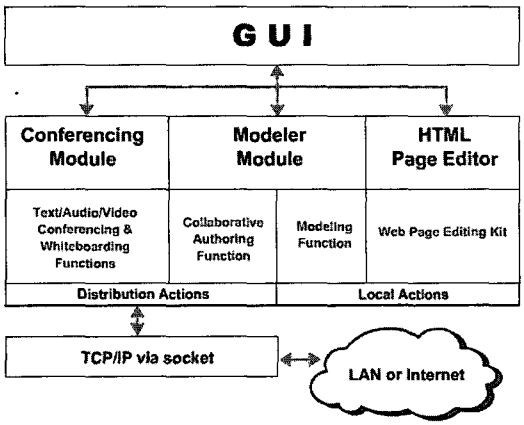


Fig.6 System architecture of our first prototype

4.3 Functional Specification and User Interface

The modeler module has several functions as shown in Fig.7. The modeler module is for converting the original set of .html file to a hierarchical structured web model and displaying the model in the 'Work space' panel. In addition to basic modeling functions of web modeling notations, using the page preview function we create, the designers are able to check the outline of each web page easily and quickly. There is a zoom in/out function to adjust the size of the web page if necessary.

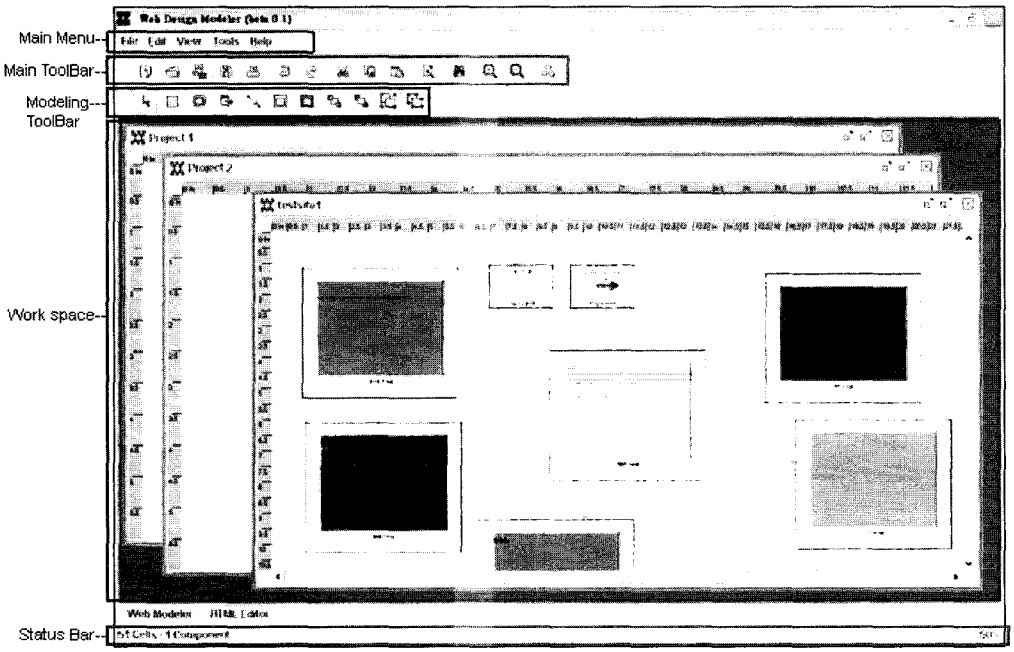


Fig.7 A screenshot of the modeler module

Fig.8 shows a snapshot of HTML page editor.

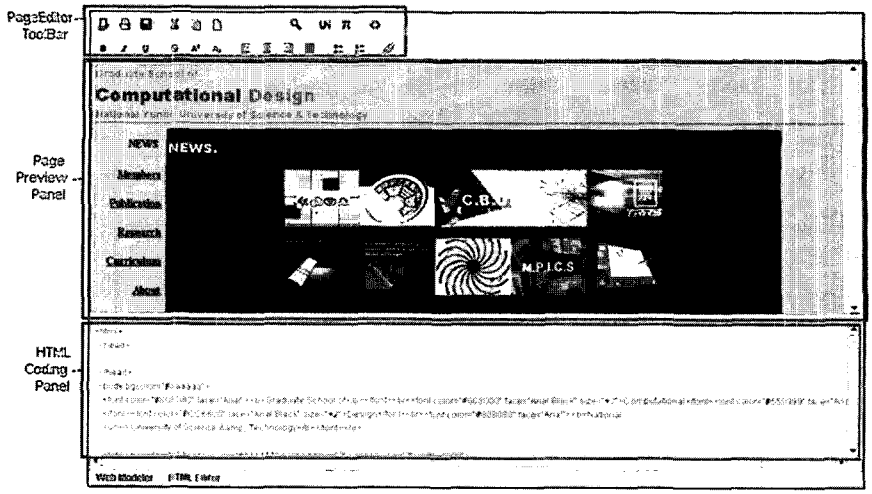


Fig.8 A screenshot of the HTML page editor

This module allows rapid previewing for editing individual pages in the web site map. The module is a simplified version of web authoring tools and is heavily related to the modeler module. The conferencing module includes several on-line communication tools such as text-based chat, audio/video communication (Fig.9) and whiteboard functions (Fig.10).

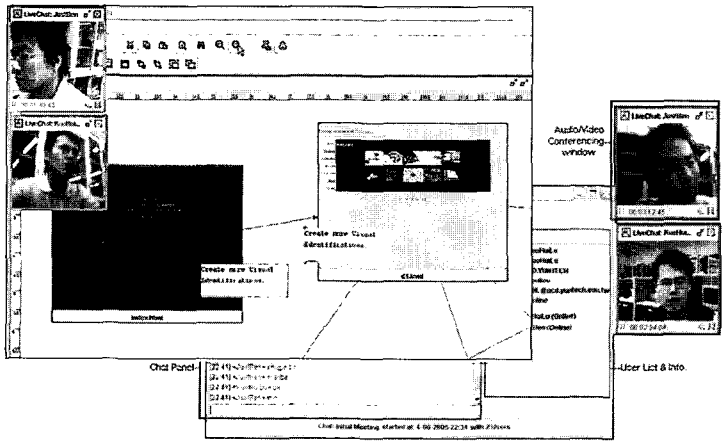


Fig.9 A screenshot of the conferencing module

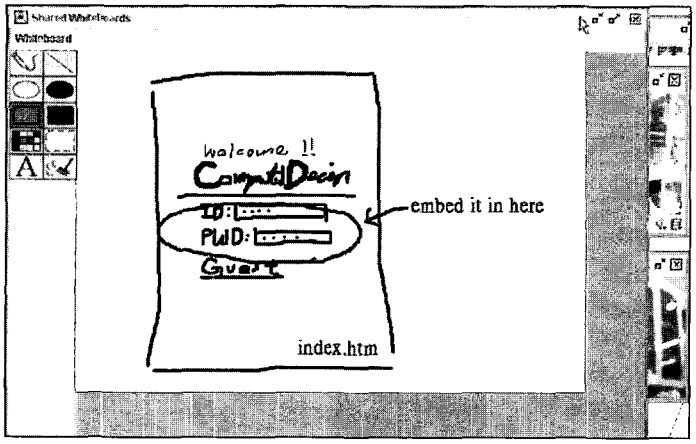


Fig.10 Sketching one's idea during communication

When a web designer is traveling on official business and wants to discuss with a programmer, the designer can execute the 'Collaborative work' function in the conferencing module by sharing his/her workspace as a server and connecting to the network. When the connection is established, they can use the text-based chat or audio/video conferencing function. In this situation, they can share the same workspace of the web modeling and discuss the problems with it. To facilitate the communication efficiently, we implement 'File uploading/downloading' function and 'Post-It' function to attach your idea into the position you want to explain.

As shown in Fig.10, whiteboard function is useful when the user wants to visualize his/her idea and rapidly shows to discuss with others. In summary, this module provides an embedded multimedia communication circumstance to facilitate solving problem between designers and programmers in a shared modeling environment.

5. Conclusions

The interactive computer-aided collaborative system we propose represents an environment for improving communication between web designers and programmers. Out of 11 factors as we mentioned in section 2.3, several communication problems such as 'personality', 'hidden agendas', 'assumptions', and 'being an expert' are related to the human factors, which may not easily find effective solution(s) using computational ways. The paper can, therefore, improve *six problems* within computational environment: (1) people have backgrounds in different disciplines; (2) a lack of mutual understanding of terminology; (3) ineffective meetings; (4) fear; (5) proximity; and (6) lack of a good communication structure/system. Table 3 summarizes the six communication problems, corresponding solutions and system modules.

We also suggest the *revised graphical modeling notations* to illustrate a method for web designers the modeling more visualized and understandable associated with designer workflow. An advantage of using this computational approach is that users can be helpful for dealing with a variety of media types and transferring them to formal graphical modeling notations intuitively. In an integrated computational environment, designers are able to effectively communicate with programmers and accomplish their project easily. This tool, however, only supports the early phase of the whole web design process. Future work could involve supporting communication links in all the web development phases, thus improving web site quality.

References

Burdman, J. (1999) *Collaborative Web Development – Strategies and Best Practices for Web Teams*. Upper Saddle River: Addison-Wesley.

Ceri, S., P. Fraternali, A. Bongio, M. Bambilla, S. Comai and M. Matera (2003) *Designing Data-Intensive Web Applications*. San Francisco: Morgan Kaufmann.

Chaffey, D. (1998) *Groupware, Workflow and Intranets: Reengineering the Enterprise with Collaborative Software*. Boston: Digital Press.

Table 3 The communication problems and solutions

Communication problems		The solutions	Related system modules
1	People come from different disciplines	Revising existing web model to be the communication media that can be cognized for web designers and programmers, so my system can unify the different cognition for design process and terminologies offering modeler module.	Modeler Module: ● Manipulation of web model for discussion. ● Collaborative modeling. ● Export web model to a picture or prototype for discussion.
2	Lack of a mutual understanding of terminology		
3	Ineffective meetings	Web designers can communicate with programmers and reduce the waste of time of meetings, because they don't need to explain detail that programmers don't care. System can support discussion by offer what they need to finish their work.	
4	Fear	System can provide an informal communication environment to reduce pressure and fear in members.	Conferencing Module: ● Text, audio and video conferencing. ● E-mail function. ● Whiteboard for convenient sketches and discussion.
5	Proximity	Providing more necessary, correct information and get more extra advantages from net meeting.	Modeler Module & Conferencing Module.
6	Lack of a good communication structure/system	The system can offer a communication channel and media for designers and programmers, so it has potential to be a key part of a good communication structure.	All Modules in the system.

Douglas, K., J. Landay and J. Hong (2003) 'Processes for developing customer-centered sites', *in The Design of Sites: Patterns, Principles, and Processes for Crafting a Customer-Centered Web Experience.* Boston: Addison-Wesley, pp. 87-105.

Ellis, C. and J. Wainer (1994) 'A Conceptual Model of Groupware', in *Proceedings of the ACM conference on Computer supported cooperative work* (CSCW '94) (Smith, Smith and Malone eds., New York: ACM Press), North Carolina, USA, pp. 79-88.

LePoire, D.J. (1999) 'Groupware Diffusion/Adoption: Literature Review and Early Case Studies', in *Proceedings of the 2nd Annual CTI Research Symposium*, November 4, Chicago, USA, http://facweb.cs.depaul.edu/ctiphd/ctirs99/online/lapiore.html (last visited on Oct 17, 2006).

Lin, J., M. Newman, J. Hong and J. Landay (2000) 'DENIM: finding a tighter fit between tools and practice for web site design', in *Proceedings of the CHI 2000 Conference on Human Factors in Computing Systems*, April 1-6, The Hague, Netherlands, pp. 510-517.

McDonald, A. and R. Welland (2001) 'Web engineering in practice', in *Proceedings of the 10th WWW10 Workshop on Web Engineering*, May 1-5, Hong Kong.

Newman, M. and J. Landay (2000) 'Sitemaps, storyboards, and specifications: A sketch of web site design practice', in *Proceedings of an International Conference on Designing Interactive Systems*, August 17-19, New York, USA, pp. 263-274.