Basic Support for Cooperative Work on the World Wide Web

R. Bentley, W. Appelt, U. Busbach, E. Hinrichs, D. Kerr, K. Sikkel, J. Trevor, G. Woetzel
CSCW Group, Institute for Applied Information Technology (GMD FIT)
German National Research Centre for Information Technology
Schloß Birlinghoven
D-53754 Sankt Augustin
Germany
email: Richard.Bentley@gmd.de

Abstract

The emergence and widespread adoption of the World Wide Web offers a great deal of potential in supporting cross-platform cooperative work within widely-dispersed working groups. The Basic Support for Cooperative Work (BSCW) project at GMD is attempting to realise this potential through development of Web-based tools which provide cross-platform collaboration services to groups using existing Web technologies. This paper describes one of these tools, the BSCW Shared Workspace system—a centralised cooperative application integrated with an unmodified Web server and accessible from standard Web browsers. The BSCW system supports cooperation through 'shared workspaces'; small repositories in which users can upload documents, hold threaded discussions, and obtain information on the previous activities of other users to coordinate their own work. The current version of the system is described in detail, including design choices resulting from use of the Web as a cooperation platform and feedback from users following the release of a previous version of BSCW to the public domain.

Keywords

World Wide Web, Computer Supported Cooperative Work, Groupware, Information sharing

1. Introduction

The explosive growth of the World Wide Web and its penetration into academic, commercial and domestic environments is well documented. The combination of a global addressing system, network protocol, document mark-up language and client-server architecture provides for a simple method for users to search, browse and retrieve information as well as share information of their own with others. However, while clearly powerful and useful, this approach does not directly support more collaborative forms of information sharing, where widely-dispersed working groups work together to jointly author, comment and annotate documents, and engage in other forms of collaboration such as group discussions.

There are a number of reasons to suggest that support for such collaborative working based on information sharing is becoming more necessary. Trends in the current business world towards decentralisation, joint ventures, outsourcing of business functions and so on are highlighting a need for effective methods of sharing information and coordinating activities. Although applications such as video conferencing may address some of the problems, empirical evidence suggests that systems which provide access to (shared) information, at any time and place and using minimal technical infrastructure, are the main requirement of groups

collaborating in a decentralised working environment (Rao 1995, Gorton, Hawryszkiewycz and Fung 1996).

Although existing groupware technologies such as Lotus Notes often provide such services within organisations, problems arise when organisational boundaries must be crossed, and issues of integration and interoperability with different computing platforms, databases and other application software must be addressed. In this context the Web has a number of distinct advantages as the basis for tools to support collaborative information sharing:

- Web client programs (browsers) are available for all popular computing platforms and operating systems, providing access to information in a platform independent manner,
- browsers offer a simple user interface and consistent information presentation across these platforms,
- data formats which can not be presented directly by the browser can easily be delegated to external 'helper' applications for further processing,
- Web browsers are already part of the computing environment in an increasing number of organisations, requiring no additional installation or maintenance of software for users to cooperate using the Web,
- many organisations have also installed their own Web servers as part of an Internet presence or a corporate Intranet and have familiarity with server maintenance and, in many cases, server extension through programming the server API.

Given these characteristics, the extension of the Web to provide richer forms of cooperation support for working groups is both appropriate and desirable. This issue is the focus of the Basic Support for Cooperative Work project (BSCW) at GMD, which as its main goal seeks to transform the Web from a primarily passive information repository to an active cooperation tool. The basis for this work is the BSCW Shared Workspace system, an application which extends the browsing and information download features of the Web with more sophisticated features for document upload, version management, member and group administration and more, to provide a set of features for more collaborative information sharing accessible using unmodified Web browsers, and therefore across different platform and network infrastructures. This paper focuses on the design and implementation of the current version of the BSCW system, BSCW2, which has been extensively re-designed based on user feedback from the release of a previous version of the system (Bentley, Horstmann, Sikkel and Trevor 1995) to the public domain[1].

2. The World Wide Web as the basis for cooperation tools

From its inception the Web was intended as a tool to support a richer, more active form of information sharing than is currently the case. For example, from its earliest drafts the HTTP protocol specification included features allowing users to upload documents to a Web server as well as the current model of downloading information and completing HTML forms. Additional features were also described specifically to address the problems of collaborative working, such as the 'check-in' and 'check-out' methods of locking documents for editing to ensure consistency in a multi-user environment. To date these aspects of the Web have, perhaps temporarily, been

---

[1.] All BSCW software described in this paper is freely available for downloading; to obtain the software, use our public BSCW server at GMD, or for more information on the project see http://bscw.gmd.de

sidelined while development of Web browsers, servers and standards has focused largely on the more passive aspects of information sharing such as browsing and publication.

Despite the lack of direct support for collaboration, the current Web protocols and standards do hide much of the complexity of deploying applications in a distributed, heterogeneous environment. The most common method of doing this is to extend a standard Web server via the Common Gateway Interface (CGI)—the standard server API—with new application functionality or 'glue' code to an existing application, presenting the 'interface' to the application as a series of HTML pages which can be displayed by a standard Web browser. With this method, developers can take advantage of the existing base of browsers as client programs for their applications, and the other advantages listed above, but must also accept the constraints of the basic Web client-server architecture, protocol and the current limitations of the browsers themselves. These constraints inhibit the deployment of some kinds of applications, particularly those requiring:

- *continuous media*: HTTP does not directly support the specification of a guaranteed transmission rate between server and client. In general, data transfer results in bursty transmissions which are unsuitable for (real-time) continuous media like audio and video,
- *information replication*: The architecture provides no direct support for information replication, which may be required for disconnected working, rapid feedback or reliability,
- *peer-peer communication*: There is no support for server-server, (server initiated) server-client or client-client communication, problematic for applications where the server needs to play an active role (e.g. to notify users of changes to information or maintain information consistency over several servers),
- *rich user interfaces*: Although HTML supports features such as simple form-filling widgets, it is not a user interface toolkit and the interaction styles which can be supported are very limited.

In its current form the Web is more suited to centralised cooperation tools which have no strong requirements for highly-interactive user interfaces, rapid feedback and 'feedthrough' (user interface updates in response to others' interactions) or a high degree of synchronous notification. It is therefore unsurprising that much of the work in this area has focused on more 'asynchronous' (i.e. distributed in time) forms of cooperation support; to do otherwise requires more extensive innovation and development of proprietary mechanisms which often do not run across all platforms and require more complex installation of additional client and server software. Examples in this latter category include the Contact system for collaborative authoring (Kirby and Rodden 1995) and the work described by (Frivold, Lang and Fong 1994), which extend the Web with additional client and server mechanisms for synchronous notification.

For systems which provide asynchronous cooperation support directly within the frame of the Web architecture, the area of text-based conferencing is the most active. Systems such as HyperNews[2] extend Web servers with features for managing threaded group discussions, similar to Usenet newsgroups. Articles and replies are created using HTML forms, and the system handles article storage and generation of HTML pages which display the structure of the discussion threads and the articles themselves. Such systems often offer more functionality than

---

[2]. http://union.ncsa.uiuc.edu/HyperNews/get/hypernews.html

Usenet newsgroups, such as archiving and password-protection, but these features are also part of equivalent, non Web-based commercial tools which tend to offer richer user interfaces than is possible using standard Web browsers and HTML. The main advantage of Web-based conferencing tools is their deployment using standard browsers and the ability to embed hypertext links in articles to other information, held on different servers, in FTP archives and so on. As integrated news-readers and browsers mature (such as the Netscape Navigator Usenet client) this advantage may disappear.

To take advantage of the huge growth of the Web a number of companies are trying to re-package existing client-server groupware products as 'Web-enabled' applications. These include workflow systems such as ActionWorkflow Metro from Action Technologies[3] and shared database and messaging applications like IBM Lotus's Domino extension to their Notes product[4]. Only a subset of the functionality can be accessed from standard Web browsers, however, and these tools require considerable investment to install and customise to the needs of the organisation. For many collaboration tasks, a more lightweight solution is desirable.

Another recent trend in this area has been a move towards Web-based remote filesharing applications. Systems like Apple's NetFinder[5] for the Macintosh and Microsoft's WebPost SDK[6] for the PC offer the document uploading and downloading functionalities of FTP, but integrate this with the Web architecture to support filesharing via a Web server. A consequence is the ability to manage documents on a remote Web server without having access to the server machine, easing the problems of Web site administration. However such tools use proprietary protocols and as a result do not support cross-platform filesharing. They also offer little extra functionality over that supported by more advanced FTP clients, and provide no direct support for collaborative sharing such as consistency control mechanisms. Studies from the field of Computer Supported Cooperative Work (CSCW) also highlight the importance of 'awareness' of the activities of other users as being fundamental to the effective coordination of group work, and a central component of any system to support collaboration (e.g. Dourish and Bellotti 1992, Fuchs, Pankoke-Babatz and Prinz 1995). Currently, Web-based remote filesharing applications provide no support for such group awareness.

An area of Web development which is beginning to look at this issue of group awareness is the Virtual Reality Modeling Language (VRML). VRML (Ames, Nadeau and Moreland 1996) is a language for describing 3-D worlds, rather like HTML describes 2-D pages. Like HTML, however, VRML has grown to include extra features such as specification of behaviour of VRML objects to animate the virtual world. Currently researchers are working to extend VRML even further to allow multiple users to explore the same world, and to embody users' positions in the world with representations such as 'avatars' (see for example the work of Lea, Honda and Matsuda 1996, Broll 1996). It is clear that this approach holds a potential for future collaborative work on the Web but few browsers are currently available which support

---

[3.] http://www.actiontech.com/metrotour/

[4.] http://www.internet.ibm.com/domino.htm

[5.] http://cybertech.apple.com/AppleNetFinder.html

[6.] http://www.microsoft.com

multi-user VRML worlds, and major difficulties must be overcome before these systems offer real support for collaboration.

Each of the applications discussed above has some potential to transform the Web from a passive information repository to a more active cooperation tool. The BSCW Shared Workspace system combines aspects of each of these approaches to offer an integrated cooperation service, accessible within the constraints of the current Web architecture from standard Web browsers. The system provides a set of basic mechanisms for information sharing, and integrates the remote file sharing aspects of tools like NetFinder with a model of shared workspaces from groupware systems like Notes. A shared workspace, accessible to members of a group for coordinating and organising their work, provides facilities such as threaded discussions like those supported by HyperNews as well as tools for document management, versioning and the like. To provide users with a degree of group awareness, all the core facilities are integrated with a simple event service which provides details on what has been done within a workspace, when, and by whom.

The novelty of the BSCW system lies in the provision of basic features for cooperation in an integrated service, accessible from different computing platforms and making no demands on users to adopt new word processing, spreadsheet or other application software. It is our belief that this combination of a broad but lightweight set of closely-integrated mechanisms, based around the notion of sharing information, is very important for widely-dispersed working groups which may cross organisational and national boundaries, operate in heterogeneous environments, and engage in tasks with very different requirements for support.

3. Sharing information with the BSCW Shared Workspace system

The BSCW Shared Workspace system is an extension of a standard Web server through the server CGI Application Programming Interface. A 'BSCW server' (Web server with the BSCW extension) manages a number of shared workspaces; repositories for shared information, accessible to members of a group using a simple user name and password scheme. In general a BSCW server will manage workspaces for different groups, and users may be members of several workspaces (e.g. one workspace corresponding to each project a user is involved with).

A shared workspace can contain different kinds of information such as documents, pictures, URL links to other Web pages or FTP sites, threaded discussions, member contact information and more. The contents of each workspace are represented as information objects arranged in a folder hierarchy. Members can transfer (upload) information from their machines to the workspace and set access rights to control the visibility of this information or the operations which can be performed for others. In addition, members can download, modify and request more details on the information objects by clicking on buttons—HTML links that request workspace operations from the BSCW server, which then returns a modified HTML page to the browser showing the new state of the workspace.

Figure 1 shows a sample workspace view. The workspace contains a number of different kinds of documents, a URL link to another Web page ("BSCW project page"), an 'article' object representing an item for discussion ('Features of 2.0'), a sub folder containing further workspace information and a document under version control ('Project publications'). The last 'significant' operation performed on each object is presented, and a list of clickable 'event icons' give information on other recent changes (Section 3.2). A set of operations which can be performed

on each object is given below the object icon and name, and a description is also presented if one has been supplied (as with the GIF image 'BSCW icon').

Figure 1. HTML user interface to the BSCW Shared Workspace system

Access to workspace functions is provided by the buttons at the very top of the page as well as the text HTML anchors below each object. The former operate on the current folder being shown, so that 'add URL' will return a HTML form for specifying the name and URL of a URL link object to be added to the current folder, while the latter perform operations on the individual objects, such as 'rename', 'edit description' and so on. As a short cut, the checkboxes to the left of each object in combination with the buttons above or below the list of objects allow operations on multiple object selections.

Clicking on an object name will perform different operations depending on the type of the object; clicking on a document will download it, possibly for display by the browser (for HTML files or GIF images for example), display by an external application (as with a Microsoft Word document) or saving to disk, while clicking on a folder 'opens' the folder and replaces the current view with a display of the folder contents. This last method of navigating 'down' through

a folder hierarchy is supplemented by a navigation bar at the top of the page presenting the current location and path; clicking on the first element of the path (":bentley" in Figure 1) returns to the current user's 'home folder', which lists all the workspaces of which the user is a member, and therefore has access to.

3.1. Storing documents in a shared workspace

An important aspect of the BSCW system with respect to current Web technology is the ability to upload documents to the central server where they can then be accessed by others. As mentioned in Section 2, direct support for document uploading was originally part of the HTTP protocol (the HTTP 'PUT' method) but is generally not implemented by current Web browsers or servers. At the time of writing only the more recent versions of the Netscape Navigator browser support any form of document uploading. Netscape's method is based on a draft proposal for uploading files using the method for sending HTML form data (the HTTP 'POST' method)[7]. This method is not yet supported by most servers, but the BSCW system implements it directly.

For a Netscape browser, clicking on the 'add document' button shown in Figure 1 returns a HTML form allowing users to select a document on the local machine to upload and specify the document type and the name to give the document in the workspace. However, not everyone uses the Netscape browser, and those that do may use older versions that do not provide the uploading feature. In addition the Netscape method has a number of deficiencies including a lack of progress reporting; there is no way to distinguish a server crash from a partially-complete upload.

To address these problems, and support document uploading for users of browsers other than the latest versions of Netscape, we have developed a small 'helper' application to augment a standard Web browser. This helper, versions of which have been developed for Macintosh, PC and Unix platforms, uses the same protocol as Netscape to transmit documents to the BSCW system, but provides a much richer user interface as shown in Figure 2.

The helper supports selection of multiple documents for upload, automatic detection of document type, and full feedback on the progress of the document transmission to the BSCW server. (The Windows'95/NT version also supports drag and drop of multiple documents from the desktop to an open Web browser.) Users can tailor their preferences profile (Section 3.6) to use this method of uploading rather than the Netscape method, so that clicking on the 'add document' button automatically launches the helper application.

A further innovation of the BSCW system concerns the handling of document 'type' information. Current Web browsers can be configured to automatically launch specific applications when documents of the correct type are downloaded from a Web server, and most servers 'guess' this type from the name of the document; documents ending in '.gif' might be treated as GIF format image data, for example. This method is clearly problematic when documents must be shared between users and platforms with different conventions for document naming.

The BSCW system does not use the document name to derive its type. Rather, the system stores the type (and other information about the document, such as its size, creator and so on) in an object database managed by the BSCW server. This type information is then sent to the

---

[7]   ftp://ds.internic.net/rfc/rfc1867.txt

browser when the document is requested, and is also used to select an appropriate icon for the document in the workspace listing (Figure 1). The helpers automatically select document types based on a local configuration file, which can be tailored by the user to reflect personal naming conventions and extend the range of types automatically recognised. The user can also set the type explicitly, and even change the type after the document has been uploaded to the server (though this should rarely be required).
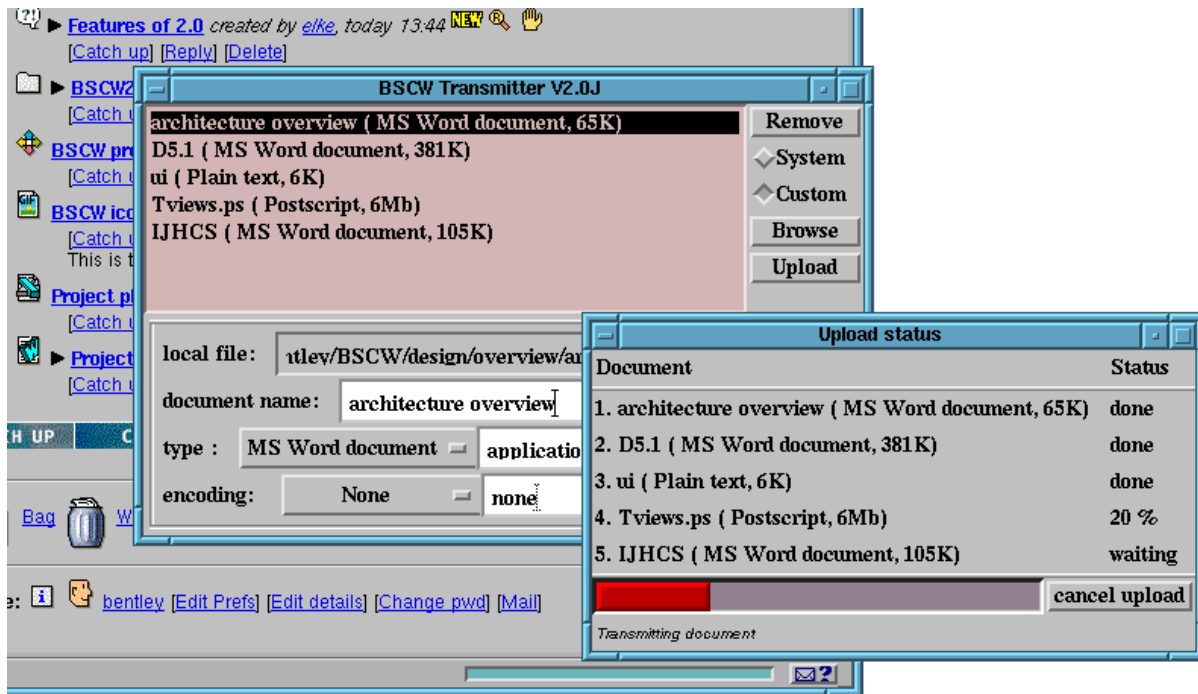

Figure 2. Uploading multiple documents to a shared workspace with the helper application

3.2. Events and activity awareness

The support provided by the BSCW system for document uploading reflects the emphasis on information sharing within heterogeneous environments, but on its own does not transform the Web from a passive information repository to an active tool for cooperation. As described above, a cooperative system should provide awareness information to allow users to coordinate their work. The event service of the BSCW system is an attempt to provide users with information on the activities of other users, with respect to the objects within a shared workspace.

Events are triggered whenever a user performs an action in a workspace, such as uploading a new document, downloading ('reading') an existing document, renaming a document and so on. The system records the events, and presents the recent events to each user as event icons in the workspace listing (Figure 1). 'Recent' in this context means events which have occurred for an object since the user last 'caught up'; an operation by which users can tell the system they are aware of the events that have occurred so far and no longer wish to see them in the workspace (rather like catching up articles in a Usenet newsgroup). Events can be caught up at different levels, from individual objects to complete workspace folder hierarchies.

The system distinguishes five types of events in the workspace listing:

- new events show an object has been created since the user last caught up,
- read events (shown by a spyglass) show an object has been downloaded/read by someone,
- change events (shown by a pen) show an object has been modified; this category includes several event types, such as 'edited', 'renamed', 'changed description' and so on,
- move events show an object has changed location, and includes 'deleted' and 'undeleted' events (showing the object has been moved to/from a wastebasket) and 'cut' and 'dropped' events (showing the object has been cut to/dropped from a user's personal bag—Section 3.3),
- touch events, represented by a hand icon, are displayed for a container such as a folder to show that something has happened to an object contained inside (either directly or lower down in the folder hierarchy).
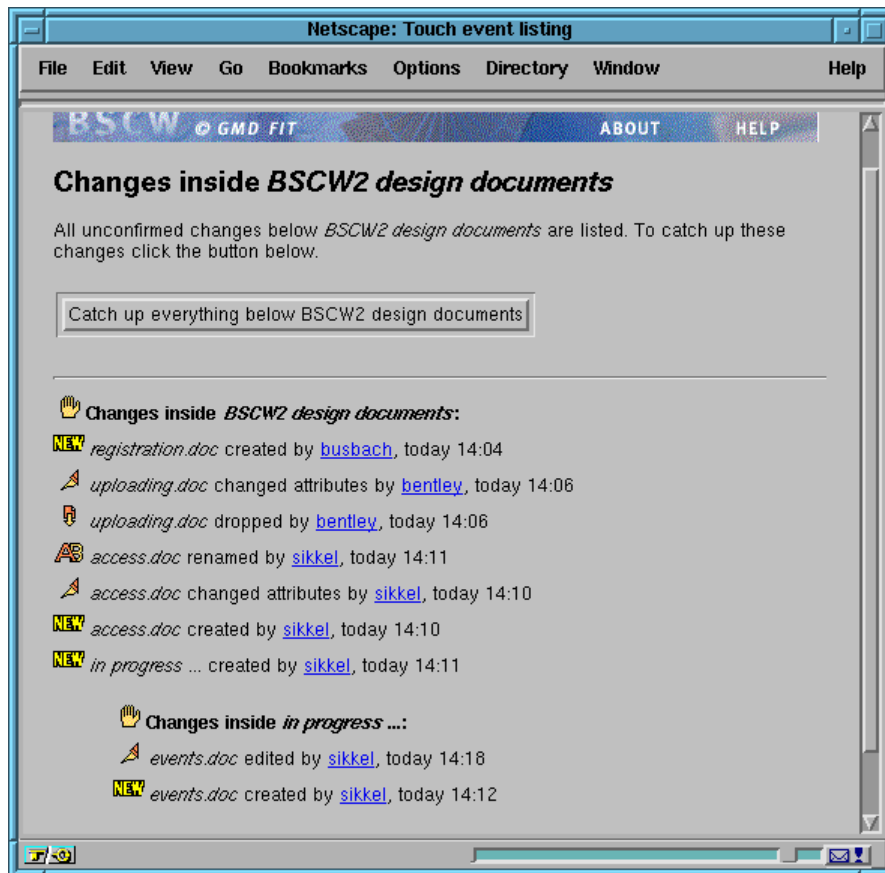

Figure 3. Displaying the 'touch' events for a workspace folder

Clicking on an event icon displays a list of all events within that category which have occurred for an object since the user's last catch up. Figure 3 shows this for the touch events of the 'BSCW2 design documents' folder, which displays the events that have occurred within this and lower folders in the hierarchy. Each event entry describes what was done, when and by whom. Although this approach for providing group awareness is very simple, feedback from users of the BSCW system indicates that information such as 'A uploaded a new version of document X', or 'B has read document Y' is often very useful for group members in coordinating their work and gaining an overview of what has happened since they last logged in. However, this feedback has also revealed the occasional need for more active notification of events, and we are currently

extending this event mechanism for email notification of workspace changes based on registered 'user interests'.

3.3. The degree of sharing

The shared workspace metaphor supported by the BSCW system allows members of a workspace to store information to make it available to other workspace members. Without any further mechanisms, this implies that everyone who is a member of the workspace has the same capabilities for manipulating this information, including editing and deleting documents. Where workspaces and member groups are relatively small this approach may be suitable, but experience from user feedback from the previous version of the system revealed that for some tasks there is a need to both increase and decrease the degree of sharing for workspace information.

The need to restrict the level of sharing is somewhat obvious; rather than create multiple workspaces with different memberships, users of the previous version of BSCW would often create a single, large workspace and want to structure this using sub-folders with different access restrictions. However, the need to open up access to a workspace to non-members is perhaps less obvious, and results from the desire of many users who wish to use BSCW as a tool for managing a standard Web server. It has become clear that in many organisations, Web server management is often a group rather than an individual responsibility; the user interface and document management features of BSCW, the event service for group awareness, and the ability to manage the documents remotely all led users to see BSCW as a suitable tool for Web site administration.

This and other scenarios of use have driven the development of the access control model for BSCW2. It is clear that for many users and groups the notion of a shared space where all information is accessible and can be modified by all members is suitable for their collaboration, and this is therefore the default when adding documents or other information to a workspace. The 'owner' of an object can however override this; clicking on the object's 'info button' in the workspace listing (Figure 1) returns a page of detailed information about the object, including its current access configuration, and access to further operations which allow this configuration to be tailored (Figure 4).

The access model is currently simplistic, but does allow fine-grained control over the visibility of objects and the capabilities of different groups of users. Several groups are pre-defined and system maintained, such as 'workspace members', while others are user-tailorable such as the 'owners' group. Access to objects for non-registered users of the server is provided by allowing the 'anonymous' group to perform operations. To publish a document a user need only allow the anonymous group to 'get' the document, and anyone can then download the document given the document URL without being asked for a user name and password. We are currently extending this so that users can define their own groups of registered users of the server for access control.

The per-object access control model allows tailoring of the degree of information sharing within the group of workspace members. Each user also has a private space, called the personal 'bag', in which to store and manipulate information. The bag is accessed by clicking on the bag icon at the bottom of a workspace listing (Figure 1), and is 'carried' between workspaces so the objects it contains are always accessible.
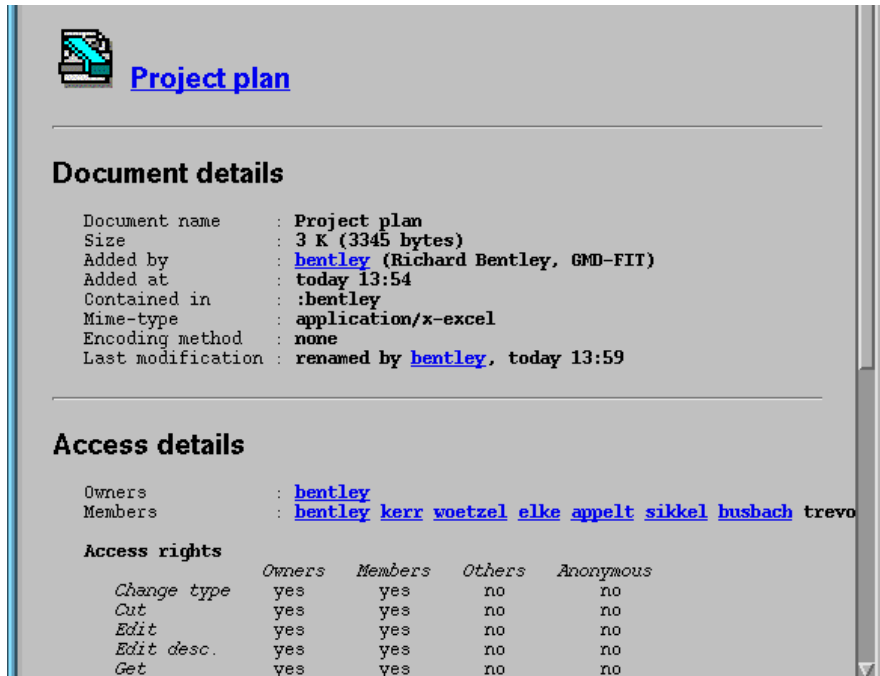
Figure 4. Obtaining extra information on a shared document

The bag is a useful construct for structuring information on a BSCW server. Users can upload documents directly to their personal bags, and then 'drop' (subsets of) these documents to different workspaces or workspace folders. Objects can also be 'cut' to the bag from a workspace folder, and the bag records the targets of the last 'cut' operation so that users can navigate to a different location and 'drop' these objects without opening the bag itself, rather like the cut and paste mechanism of a clipboard. This use of the bag as a temporary location is necessary, as the more familiar drag and drop style of moving objects between locations is not possible within the capabilities of current Web browsers.

3.4. Member administration

Access to information stored in a shared workspace can be provided for any user with a Web browser through the anonymous group mechanism. By default however, access is restricted to users who possess a valid login consisting of a registered user name and password. In addition to providing a simple level of security, this identification mechanism is required by the event and access control services which use the identity of the current user to record event information and verify access capabilities respectively. The system uses the 'basic authentication' method supported by standard Web browsers and servers to obtain the identity of the requestor. This requires each user to type their password at the start of their session with BSCW, but thereafter request authentication is handled automatically by the Web browser, transparently to the user.

New members are added to a workspace through an 'invitation' from an existing workspace member. Invited users may already be members of other workspaces, and therefore have a login for the server. Alternatively, users might not have a login and therefore must register a user name and password with the system. In the latter situation, an invitation to join the workspace is sent to the user via email, and included in this email is a unique URL (containing

an encrypted 'token') which the invited user can send to the BSCW system to access a registration page and give themselves a user name and an initial password. Thus only the recipient of this email can register with the system and become a member of the workspace.

For some servers it may also be appropriate to allow users to register themselves and create a login, without being invited by an existing member. The system therefore supports 'self-registration', where users can access a page and give their email address, to which the registration URL is then sent. Once registered, the user will be able to create workspaces of their own and invite other members, but will not be a member of any existing workspaces. This mechanism can be easily disabled if self-registration should be prohibited for a particular BSCW server.

Figure 5. Members of the workspace 'BSCW project'

Figure 5 shows the members of the 'BSCW project' workspace. Here a user has been invited to join the workspace ('trevor@gmd.de'), but has not yet registered a user name and password. The system creates a 'pending' registration, which must be converted to a 'full' registration by the user before they can access the workspace. Using this mechanism, the system ensures that a valid email address is provided for each registered user, as without this the user will not receive the URL for upgrading the pending to a full registration. Having a valid email address is important

for direct emailing from the workspace using the 'mail' operation of each user object (Figure 5), and also for the active email notification of event information, discussed in Section 3.2.

For each registered user of the server the system maintains a personal 'address book' containing the user names and email addresses of other users. Users can invite new members to the workspace by selecting their entries in the address book and can add, remove and edit these entries as required. The address book is also used to define 'personal groups', which can be used in specifying access properties of workspace objects (Section 3.2). The group definitions are stored in the address book, and can be selected here for other operations such as group mailing.

Like the document object shown in Figure 4, clicking the info button for a workspace member will also return a page of user information such as organisational details, phone number and so on. This page is also directly accessible from entries in the workspace and event listings. As a shortcut, users can edit these personal details they provide to other users via the quick access bar at the bottom of the workspace listing (Figure 1 and Figure 5).

3.5. Further collaboration services

The primary goal of the BSCW project is to construct a platform which provides basic features for supporting cooperative work for widely-dispersed working groups, independent of their computing, network and application infrastructures. Designing a useful and stable system has focused much of our attention to date on the features for simple document management, however BSCW was never intended as just a Web-based filesystem like NetFinder; the system provides a number of services intended to support different collaboration activities.

For many applications collaboration within a group will involve some form of joint document production. The system therefore provides basic support for version management. Clicking a document's 'version' button places the document under version control, so that new versions of the document can be added to a linear version history without destroying previous versions. The document 'Project publications' in Figure 1 has been placed under version control, but the current version, version 2, is accessible by clicking the document name. Access to previous versions is provided by clicking on the '(versions)' link in the document entry.

The system also provides facilities to assist with collaborative editing. It is possible to add a 'note' to a version-controlled document which will be displayed whenever a user attempts to download or perform an operation on the document, or when the user clicks on the note icon in the document's entry in the workspace listing. In Figure 1 the document 'Project publications' has an attached note. The note can be used (amongst other things) to indicate that the document is currently being edited, and is a form of 'soft-lock' (Bäcker and Busbach 1996) which provides awareness of the current situation but does not enforce a stricter model of locking. It is our experience that where users are trying to coordinate their work in a cooperative setting, this mechanism is usually adequate to ensure conflicts do not occur. For other domains such as software development consistency-control based on soft locking may be much less appropriate.

A further service offered by the BSCW system is support for threaded text-based conferencing. 'Article' objects are simple messages which can be added to a workspace folder, appearing in the folder listing in the same way as a document, folder or URL link. The goal here is not to provide all the features common to sophisticated text-conferencing systems like HyperNews, but to offer basic functionality integrated with the shared workspace. Figure 6 shows a threaded discussion in the workspace listing and the user interface to an article object.
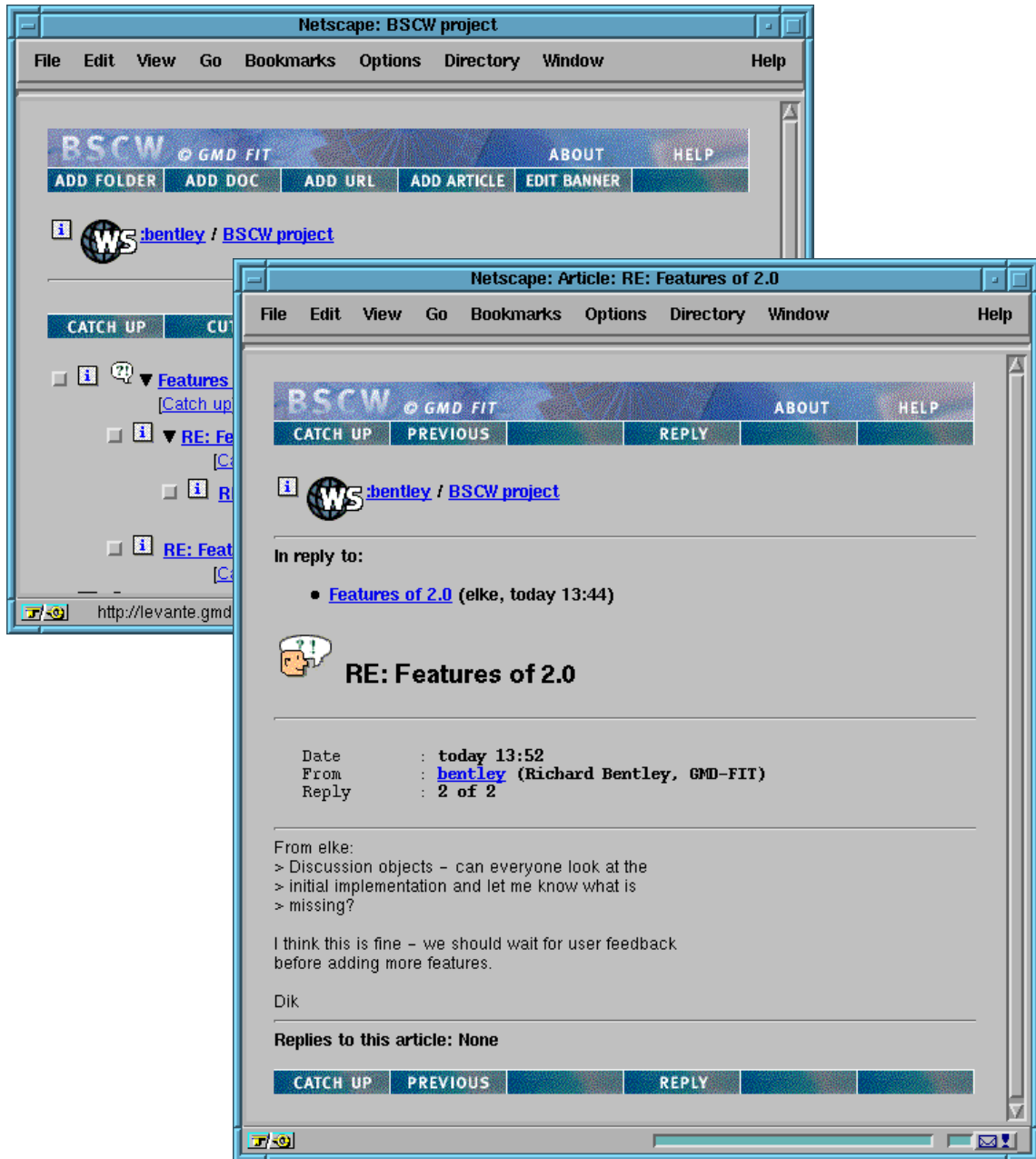
Figure 6. Articles in the workspace listing and display of an article object

The BSCW system therefore offers a range of collaboration services, integrated in a single shared workspace framework and accessible using a lightweight technical infrastructure. Section 5 shows how we are expanding this range of services beyond asynchronous, text-based mechanisms while keeping to our goal of supporting cooperation independent of platform, network and application infrastructure.

3.6. User interface considerations

None of the user interface mechanisms for the shared workspace as described above rely on special features of specific Web browsers (such as Netscape frames), but conform strictly to the

HTML 2.0 specification. As many newer Web browsers now support more advanced features as defined in more recent drafts of the HTML specification, we have set our baseline relatively low. Our experiences from user feedback following release of the previous version of the BSCW system however (Bentley and Appelt 1997) suggest that many users are still using older versions of Web browsers that do not support the more advanced HTML features. Analysis of the access logs for our public BSCW server, which record details of each accessor's browser version and type, confirms this to be the case. As described in Section 5, we are looking at methods of improving the user interface using more advanced features supported by the more popular Web browsers, but a strict requirement is that all functionality should be accessible using the baseline Web browser.

This design decision is intended to make the system accessible for a broad user population, but this alone is insufficient; a rich user interface designed for a large screen workstation on a local network is not so useful for the majority of the on-line community who currently use small screen PCs over a modem link (Berghel 1996). A view which transfers in seconds and is completely displayed in the former case may take much longer and require a great deal of scrolling in the latter. We have therefore provided methods for users to customise the user interface to reflect their individual environments, requirements of their tasks and their individual preferences.

Using the (A)ctions and (D)escriptions buttons (Figure 1) users can 'fold in' the actions and/or descriptions underneath each object in the workspace listing so they are not displayed. This considerably reduces the screen area required to list a folder and reduces the time for a listing to transfer. The fold 'level' can be set on a per-folder basis, and the settings are recorded in the user's preferences profile, which can be edited directly using the 'edit prefs' option (Figure 5).

We have extended this support for per-user preferences to other aspects of the interface presentation. For example, it is common when using the Web at times of high loading or over a slow connection to suppress image transfer, greatly reducing the amount of information returned by the server for each page. With pages containing many icons, however, the resulting presentation is often confusing, and as different Web browsers represent the omitted graphics in different ways it is hard to design an effective interface. With the BSCW system the user can download and save all the icons locally, on a local disk or (better if more than one user at a site uses the BSCW server) on a local Web server, and thus only the text of the HTML page needs to be transferred with each request. Similarly, users can also select their preferred language for the user interface, so that different users can receive dialogues and listings from the same workspace in different languages. English, German and French are currently supported, but extension to other languages is straightforward[8].

In the Web community some emphasis is being placed on extending browsers, servers and protocols to better support alternative presentations. It is now common, for example, for browsers to send as part of each request a string identifying their type, version, enhancements etc., and some servers and applications use this information to customise the response based on

---

[8]. All user interface messages are stored as plain HTML files in a language directory. Adding support for a different language requires creating a new language directory with translated versions of these HTML files. This arrangement also allows interface customisation for other purposes, e.g. to support organisational style guides or different user interfaces for expert and novice users.

records of the capabilities of each browser. These aspects (and more advanced modifications as proposed for the HTTP protocol such as 'content-negotiation') may address some aspects of the need to customise presentations to the user environment, but are unlikely to address requirements of users' current tasks. We believe this is best supported by allowing users to tailor their interfaces to present information in an appropriate manner, and look to extend the support for per-user preferences in future versions of BSCW.

4. Implementation of the BSCW Shared Workspace system

The BSCW system provides a modular extension of the World Wide Web's client-server architecture without requiring modification to Web clients, servers or protocols. The core is a standard Web server extended with the BSCW software through the standard server API, or Common Gateway Interface (CGI). The system is written entirely in the interpreted programming language Python[9], and the only additional software required to use the system besides a Web server is the Python interpreter. It currently runs on most Unix platforms, including the public domain version Linux, and we are in the process of porting to Windows NT. This section gives a brief overview of the system's implementation.

Standard Web servers can be configured to delegate certain requests to external, developer-supplied code components through the CGI API rather than handle them directly. The extension code must parse the request and compute a response, which is then returned by the Web server to the requesting browser. The BSCW system uses this mechanism to route BSCW requests to the appropriate components externally to and independently of the Web server itself.

At the highest level of abstraction the system can be decomposed into three layers which deal with request handling, operation handling and persistent object storage (Figure 7). In the request handling layer the details of the request are formatted as an internal representation called a request object, which is an abstraction over the method and protocol used to transmit the request. Thus, although BSCW2 currently supports only Web-based access, this approach allows consideration of other forms of access. We have for example prototyped an email interface to the system.
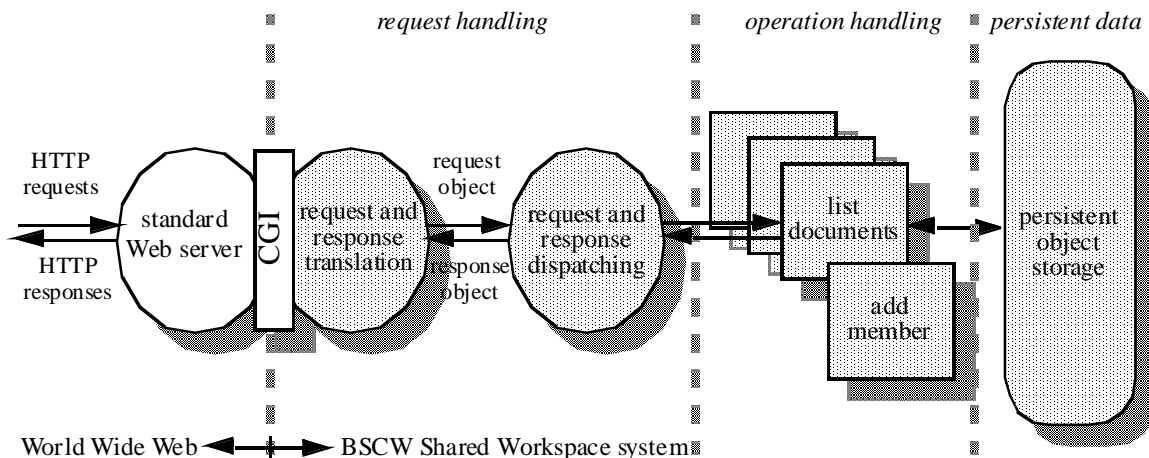

Figure 7. Structure of the BSCW system

---

[9]. http://www.python.org/

The request object is dispatched to the operation handler which implements the BSCW functionality requested, such as listing the contents of a workspace, adding a new member and so on. The operation handlers interact with the persistent object store to process the request, creating, deleting and modifying objects as necessary, before generating a response object to return to the requestor. The response object is returned to the request handling layer for translation into a concrete format suitable for the access method employed. (So for BSCW2 the response format will be a HTML page which can be displayed by a standard Web browser).

This layered architecture therefore allows extension of BSCW in a number of different ways. Besides new request handlers for different methods of access, new operation handlers can be added to provide new functionality or as 'wrappers' around an existing application. It is also straightforward to access the persistent store to store new kinds of objects without modifying the storage routines themselves. This extensibility allows new services to be integrated with BSCW in a straightforward manner, and the choice of the interpreted, dynamically-typed language Python as the implementation language directly supports rapid prototyping. Some of the extensions we have prototyped or are planning to implement are discussed in the following section.

5. Current status and future developments

Version 2 of the BSCW system was released to the public domain in June 1996. All the code for the server and the helper applications is available and offered free of charge for non-commercial use. In addition, we have installed a public server at GMD with which users can register to create a login and to add their own shared workspaces. We are thus continuing the process begun with the release of BSCW version 1 in October 1995, which attracted over 1500 registered users on our public server and over 200 downloads of the server software for local installation. The development process for BSCW is therefore iterative; experiences from user feedback following the release of BSCW1 informed much of the development of BSCW2, particularly the user interface aspects, and this process will continue over the coming months.

The primary goal now is to 'bound' the BSCW system and provide an open implementation of the basic cooperation mechanisms suitable for extension, both for research purposes and for customisation to the needs of particular organisations and application domains. We then intend to use the BSCW 'kernel' implementation ourselves as a platform for further research in the area of CSCW, allowing our research prototypes to be deployed and evaluated in realistic domains due to the support for existing platforms and applications which BSCW provides.

One area we are currently investigating is the support for server-side document compression and conversion. Currently information providers must decide the format in which their information is made available on their Web servers, and if users do not have applications which support these formats or cannot convert them locally this information is inaccessible. We are looking at ways of providing a suite of translation tools at the server which can be dynamically chained together, in a manner rather like a Unix pipe, to transform documents to a user-specified format. As an example of application, users with a slow connection would be able to request pre-compression of documents before downloading, and select a compression format which they can decompress locally.

Another enhancement concerns the usability of the system. The current architecture of the Web separates the presentation of a Web-based application at the browser from the 'deeper'

levels of interaction which are managed by the server. This allows simple operations such as scrolling and text editing to be handled autonomously by the browser, but other operations must be sent as a request to the server, and are thus subject to network delays, even if these operations only change the state of the user interface. In BSCW2 we have tried to reduce the need to go to the server with the multiple selection toggles, allowing the same operation to be applied to multiple objects with only one request. Despite this, there are still occasions where the browser must send a request simply to update the state of the user interface; for example, the 'select all'/'select none' toggles (Figure 1) where the HTML workspace listing must be re-generated to modify the selection checkboxes associated with each entry in the workspace listing.

For this and similar situations such as folding/unfolding actions, descriptions and folders, we have developed a method for enhancing the HTML sent back to the browser with small pieces of code written in JavaScript, which are executed by the browser to update the presentation of a HTML page. JavaScript is a simple but powerful scripting language which, among other things, can be used to add flexibility to HTML pages. JavaScript is currently only supported by recent versions of the Netscape Navigator and Microsoft Internet Explorer browsers, thus we will use it for features such as the 'select all/none' case where its application is only to enhance the interactivity—all system functionality will remain accessible using non-JavaScript browsers.

This approach, of providing enhancements to BSCW for certain Web browsers and end-users is certain to increase in the future as we look to deploy more specialised services for particular application domains. An example of this is the work we are currently doing to provide an environment for managing Web sites with the BSCW system (Section 3.3). Here we have used the Netscape Navigator Gold Web browser, which provides a WYSIWYG interface for editing as well as displaying HTML pages, to develop a prototype of an integrated environment for collaborative management of the pages on a Web server. As the richness of Web pages and the complexity of Web sites grows, it is clear that integrated tools to assist with Web site management will become increasingly necessary.

Finally, in January 1996 a project called CoopWWW[10], funded by the Telematics Applications Programme of the European Union, was started to extend the current BSCW system to provide a set of advanced cooperation services. These services include support for cooperative decision making, interfacing to directory services such as X.500 and synchronous video conferencing, integrated with the shared workspace metaphor supported by BSCW. For example, with one of our CoopWWW project partners we have developed the concept of a 'meeting' object, which allows users to set a date for the meeting, invite participants (from their address books), set the agenda and so on. At the appointed time this object is instantiated as a CU-SeeMe[11] video conferencing session, which allows participants with the CU-SeeMe software, available for Macintosh and PC platforms, to participate in the conference. Users can join the meeting through the BSCW system by clicking on the 'join' operation which launches CU-SeeMe on their computer and connects to the conference management system (the CU-SeeMe 'reflector') with the correct parameters. This kind of integration demonstrates the novelty of BSCW as providing a platform for cooperation services; in this case, the meeting object need not

---

[10]. http://orgwis.gmd.de/COOPWWW/

[11]. http://goliath.wpine.com/cu-seeme.html

only be a method to join the video conference, but might also be a point of coordination beforehand to formulate the agenda or afterwards an access point to the meeting minutes.

6. Conclusions

The BSCW Shared Workspace system is a Web-based CSCW tool offering basic facilities for collaborative work. This paper has described the current version of the system, BSCW2, which shows how the Web can be transformed from a primarily passive information repository to an active tool for cooperation, without compromising the benefits of the Web as a cross-platform tool for information sharing. The design of this system has been informed by the release of a previous version to the public domain and, as BSCW2 is now publicly available, we expect to repeat this process in the coming months.

The World Wide Web is developing at a furious pace, with new innovations appearing with every release of Web browser and server software. As such, designing any application for the Web is in many ways designing for a moving target, and it is not clear which innovations will be discarded and which will become accepted standards in the future. Our approach is to keep the requirements for using the basic cooperation services of BSCW to a minimum, based on accepted standards like HTML and HTTP, to make the system available for a large population of users with a minimal technical infrastructure. At the same time, we are investigating the possibilities offered by innovations like JavaScript and WYSIWYG HTML editing to provide more specialised services. In doing so, we hope to ensure BSCW continues to provide basic but useful mechanisms to support cooperation for widely-dispersed working groups, across computing platforms.

Acknowledgments

References

Ames, A., Nadeau, D. and Moreland, J. (1996), The VRML Sourcebook, John Wiley and Sons.

Bäcker, A. and Busbach, U. (1996), DocMan: A document management system for cooperation support, in Proceedings of 29th Hawaii International Conference on System Sciences, volume III, Maui, 3-6 January, pp 82-91.

Bentley, R., Horstmann, T., Sikkel, K. and Trevor, J. (1995), Supporting collaborative information sharing with the World Wide Web: The BSCW Shared Workspace system, in Proceedings of the 4th International World Wide Web Conference, Boston, Massachusetts, 12-14 December, O'Reilley & Associates, pp 63-74.

Bentley, R. and Appelt, W. (1997), Designing a system for cooperative work on the World Wide Web: Experiences with the BSCW system, in Proceedings of the 30th Hawaii International Conference on System Sciences, Maui, Hawaii, 7-10 January, in press.

Berghel, H. (1996), The client side of the World Wide Web, in Communications of the ACM, 39(1), January, pp 30-40.

Broll, W. (1996), VRML and the Web: A basis for multi-user virtual environments on the Internet, in Proceedings of WebNet'96, San Francisco, 16-19 October, in press.

Dourish, P. and Bellotti, V. (1992), Awareness and coordination in shared workspaces, in *Proceedings of CSCW'92*, Toronto, ACM Press, 31 October-4 November, pp 107-114.

Frivold, R., Lang, R. and Fong, M. (1995), Extending WWW for synchronous collaboration, in Computer Networks and ISDN Systems: Proceedings of the Second International World Wide Web Conference, 28 (1-2), December, pp 69-75.

Fuchs, L., Pankoke-Babatz, U. and Prinz, W. (1995), Supporting cooperative awareness with local event mechanisms: The GroupDesk system, in Proceedings of ECSCW'95, Stockholm, Sweden, 11-15 September, Kluwer Academic Publishers, pp 247-262.

Gorton, I., Hawryszkiewycz, I. and Fung, L. (1996), Enabling software shift work with groupware: A case study, in Proceedings of 29th Hawaii International Conference on System Sciences, volume III, Maui, 3-6 January, pp 72-81.

Kirby, A. and Rodden, T. (1995), Contact: Support for distributed cooperative writing, in Proceedings of ECSCW'95, Stockholm, Sweden, 11-15 September, Kluwer Academic Publishers, pp 101-116.

Lea, R., Honda, Y. and Matsuda, K. (1996), Virtual Society: Collaboration in 3D spaces on the Internet, in International Journal of Computer Supported Cooperative Work: Special Issue on CSCW and the Web, Kluwer Academic Publishers, in press.

Rao, V. (1995), The implementation of satellite offices: Initial recommendations based on observations from one site, in Proceedings of 28th Hawaii International Conference on System Sciences, volume IV, Maui, 3-6 January, pp 426-436.