WebOutliner: A Web-Based Tool for Collaborative Space Programming and Design

Wassim Jabi, University at Buffalo, The State University of New York, USA

Abstract

This paper discusses a web-based tool that allows members of a design team to collaboratively specify a hierarchical spatial program for an architectural project. 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. Given that these components are hierarchical allows users to filter information, analyze and compare design parameters and aggregate hierarchical amounts in real-time. Furthermore, the software goes beyond outlining functions to support synchronous collaborative design by linking each item in the spatial program to a detail page that allows file uploading, real-time group marking of images, and textual chat. Thus, the software offers a seamless transition from the largely asynchronous definition of an architectural program to synchronous collaboration. In addition, and in contrast to commercially available groupware, the software allows multiple collaboration sessions to run at the same time. These sessions are artifact-based in the sense that they get automatically initiated once participants visit the same architectural space in the program hierarchy. The software employs a three-tier object-oriented, web-based scheme for a richer representation of hierarchical artifacts coupled with a relational database for server-side storage. The prototype integrates this technology with Java-based tools for synchronous web-based collaboration.

1 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. 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, 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.

2 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 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.

A second breed of web-based collaboration technologies is based on the idea of a shared virtual workspace (Takemura and Kishino 1992). 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 webex.com offers for free, allow 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. 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.

3 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 note here that I am not arguing for a replacement of current collaboration software, but simply the need to augment them.

4 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, and (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.

5 WebOutliner

Fundamentally, WebOutliner is a web-based application that represents and manipulates hierarchical elements using Apple’s WebObjects technology (Wilson and Ostrem 1998). WebOutliner is built using specifications and earlier research found in (Jabi and Hall 1995a; Jabi and Hall 1995b; Jabi 1996a; Jabi 1996b; Jabi 1998; and Jabi 1999). WebOutliner is composed of reusable components that can be adapted for several needs. In its current implementation, WebOutliner manages the components of a simple building program. Following an object-oriented approach, WebOutliner’s components belong to one of three categories: (1) Model, (2) View, and (3) Controller (Krasner and Pope 1988). 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 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.

5.1 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). 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 6). 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 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).

5.2 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 (Figure 11). It is in this simple move that a user transitions to synchronous collaboration. If another user happens to visit the same item at the same time (or they have scheduled a meeting in that space ahead of 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 12).
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 conference room. At the same time, Bob is adding more spaces to Level 1.

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

6 Conclusion

This paper illustrates the need for domain-specific collaborative software. Integrating a database, an object-oriented application server, and a web-based interface can prove to be a powerful and flexible solution for customizing behavior and adding intelligence to shared artifacts. Obviously, user testing is needed to verify these claims. Future research will concentrate on such testing and will be reported. The specification of a building program is used here only as a simplified example of a domain-specific functionality that cannot be found in any of the commercial collaborative 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 a dynamic shared workspace for collaboration in architectural design. The workspace will manage design workflow by allowing users to synchronously and asynchronously create tasks, proposals, and artifacts. Mechanisms will be embedded that allow evaluation and selection of design alternatives as well as the capture of design rationale.

Acknowledgements

I wish to thank Professors Harold Borkin, James Turner, Judy Olson and Edmund Durfee for supervising the doctoral research project of which WebOutliner is a part. I also wish to thank Theodore Hall who provided valuable feedback on earlier experiments with collaborative software and co-authored with me two papers on the subject. Apple and WebObjects are trademarks of Apple Computer, Inc. Microsoft and NetMeeting are trademarks of Microsoft Corporation. CU-SeeMe is a trademark of CUseeMe Networks, Inc. Webex.com is a trademark of Webex Communications, Inc. Groupboard is a trademark of Groupboard.com.
ACADIA 2000: Eternity, Infinity and Virtuality

Figure 6. Toolbar and Commands.

Figure 7. Expanded Building Program

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

Figure 9. Editing Name and Area of an Element.

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


