

Integrating Databases, Objects and the World-Wide Web for Collaboration in Architectural Design

Wassim Jabi, Assistant Professor

Department of Architecture
University at Buffalo, State University of New York
jabj@ap.buffalo.edu
<http://www.ap.buffalo.edu/~jabj/>

Abstract

Architectural design requires specialized vertical knowledge that goes beyond the sharing of marks on paper or the multi-casting of video images. This paper briefly surveys the state-of-the-art in groupware and outlines the need for vertical and integrated support of synchronous and asynchronous design collaboration. The paper also describes a software prototype (WebOutliner) under development that uses a three-tier persistent object-oriented, web-based technology for a richer representation of hierarchical architectural artifacts using Apple's WebObjects technology. The prototype contributes to earlier work that defined a framework for a shared workspace consisting of Participants, Tasks, Proposals, and Artifacts. These four elements have been found through observation and analysis to be adequate representations of the essential components of collaborative architectural design. These components are also hierarchical which allows users to filter information, copy completed solutions to other parts of the program, analyze and compare design parameters and aggregate hierarchical amounts. Given its object orientation, the represented artifacts have built-in data and methods that allow them to respond to user actions and manage their own sub-artifacts. In addition, the prototype integrates this technology with Java tools for ubiquitous synchronous web-based access. The prototype uses architectural programming (defining the spatial program of a building) and early conceptual design as examples of seamlessly integrated groupware applications.

Keywords

Computer-Supported Collaborative Design, WebObjects, Synchronous and Asynchronous Collaboration, Java Applets, Application Server, Web-based Interface.

Introduction

Software that supports collaborative work (groupware) usually belongs to one of two categories: synchronous and asynchronous. Synchronous groupware supports the real-time aspects of collaborative work such as group meetings. Examples of synchronous groupware include chat programs, video-conferencing, application sharing, and shared whiteboards. Asynchronous groupware supports the longer-term aspects of collaboration such as workflow and project management, issue discussions, and review processes. Examples of asynchronous groupware include e-mail, newsgroups, shared workspaces, and task management software. Some aspects of collaborative work, such as voting, can happen either synchronously or asynchronously. Those types of activities need software support that is flexible enough to operate in either mode. While current applications may support one or more of these functions, many have failed to fully integrate the synchronous and asynchronous requirements of collaborative work. Furthermore, the majority of groupware applications supports general-purpose collaboration and, thus, is not equipped to support domain-specific functionality. Architectural design requires specialized vertical knowledge that goes beyond the sharing of marks on paper or the multi-casting of video images. This paper briefly surveys current groupware applications and outlines the need for vertical and integrated support of synchronous and asynchronous design collaboration. The paper also describes a software prototype under development that uses a three-tier persistent object-oriented, web-based technology for a richer representation of hierarchical architectural artifacts using Apple's WebObjects technology (Apple, 1998). The prototype contributes to earlier work (Jabi, 1996a, 1996b, 1998) that defined a framework for a shared workspace consisting of Participants, Tasks, Proposals, and Artifacts. These four elements have been found through observation and analysis to be adequate representations of the essential components of collaborative architectural design. These components are also hierarchical which allows users to filter information, analyze and compare design parameters and aggregate hierarchical amounts. Given its object orientation, the represented artifacts have built-in data and methods that allow them to respond to user actions and manage their own sub-artifacts. In addition, the prototype integrates this technology with Java tools for ubiquitous synchronous web-based access. The prototype uses architectural programming (defining the spatial program of a building) and early conceptual design as examples of seamlessly integrated groupware applications.

Examples of current groupware applications

Given the lack of bandwidth, real-time videoconferencing has met limited success and is only available to large corporations and to a relatively small number of distance-learning centers within academic institutions. Instead, many users of the Internet rely on desktop conferencing applications such as Cornell University's CU-SEEME and Microsoft's NetMeeting for near real-time audio and video communication (Figure 1). NetMeeting also has whiteboard capabilities, the exchange of files and the sharing of applications. Aside from reliability problems and network congestion, NetMeeting on its own provides only general-purpose collaboration functionality. The sharing of single-user applications is not the most effective way to multi-author a document. NetMeeting shines in its one-on-one videoconferencing capability. However, CU-SEEME has the advantage of its ability to multi-cast several video images such that a group meeting can take place.

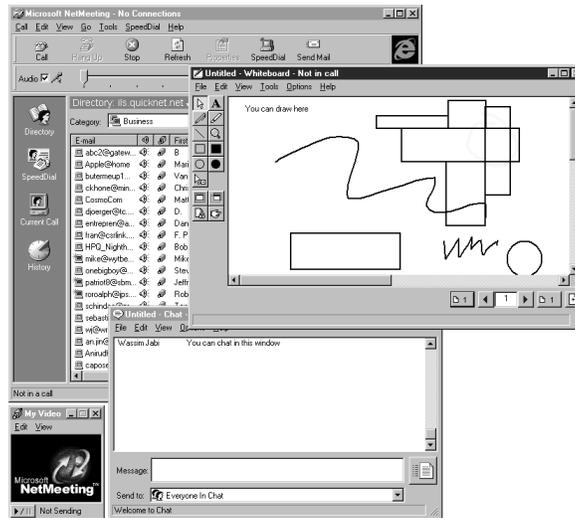


Figure 1: Microsoft NetMeeting Screen Shot.

A new breed of web-based collaboration technologies is based on the idea of a shared virtual workspace. In some cases, the application may use a synthetic three-dimensional space that multiple users can inhabit and meet through a representational avatar. In other cases, these applications, such as the one webex.com offers for free, allows you to have a non-dimensional virtual office. Using your web browser, you can upload documents and guests can leave you messages, request meetings, check your shared calendar and conduct real-time meetings using chat software, a whiteboard, and multi-casting of PowerPoint presentations.

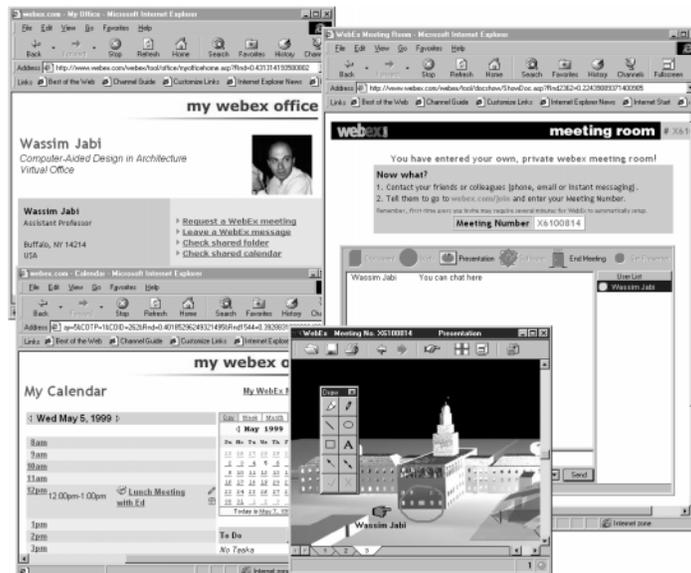


Figure 2: Webex.com Virtual Office Screen Shot.

With the advent of the Java programming language, several smaller shareware applications (applets) are available that allow whiteboard functionality, document annotation, and real-time textual chat. An example of that class of software is Groupboard from www.groupboard.com (Figure 3). Again, we find that this type of application is suited for casual meetings that need only to sketch and chat without the need to carry out organized tasks that need any specialized software.

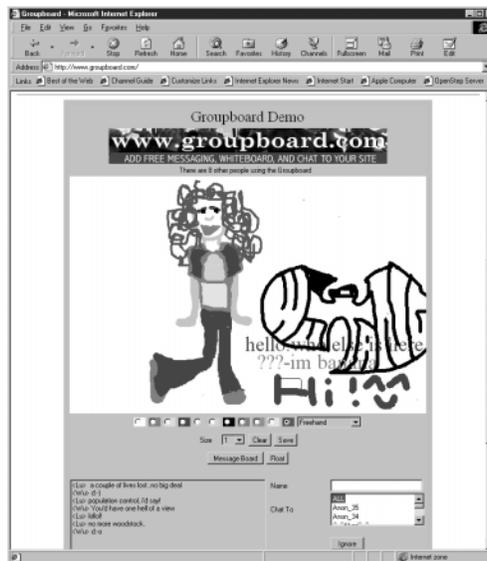


Figure 3: Groupboard.com Whiteboard Screen Shot.

The Case for Domain-Specific Groupware

General-purpose groupware applications such as those described in the previous section are very well suited for their intended purpose. They allow geographically dispersed users to meet, communicate, point at, annotate and share synthetic artifacts in near-real time. Some also allow the remote control of a computer or application. Yet, there is a whole class of needs that is not addressed in these applications (Nardi, 1994). For example, if a musician would like to collaboratively create a piece of music, then someone must write a collaborative music authoring application from the ground up or convert a single user application into a multi-user one. In most cases, the latter has proved to be inadequate due to differences in user interaction modes. The same can be applied to the field of architecture. Architects sometimes need to carry out domain-specific tasks that general-purpose collaborative tools simply cannot handle. In this paper, I use the example of co-creating a building program (the specification of the spatial requirements for a building) to illustrate the need for domain-specific groupware. It is important to not here that I am not arguing for a replacement of current collaboration software, but simply the need to augment them.

WebObjects

Apple's WebObjects software is a high performance application server (Figure 4). It provides an object-oriented application framework for developing three-tier client-server applications as well as dynamic publishing. The three tiers are (1) Storage, (2) Application-Layer, (3) Web Interface. By using components in templates that are linked to actions and data in the WebObjects application, the server can act on user actions dynamically, configure a response HTML page and return that to the user for further action. WebObjects provides mechanisms for maintaining information, called state, during a session or even after a session has terminated. Because of that functionality, multiple participants (clients) can view and edit the same information while being logged on to the same WebObjects application.

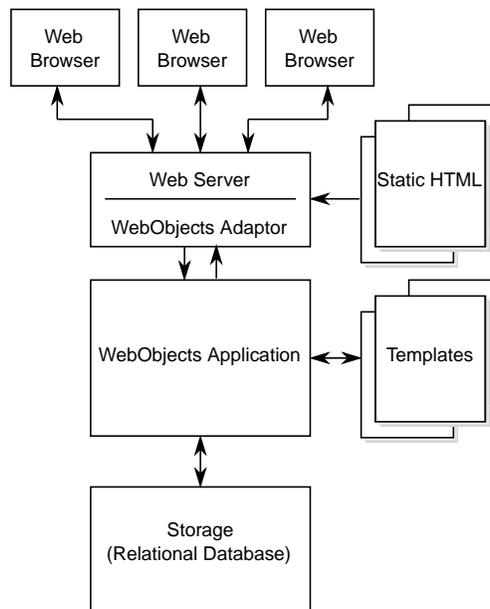


Figure 4: Apple WebObjects framework.

WebOutliner

Fundamentally, WebOutliner is a web-based application that represents and manipulates hierarchical elements using Apple's WebObjects technology. WebOutliner is composed of reusable components that can be adapted for several needs. In its current implementation, WebOutliner manages the components of a building program. Following an object-oriented approach, WebOutliner's components belong to one of three categories: (1) Model, (2) View, and (3) Controller. The Model is mainly composed of a Tree-like object that stores data and methods that represent an architectural space. A space can contain within it other sub-spaces and so on. The total area of a space is computed as the sum of the areas of its descendant spaces plus an additional area that is unassigned to any of its descendants. A space manages its own sub-spaces. For example, if a space is copied and pasted as a sub-space of another space, it will copy and

paste with it all its own descendant sub-spaces. In practical terms, a user can copy a whole department with multiple sub-rooms from one level and paste it on a second level. All the sub-rooms will be copied with it. If a space is edited, added, or deleted, the area of its parent space will be re-computed to reflect that change.

Asynchronous collaboration

When a user connects to WebOutliner's web site, he/she is presented with the current status of a building program. The user is asked to enter his/her name for identification purposes. A magnifier button is provided that enables the user to search for a particular item in the hierarchy (Figure 5).

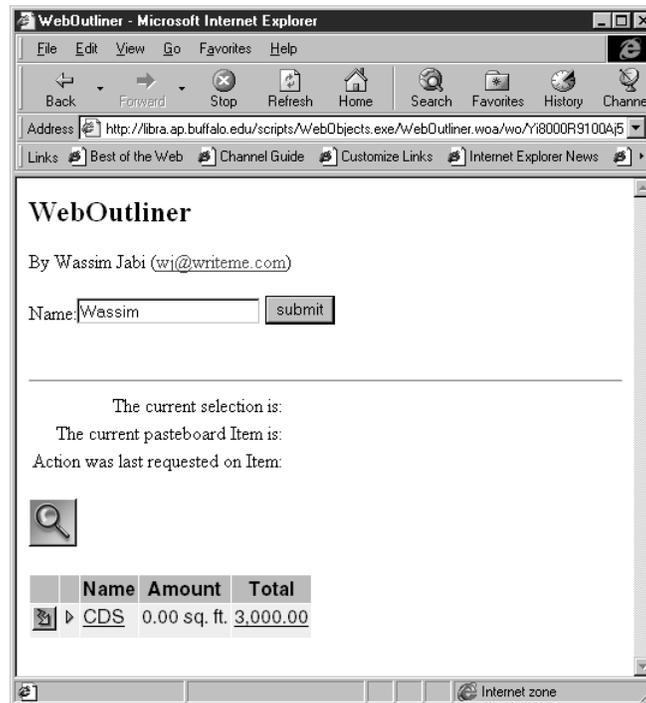


Figure 5: Initial WebOutliner Screen.

Next, a hierarchical list is presented. By clicking on the triangle next to an item, a user can collapse or expand an item to reveal or hide its sub-items (Figure 6). By clicking on the underlined item name, that item is selected and a toolbar of options is displayed that allow the editing and manipulation of that item (Figure 7). The user is able to copy/cut/paste/delete items to edit the hierarchy. He/she can also re-order the items that have a common parent item. The user can also zoom-in on an item temporarily making it the root item. Using this mechanism, a user can filter the overall building program and concentrate only on part of it (Figure 8). By clicking on the Edit Item button, a user can change the name of an item and enter a numeric value. In this example, we are using the numeric value to represent area (Figure 9). Obviously, it is a simple matter to extend the application to include other parameters such as cost and volume. Once the area is entered, the overall area is aggregated using a recursive algorithm (Figure 10).

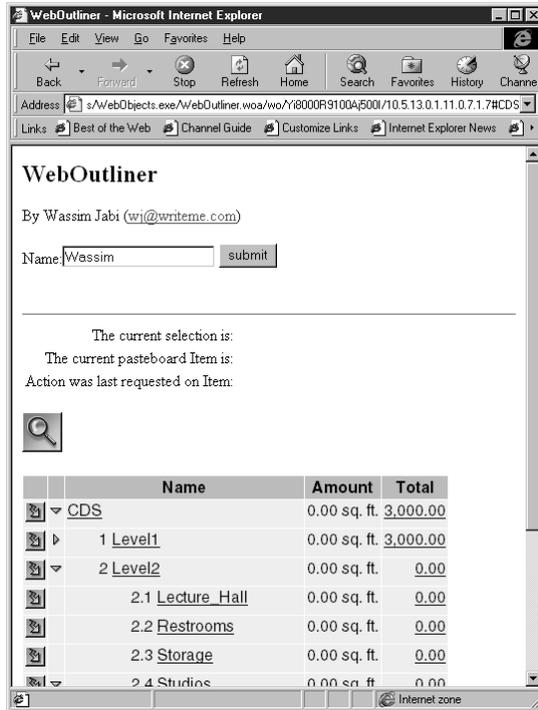


Figure 6: Expanded Building Program

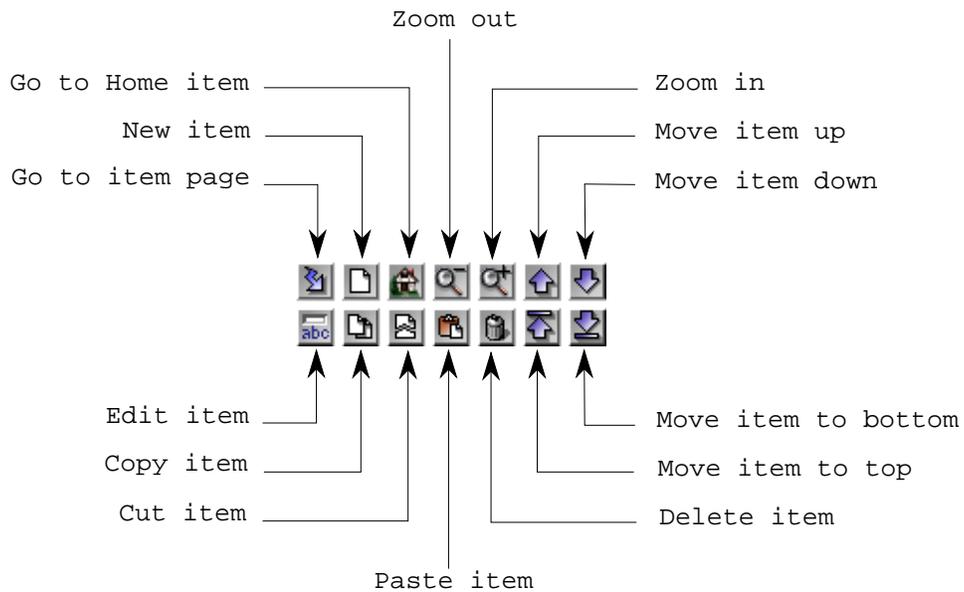


Figure 7: Toolbar and Commands.



	Name	Amount	Total
	2 <u>Level2</u>	0.00 sq. ft.	<u>0.00</u>
	2.1 <u>Lecture_Hall</u>	0.00 sq. ft.	<u>0.00</u>
	2.2 <u>Restrooms</u>	0.00 sq. ft.	<u>0.00</u>
	2.3 <u>Storage</u>	0.00 sq. ft.	<u>0.00</u>
	2.4 <u>Studios</u>	0.00 sq. ft.	<u>0.00</u>
	2.4.1 <u>test studio</u>	0.00 sq. ft.	<u>0.00</u>
	2.4.1.1 <u>Untitled</u>	0.00 sq. ft.	<u>0.00</u>
	2.4.2 <u>Workshop</u>	0.00 sq. ft.	<u>0.00</u>

Figure 8: Zoomed-in on a Sub-item to Filter Out Information.

	Name	Amount	Total
	▼ CDS	0.00 sq. ft.	<u>3,000.00</u>
	▶ 1 <u>Level1</u>	0.00 sq. ft.	<u>3,000.00</u>
	▼ 2 <u>Level2</u>	0.00 sq. ft.	<u>0.00</u>
	2.1	<input type="text" value="800"/>	<u>0.00</u>
	<input type="text" value="Lecture_Hall"/>	sq. ft.	
	2.2 <u>Restrooms</u>	0.00 sq. ft.	<u>0.00</u>
	2.3 <u>Storage</u>	0.00 sq. ft.	<u>0.00</u>
	2.4 <u>Studios</u>	0.00 sq. ft.	<u>0.00</u>
	2.4.1 <u>test</u>	0.00 sq. ft.	<u>0.00</u>
	studio		

Figure 9: Editing Name and Area of an Element.

	Name	Amount	Total
[-]	CDS	0.00 sq. ft.	3,800.00
[+]	1 Level1	0.00 sq. ft.	3,000.00
[-]	2 Level2	0.00 sq. ft.	800.00
[+]	2.1 Lecture_Hall	800.00 sq. ft.	800.00
[-]	2.2 Restrooms	0.00 sq. ft.	0.00
[-]	2.3 Storage	0.00 sq. ft.	0.00
[-]	2.4 Studios	0.00 sq. ft.	0.00

Figure 10: The Building Program Computes and Maintains an Aggregate Area.

Seamless transition to synchronous collaboration

By clicking on the upper left button, a user can go to that item's particular page to examine it in more detail. It is in this simple move that a user transitions to synchronous collaboration. If another user happens to perform the same function (i.e. both participants are interested in the same item at the same time), the users will be presented with a page that contains real-time Java applets for chat, and a shared whiteboard. They are immediately aware of who else is working on this item and they can communicate with them in near real-time (Figure 11).

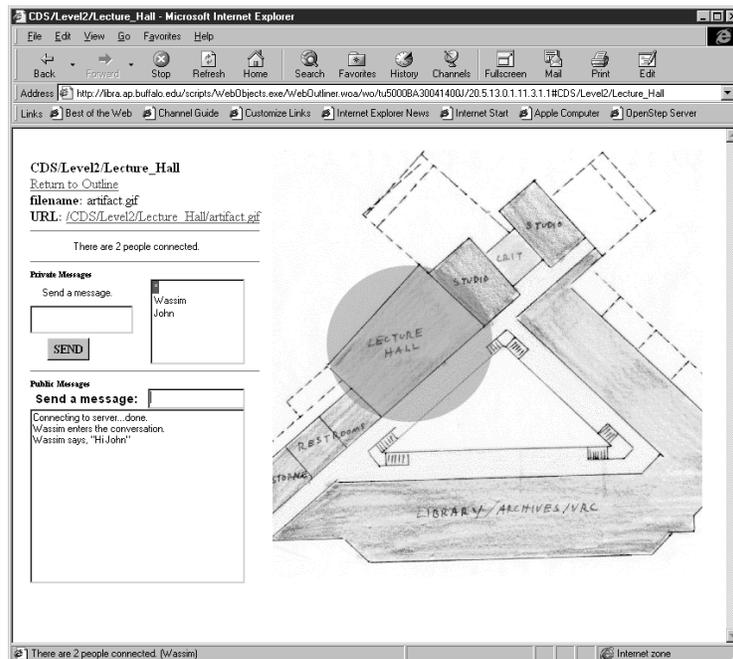


Figure 11: Synchronous Collaboration on an Item in the Hierarchy.

The power of WebObjects is evident here in that, the design of the application only has one template for the item's page. However, by assigning the item's unique ID to that template, multiple users can synchronously collaborate on multiple items at the same item. For example, John and Mary can be collaborating synchronously on the design of the Lecture Hall while Ed and Jane can be discussing the design of the Restrooms. At the same time, Bob is adding more spaces to Level 1.

It is important to note that, currently, the represented artifact is a simple GIF image. Yet, theoretically any synthetic artifact can be uploaded embedded in this page, and displayed using the proper plug-in (e.g. VRML file, animations).

Conclusion

This paper illustrates the need for domain-specific collaborative software. Integrating a database, an intelligent object-oriented application server, and a web-based interface proved to be a powerful and flexible solution that can be used to customize behavior and add intelligence to shared artifacts. The specification of a building program is only an example of a domain-specific functionality that cannot be found in any of the commercial applications available today. This new class of tools should be regarded as complimentary to current general-purpose tools.

As mentioned earlier, WebOutliner is a component within a larger framework, currently under development, that will provide an intelligent shared workspace for collaboration in architectural design. The workspace will allow users to synchronously and asynchronously create tasks, proposals, and artifacts. Mechanisms will be embedded that allow evaluation of design alternatives.

References

- Apple Computer Inc. (1998), WebObjects Developer's Guide
- Jabi, W. (1998); The Role of Artifacts in Collaborative Design; Proceedings of the Third Conference on Computer-Aided Architectural Design Research in Asia, Osaka, Japan: CAADRIA, Department of Architecture and Civil Engineering, Kumamoto University, (pp. 271-280)
- Jabi, W. (1996); An Outline of the Requirements for a Computer-Supported Collaborative Design System, Open House International, Vol. 21, No 1 (pp. 22-30)
- Jabi, W. (1996); "Domain-Specific Tools for Collaboration in Architectural Design", *Proceedings of the 3rd International Conference on Design and Decision Support Systems in Architecture and Urban Planning*, Technical University of Eindhoven, Spa, Belgium, 1996, pp. 248-259.
- Nardi, B. (1994); Collaborative Multimedia: Getting Beyond the Obvious, Proceedings of The Second ACM Conference on Multimedia, New York: ACM (pp. 119-120)