

WEB-BASED FAMILY NOTICEBOARD

Wenyu Huang, Andy Sloane and Arthur Harris

CoNTACT Research Group, School of Computing and Information Technology, University of Wolverhampton, Wolverhampton, United Kingdom.

Key words: Home, family, web application, user interface and interaction, DHTML, HTTP, notice, web server, web browser, java, servlet, database.

Abstract: We outline how to implement a three-tier family notice board application over the Web using ready-made web browsers as the interactive medium. The application deploys the dynamic HTML layers to create an intuitive, simple and yet powerful user interface designed for general home users who have little experience with computer interaction or windows interactive style. This paper shows the design initiative, objectives, principles, and the choice of underlying technologies in respects of the browser, server and database. It also details the solutions we found to overcome the constraints imposed by the interactive medium and the underlying technologies.

1. INTRODUCTION

The rapidly evolving World Wide Web and penetration of computers into the home presents an opportunity to provide new ways of electronic communication between family members locally and remotely. For experienced computer users, there are already email systems at present, but its interface is not simple enough to appeal especially to non-computer-experienced users at home, and it is not designed for family communal notice exchange. Specifically, our web-based family notice board (it will be called Noteboard hereafter) is designated as an information application for naive home users. Moreover, it allows multiple users concurrently and provides distributed access.

A web application has advantages, over native code, of widely distributed access whilst allowing easy changes. However, the design of HTML, HTTP, and scripting languages impose constraints on the interface design. This paper documents the design objectives, principles, and their implementation. At the same time, the constraints and the correspondent choice are detailed.

2. FAMILY NOTICE BOARD INITIATIVES

The driving forces behind this project are the user requirements and technology readiness. We seek the best solution based on the near future condition of home computing infrastructure and the availability of software technologies.

2.1 User needs

The home user needs secure, cheap, and easy to use communication systems to give them more convenience. IBM has investigated home customers' requirements and concerns with respect to the integrated home networking infrastructure. From the user requirements of safety, economy, and convenience, the following user needs were derived: Using home PC to facilitate communications both within the family, and between family members and the outside world [2].

Furthermore, households with multiple dwelling places, or with their members in different locations, have experience of talking via telephone, leaving voice mail on answering machines, sending email, or posting letters, to other members of the family. People need multiple means to communicate when family members are not in the same place. All the means mentioned above are basically one-to-one personal communications. Family notice exchange has a communal characteristic. These are the reasons for consideration of a web-based family notice board for the "distributed home".

Finally, in our user requirement investigation, we interviewed five persons drawn from convenient samples. The interviews were guided by predefined open questions. The interviewees and their families have established some non-electronic methods for family notice exchange at home. For example, one may write a message on notepaper and position it somewhere as a family "notice board", such as on the dinner table or on the fridge. The amount of this kind of note exchange per day was not as significant as emails or telephone calls. However, it is a complementary method for communication when the emails and telephones are not available or convenient. It is a necessity when communal message exchange is required. On the other hand, people often found that the notepad or the

pencil/pen was missing when they wanted to write a note. Sometimes they noticed the note had not drawn the recipients' attention. One of the interviewees indicated that sometimes the recipients pretended they had not read the note when they did not want to do what they were asked in the note. In the design of Noteboard, these scenarios were considered.

3. USER ABILITY

Considering the explosion of the popularity of the web, we assume that most of our target home users have some experience with web browsers. However, we also realise that some of the home users have no such experience at all or even have techno-phobia, such as some of the elderly, or young children. The application then should have as little learning curve as possible for them.

4. HOME COMPUTING INFRASTRUCTURE

At present, home-computing infrastructure normally has at least one personal computer. This usually has the capability of web browsing and of hosting a simple web server. Better still is Internet access and a home network between the home computers. Noteboard can be implemented on the basic requirement, that is, one personal computer, which is not necessarily running the Windows Operating System provided it can host a web server and a web browser. Noteboard can exploit more powerful infrastructure to serve remote access and better performance over web.

4.1 Software technology chosen and their constraints

The foundations of web applications are object-orientation and component technology for cross platform applications, browser-side scripting, the three-tier architecture (browser, server, database). [3]

The basic protocol for the web is HTTP. It enables global access to an application, but for security reasons, it prevents a web application from visiting the native file system. Therefore we need the three-tier architecture to run applications over the web. A servlet-enabled web server forms the middle tier of the web application. Java is the widespread object-oriented technology for cross platform applications. Its servlet package provides a mean of server-side programming to support web applications running remote interfaces. We use the Java™ SQL package and JDBC™ driver via

Microsoft® ODBC driver to access the database (the third tier) which, in this case, stores the data the native file system prohibits.

On the browser side, recently developed DHTML technology enriches web applications' interfaces although it is not as flexible as Windows tools for native applications [3]. JavaScript or Jscript, the object-based scripting language, fulfils the first-tier control of Dynamic HTML interfaces on web browsers. We explored the potential of DHTML on layers in the web user interface design.

5. ANALYSIS AND DESIGN

In the next section we describe the design objectives, principles, constraints imposed by underlying technology, solutions and implementation of the design.

5.1 Objectives, principles and the interface design

Our overall goal is to create a rational environment for a group of users to interchange notices communally and remotely. The design principle is to lower the cognitive burden wherever it is possible [5]. Therefore, the system is designed as such so that users can learn to use it by clicking anywhere on the screen without missing out important functions or causing serious consequences.

Generally family members would easily accept a simple and intuitive interface [5]. Noteboard's user interface has as few as possible different views/pages as outlined in Table 1. The minimum number of clicks required by a task [7], is shown in Table 2 compared with a bench mark system. Obviously, the latter has much more higher learning curve and the same or more clicks to fulfil a task.

Table 1. Amount of objects in the system

Pages	Views on pages	Sub-views on notice-reading	Sub-views on notice-writing
1	1. notice-reading 2. notice-writing	1. received notices 2. sent notices 3. bin	0

It avoids right clicks and selecting from lists or menus that requires users' sophisticated control over a mouse, which could be difficult for the elderly and the young. The interface allows users to easily acquire the skills required

and efficiently use the system without formal training or reference to manuals. Figure 1 shows a screenshot of the interface.

Table 2. Amount of clicks required to fulfil a task

Task	Task description	Clicks	Microsoft® Outlook Express (an email system)
1	reading notices destined to someone	1	0 or 1 if not in the default folder
2	reading notices sent by someone	1	0 + preference setting, or 4
3	reading notices in the bin	1	1
4	discarding a notice into the bin during task 1 and 2	1	1
5	restoring a notice back onto the notice board during task 3	1	4 click or 1 drag
6	destroying a notice in the bin during task 3	1	1
7	destroying all notices in the bin during task 3	1	3
8	new notice notification	0	0
9	new notice identifying during task 1 and 2	0	0
10	checking if a notice has been read	1	1
11	to get ready to write a notice	1	1
12	setting or removing a receiver of a notice during notice-writing	1	1 to 4
13	setting or removing a sender of a notice during notice-writing	1	Not available after preference setting
14	preference setting	0	Numerous and learning

Low cost is another user requirement to be satisfied. Therefore, the application makes use of free software as much as possible to reduce the cost of the system. At present it is running on Microsoft® Internet Explorer 4.0 or later or Netscape 4.0 or later, JSWDK 1.0 (servlet-enabled web server) and JRE (Java™ Runtime Environment). They are free and available across platforms although the underlying database system Microsoft® Access is not so. However, it is relatively easy to migrate a database from Access to other platforms.

Minimising system management suits most families where there is rarely a professional technician or administrator at hand. Noteboard facilitates any one of the users as family system administrator who could centrally maintain the notice board for the whole family, but does not burden him/her with the task. His/her only responsibility is setting up the photographs that represent family members as required and tidying up unwanted notices periodically.

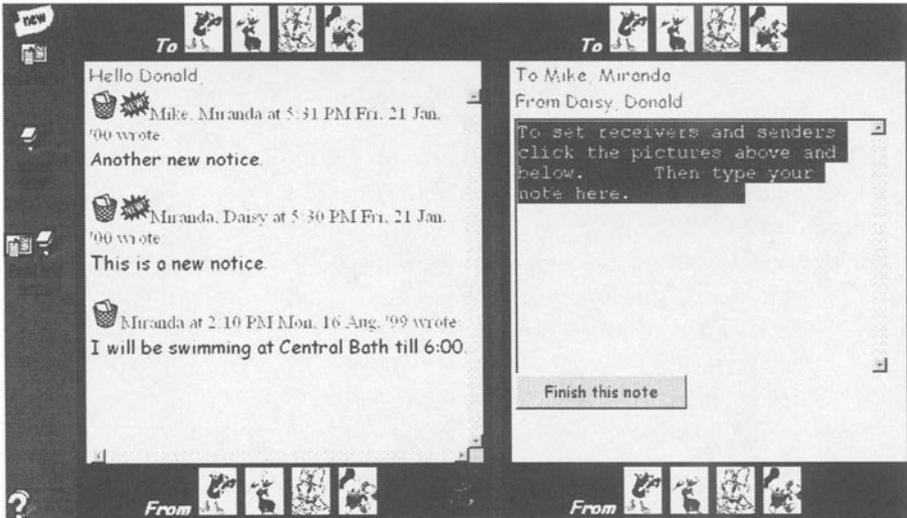


Figure 1. The interface's features⁴⁴

The interface metaphor from daily documentation processing is employed (see Figures 1 and 2) to aid users, who are familiar with the non-electronic notice exchange, to master the electronic application.

5.2 Constraints and resolutions

Because web applications can not access local file systems, the three-tier architecture is employed and the third-tier database is used to store the necessary data. HTML has two ways to transfer data to the server, clicking an anchor and then following a link or submitting a form to a server and

⁴⁴ 1. It presents persons' pictures and icons as an anchor, which invoke JavaScript and/or servlets to fulfil certain tasks. 2. It uses icons to present the status of messages, such as indicating notices thrown in the bin. 3. It uses links (on the left-hand side) as a menu leading to different functions, which are represented in various dynamic layers. 4. It uses bubble speech displaying brief help when users place the mouse over an object. 5. It groups persons' pictures as a select list, which is easily manipulated by users.

getting a new page from the server as a response. Consequently, the only way for the server to transmit information to a browser is in response to such click actions or the form submit initiated by the user [3]. In the past, the browser could only show a whole new page after following links or form submit on the whole browser window or a frame. (Use of frames was rejected in our application due to its serious drawbacks on web interface design [6].) With recent Dynamic HTML development, the DHTML layer's external file loading adds a new method of browser-server communication. By loading an external file from a server, a layer sends data with the URL to the server, the server returns a new file to the browser. Noteboard follows such layer loading action to show dynamic HTML files (the server's responses) on any layers placed on any part of the browser window.

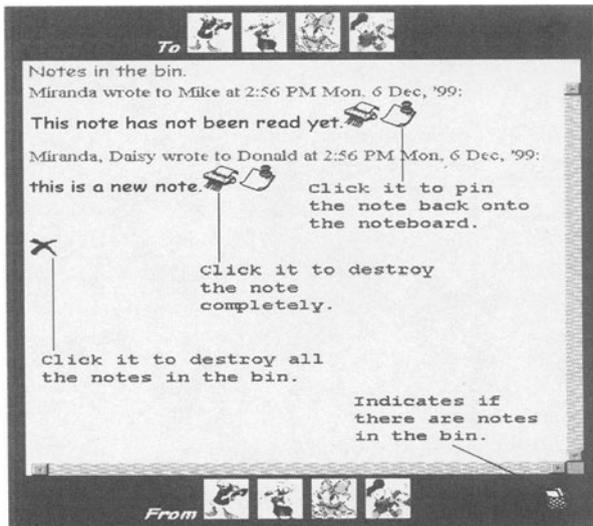


Figure 2. An annotated screenshot as a context-aware help

JavaScript's ability to load a layer's external file and layer-writing combine with Java™ servlets on the server to offer a more sophisticated control over the web interface [4]. A DHTML layer acts as an independent display area anywhere on the DHTML page. Such a layer can be shown, hidden, moved and embedded into another layer. JavaScript can dynamically write DHTML statements to a layer and load an external DHTML file onto a layer after user clicks an anchor. At first, the URI for loading an external file can be a HTTP request to a java servlet on a server, and can contain data and status transmitted to the servlet as the request's parameter values. Then, the servlet gets connections to a database if necessary, queries and manipulates the tables in the database, and generates a new HTML file (the external file

of the layer asked for). The server transmits the new file back to the browser. Finally, the browser shows the new file on the designated layer. Figure 3, 4, 5 and 6 shows the data flow between the browser, the server, and the database.

It is recommended that the dynamic layer only load static content. Displaying dynamic HTML file on a layer is claimed difficult, but Noteboard realises displaying dynamic HTML files on dynamic layers in practice. It is essential for designing our powerful web interface. We believe the barrier is the concurrency of the browser and server communication. Noteboard accomplishes asynchronous communication between browser and server, which was claimed impossible [3]. We utilise the fact that in the <body> tag the 'onload' event invoked JavaScript is always executed after the body loaded. For example, say two DHTML files A and B need to be loaded asynchronously from the server; we load file B in file A's 'onload' JavaScript.

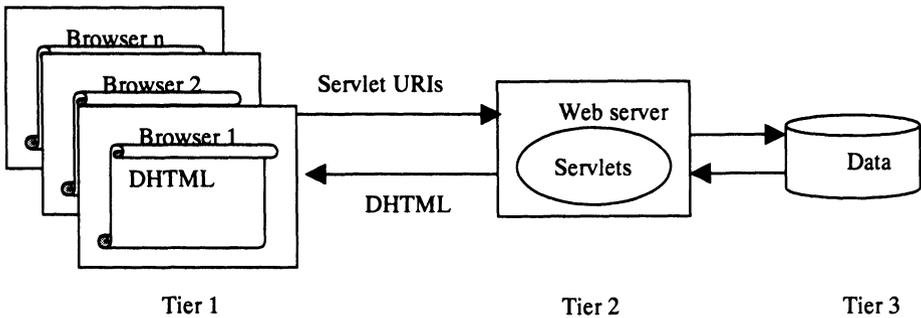


Figure 3. Three-tier architecture

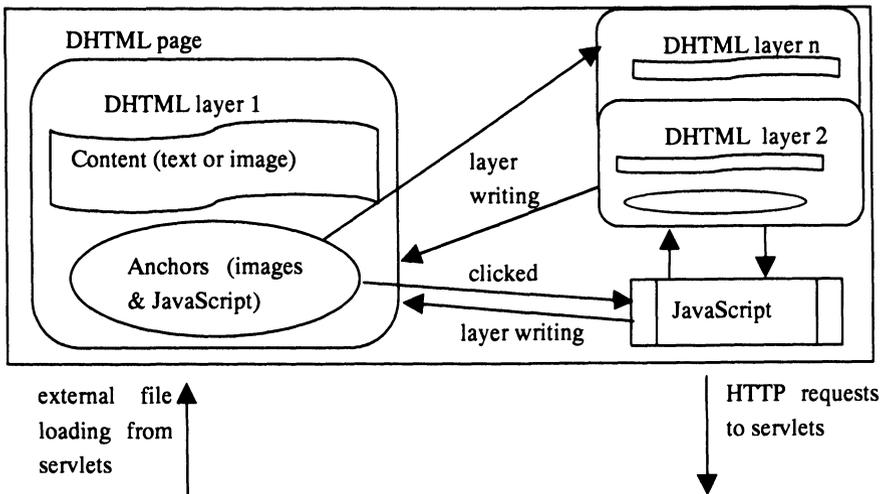


Figure 4. Tier 1 presentation scenarios

The browser normally displays new HTML files from the server. Sometimes we do not need such response except when sending data to the server; sometimes we only need non-html data for JavaScript's further processing. In these situations, we set up a hidden layer to store the pseudo-response or the non-HTML data. This method broadens the communication content and gains more sophisticated control over the web interface.

6. PROBLEMS AND FUTURE

In the implementation of the web application, we encountered various problems. Most of them are on the browser side. The different browsers' implementations of HTML, Cascading style sheets and scripting languages cause numerous difficulties in our attempt to develop the application across-browsers. The built-in object models in the Internet Explorer (IE) and Netscape are different. For examples, the object layer has different properties and is in different hierarchical positions; Netscape's layer content is not scrollable; IE's 'back' button does not work properly when the page contains dynamic layers; the text rendering format in the dynamic layer on the two browsers are quite different. By solving these problems, we have made Noteboard available across browsers.

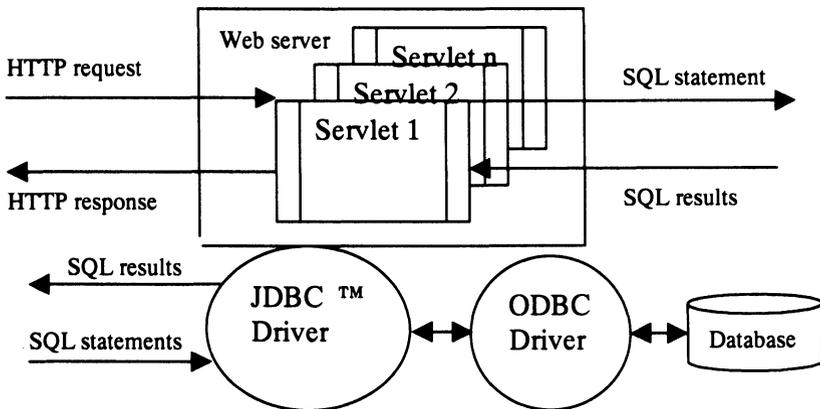


Figure 5. Tier 3 Database connection scenarios

Netscape does not generate a scrollbar on its browser window when the layer on a page is larger than the window. HTML does not support some of the basic immediate feedback normally found in the Windows user interface, such as to grey out or to highlight the picture or the icon just selected by the user. Such problems need to be addressed in future.

Family notice exchange is merely one kind of information flow around the home. Others include web information, television programmes, family historical records, and so on. In order to gain efficiency and convenience, Noteboard should be part of an integrated home information system, for which we will build gradually. The nature of the hyper-linked HTML documents gives the application the potential to be seamlessly integrated into future home information systems that should be interoperable with web application.

7. CONCLUSION

Noteboard is a three-tier distribution system for writing, reading, deleting notices, and notifying notice status between a group of certain users over the web. It is significantly different from other Internet communication applications because it deploys DHTML layers to assemble an intuitive interface, and allows communal notice reading and writing. Its implementation is different because it employs a server and a central database to manage all notices for all users in the group.

REFERENCES

- [1] Schussel G. (1996) "Client/Server: Past, Present and Future", <http://news.dci.com/geos/dbsejava.htm> (accessed on 10, Aug 1999).
- [2] Rusnak (1997) "Anywhere in the home", IEEE COMMUNICATIONS SOCIETY, 1997 Fourth International Workshop on Community Networking proceedings, Sept. 11-12, 1997, Atlanta, GA, USA. New York, (USA): IEEE Communications Society Press, 19-24.
- [3] Rice J. (1996) "Using the web instead of a windows system", CHI 96 Vancouver, BC, Canada, 103-110.
- [4] Steinman D. (1999) The Dynamic Duo Cross-browser Dynamic HTML. <http://www.dansteinman.com/dynduo/> (accessed in October 1999).
- [5] Nielsen, J. (1993) "Usability engineering", Academic Press, London.
- [6] Nielsen, J. (1996) "Why frame suck (Most of the time)", in Nielsen J. ed. "The Alertbox: Current Issues In Web Usability", <http://www.useit.com/alertbox/9612.html>, (accessed on 23 June 1999).
- [7] Nielsen, J. (2000) "Hard-to-use sites will fail", in COMPUTIMES of The Irish Times, <http://www.ireland.com/newspaper/computimes/2000/0110/compu1.htm>. (accessed on 11 Jan 2000).