A Multi-User Design Workspace

James Rutherford

Department of Architectural and Design Science
University of Sydney
Sydney NSW 2006
AUSTRALIA

Advances in digital media and digital communication has fostered the growth of a new enabling technology that allows geographically displaced individuals to hold group meetings and provides the opportunity to interact in a collaborative venture. The development of computer software to aid remote collaboration has, until recently, focused on the provision of tools that enable two or more people to participate in the shared authoring *of a mixed-media (lexi-visual) documents. This paper presents a model of the design process which is founded on the transient nature of collaboration. The model is used to develop a multimedia framework to support remote collaborative design providing transitional support between synchronous and asynchronous design activity. A prototype system is used to illustrate the salient features of the framework.*

Keywords: collaborative design

1 Introduction

A building design is the result of a team effort involving a multi-disciplinary group of designers, clients, and councils. The development of a design is conducted through coordinated consultation with each professional group involved in the process, with the communication of information being facilitated by the exchange of production documentation between individuals and organiations.

Design support tools, in the form of CAD systems, have been developed to assist designers in solving a variety of domain specific problems and in documenting and communicating the progress of an emerging design solution. With the computer providing the medium for communication, recent developments in Computer-Supported Collaborative Work (CSCW) and innovations in groupware now provides a basis for people to interactively share information and to hold meetings in which the participants are geographically distributed. Computational support for remote communication and collaborative problem solving exists in many forms, ranging from shared authoring via electronic mail to multimedia application frameworks that support remote conferencing, and information browsing and retrieval. While this form of application software provides effective casual conversational tools, there is a functional disparity between design tools used for production work and those groupware applications that are used to facilitate distributed meetings. Where CAD systems are inherently single-user applications, current groupware applications lack the level of support, rigour and precision that is required to assist in the resolution of complex design problems. What is required is a means of sharing and interacting with the original representation of the design data.

The notion of a shared workspace as a means of sharing all or part of a desktop with other users facilitates the concurrent or synchronous sharing and manipulation of data. Generally this is achieved by virtue of the computing technology, usually a client server model, deployed to manage the users desktop environment. While the immediacy of such a framework facilitates the communication of information at a resolution that is necessary for effective problem solving such systems do not fully address the issue of collaboration or the nature of the information being shared.

This paper describes preliminary work (Maher & Rutherford 1993) directed towards the development of a mixed media collaborative design environment within which com I design information may be interactively defined, manipulated, and visualised in a collaborative manner.

2 A model of collaborative design

Although collaboration is an essential aspect of the design process, involving the exchange of information about the emerging product and its development through revision histories, it is perhaps wrong to assume that design is centred entirely around collaboration.

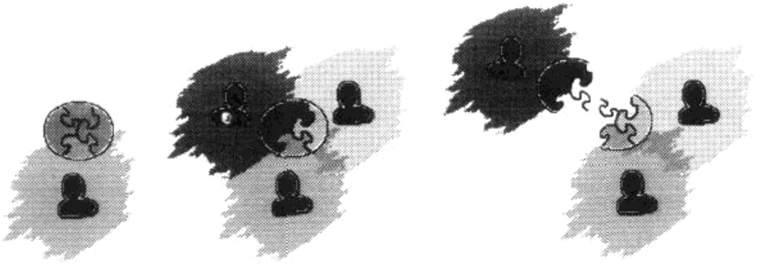

Figure 1. Transient nature of the design process.

Therefore, in developing a model for a collaborative design environment it is assumed that design is a process which is in continual flux, moving between asynchronous and synchronous activity where group membership over the life-cycle of the deign is dynamic and transitory (Condon 1993).

Figure 1 illustrates the transient nature of the design process in terms of collaboration. A designer working in isolation on some aspect of the design may choose to join forces with other members of the design team or with external consultants to either integrate design information or resolve design problems and conflicts. An accomplished designer will recognise the need to collaborate early in the design process and provide room in the design to accommodate multi-party interaction.

In such a model communication and continuity become important issues. The representation of the design, although changing over time, must be consistent with other views and uses of the design information. While production documents are created using a convention of graphic elements or a common syntax of symbolism, these entities when arranged in particular ways imply information not represented on the drawing; such as functional zoning, segregation and grouping of services. This information may be considered to be part of the design semantics, which if captured, represented and presented during the design process, would ultimately improve understanding of design intent accross disciplinary and temporal boundaries.

In order to sustain a collaborative venture a framework is required that facilitates the capture of design semantics and that accommodates both synchronous and asynchronous modes of operation, providing transitional support between them. This paper proposes a model for collaborative design and presents an implementation of a prototype collaborative CAD system illustrating the requirements for a shared multi-user workspace.

4 Requirements of a collaborative design environment & framework

A system for group-oriented design decision making should provide a means for switching between synchronous and asynchronous processes. The system should allow the use of a range of design tools, as determined relevant by the design team, that can be used in a synchronous or asynchronous mode and that can communicate with a persistent representation of the current state of the design solution. The requirements for a collaborative design environment fall into three major categories: a shared workspace, an application domain, and data management.

(1) A shared workspace provides a group of designers with shared access to the relevant applications where each designer sees the same visualisation/ data of the design. This component is responsible for distribution and multiplexing the input and output of each application and for defining the floor control for turn taking. The shared workspace is used when a group of designers work at the same time and the computer is the medium for communication. This provides explicit support for a design session.

(2) An application domain provides a variety of design tools from which the design team can choose the best suited for their specific task. Ideally, any commercial or custom-built application should be available for access in the collaborative environment.

(3) A data management component provides support for the persistent storage of design decisions and access to the current state of the design project. Ideally, any DBMS could be used in a collaborative environment. The support for synchronous collaboration adds an additional requirement that the individual that contributes to the decision making is seamlessly and automatically attributed ownership.

4.1 A framework for collaborative design

The requirements as outlined above provide the basis for developing a framework for collaborative design environments. The framework has three major components, as illustrated in Figure 2.

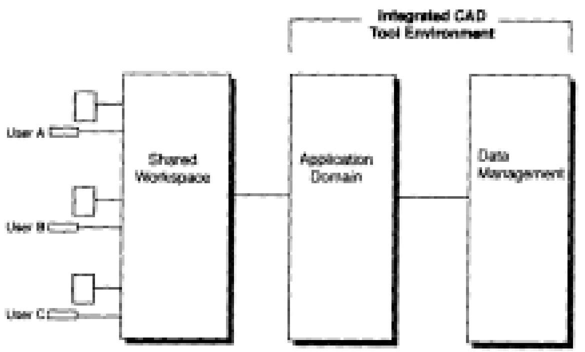

Figure 2. Schematic representation of a collaborative design tool environment.

The shared workspace component builds on the technology of CSCW, where a multi-user interface is provided for shared access to a specific tool. The combination of the application domain and the data management component provides an integrated design environment. This framework essentially combines the work of CSCW and integrated CAD to support both synchronous and asynchronous collaboration. Each of the three component are further described below.

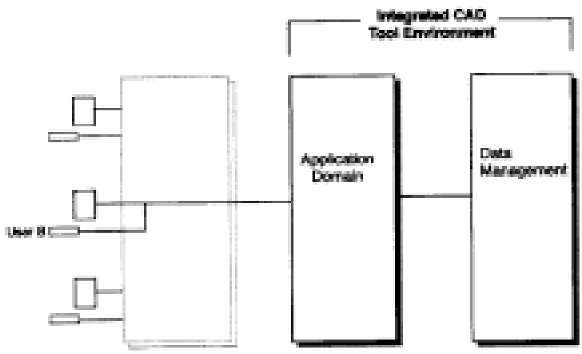

Figure 3. Transitional support

The modularity of the framework provides a simple means of accommodating the transition between synchronous and asynchronous activity. The shared workspace enables any existing application to be distributed amongst member of design team. By not invoking this module the application reverts to being a single user application, as illustrated in Figure 3, but with data management extensions that provide access to the persistent design data created during an collaborative session.

4.2 *Shared workspace*

A shared workspace, as illustrated in Figure 4, includes the utilities for distributing and multiplexing the contribution of each user to the appropriate application and for establishing a protocol for turn taking among the users.

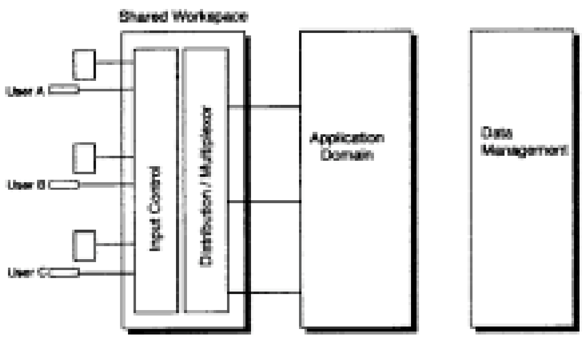

Figure 4. The shared workspace

The implementation of a shared workspace can be derived from similar work in the development of shared drawings boards or in tools for sharing Xwindow applications.

4.3 *Application domain*

The application domain for a design framework should not be limiting in the number and variety of tools that can be utilised and deployed. This is easy to achieve at the interaction layer. Simply intercepting protocol request from an Xclient provides a transparent mechanism for sharing the visual interface of any application that exists within the client server model. However, there is also a need to manage the data that application program generates. It is important to introduce a similarly transparent means of achieving this enabling applications to be "dropped' into the environment with little or no modification. Figure 5 illustrates the interaction between an application and the collaborative design environment.

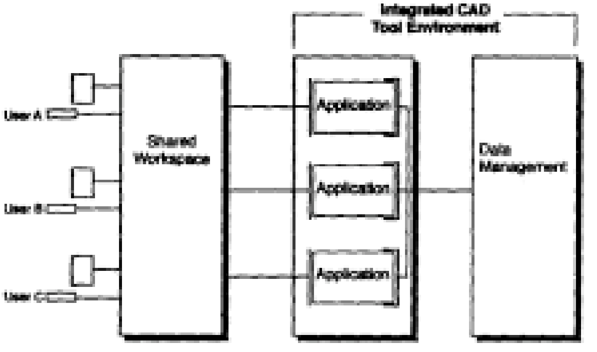

Figure 5. Application integration - transparently coupled at the interaction layer. Tightly integrated at the I/O layer.

The data management component of the framework provides the interface between the application domain and a representation of the design entities and their aggregation as functional system. A database monitor allows a designer to define a class or instance of an object. How the objets are defined and what properties or semantic content they have will be governed by how the system is to be used. Attribution, confidentiality and willingness to share data may affect collaboration. In order to motivate the electronic sharing of ideas it is necessary to introduce methods of attribution and recognition to preserve intellectual property rights, if this is an issue, and encourage collaboration by recognising those who contribute.

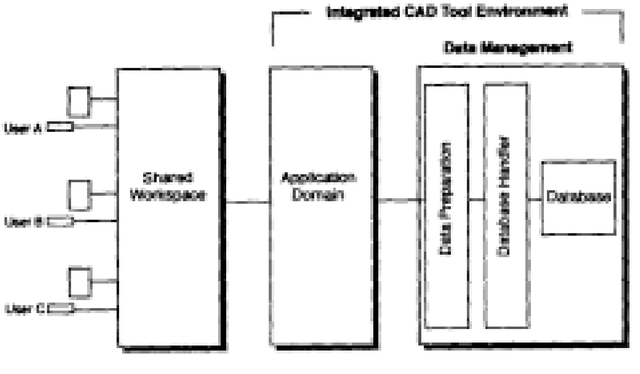

Figure 6. Data preparation in a synchronous collaborative design environment.

Professional accountability and liability are also major concerns which must also be addressed and accommodated in a robust and reliable framework. Figure 6 illustrates the considerations in a typical transaction of the data management component.

5 Implementation

The individual components of above framework have been implemented and are described in the following sections.

5.1 Implementation of shared workspace

Sharing the visual interface of an existing application is possible by virtue of client server software technology such as X Windows, which is used for the purposes of this prototype application. The X Window system is based on a client server model (Nye, 1990), as illustrated in Figure 7, in which X Window protocol requests for creating windows, colour maps etc., are sent from a client to the server. The server responds by allocating the requested entity such as a window or a graphic segment such as a circle and passing a pointer back to the client. An example of an application is a CAD program which is resourced by the XWindow server.

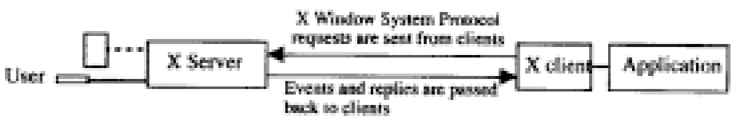

Figure 7. X Window client server model - after Nye (1990)

The advantage of a client server model, such as that provided by X Windows, is that neither the client or the server need to be resident on the same host. This facilitates a distributed application environment. Taking this model a stage further, by mimicking the functionality of a server and intercepting protocol requests from a client application it becomes possible to multiplex the visual interface of a single application giving a number of users shared access to the control of a single process.

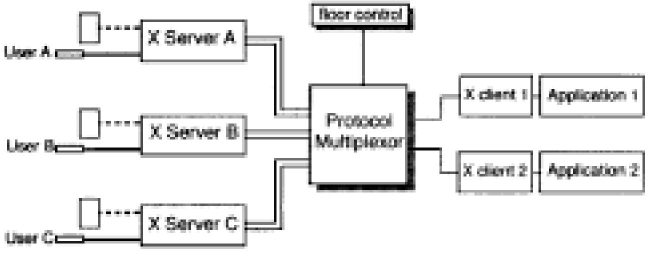

Figure 8. XShare, an X protocol multiplexor. (McFarlane 1991)

A tool kit that enables a design team to share an existing CAD system may include software such as XShare. As a tool for sharing X Window applications among a predetermined set of displays on a network, XShare sits between the client and server and intercepts X Window System protocol requests and replies from and to the client, as illustrated in Figure 8. These requests are then duplicated for each end-user. The result is a single process with a multiplexed interface, where the applications (for example, a CAD Y s ~are unchanged and operate as if they were supporting a single user with input from different displays. Since XShare allows any proprietary Xclients to be shared, existing CAD systems that run under X Windows can be shared.

5.1.1 Floor control

An integral aspect of a shared workspace is turn taking. Most groupware utilises control mechanisms to prevent or resolve interpersonal conflicts that may affect productivity. Condon (1993) identifies a political taxonomy for CSCW where many systems may be categorised as either: fascist, communist, or anarchist. According to Condon:

(1) the fascist system is one controlled by a chair person who is control of the session and decides who has control at any particular time.

(2) a communist system is one where the system itself is the ultimate authority controlling access to files to ensure data integrity and providing controlled voting procedures for the consolidation of joint documents.

(3) an anarchist CSCW system is one whereby control is freely available to every participant.

While the latter model is a more appropriate means of fostering collaboration it assumes that other channels of communication (such as voice and video) allow people to negotiate the control of the meeting and subsequent changes to the shared application. This approach can result in chaos when the other channels of communication are forgotten during decision making or when the number of participants is larger than two.

Figure 9. Token based floor control

For this reason a mechanism for scheduling designers access to the interface is required. Figure 9 illustrates an interface for establishing user control of an application. It provides a simple token based floor control mechanism comparable to passing a pen in a meeting. Control is obtained by picking up the pen, enabling a task to be completed without interruption. Control is passed to other participants by releasing the pen to the group.

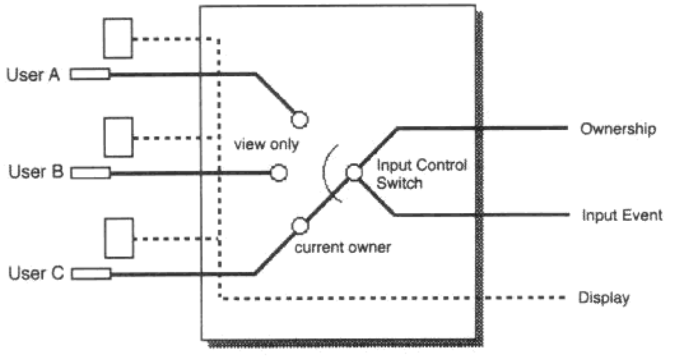

Figure 10. Input control in the shared workspace.

The shared workspace environment, comprises an X protocol multiplexor, Xshare, which facilitates the distribution of a given application domain, and a floor control process for managing the sharing of input. In addition to switching input control between the various participants, the floor control process also provides input ownership information to the data preparation tool at the back-end of the system. This has been introduced in the form of a transaction manager operating at the user interaction layer of the client/server model. The interface provides users with control mechanisms and an input status, is illustrated in Figure 10. While a person interacts with the CAD tool other members of the group are given 11 view only privileges until control is released to the group as a whole. This makes the protocol for changing the design explicit and eliminates potential conflicts in data entry.

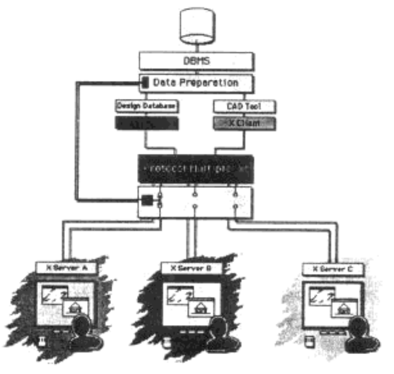

Figure 11. Schematic diagram of shared environment.

The floor control mechanism is simply a binary switch that controls the transmission of keyboard and pointer events to the server. Figure 11, illustrates the relationship between individual servers and the floor control process. The diagram also illustrates how the ownership of input is directed to the data management module.

5.1.2 Establishing a connection

The shared workspace, in addition to distributing the interface of an Xclient, acts as a session manager and is used to establish a parallel communications channel between individual participants. This is necessary in order to pass the control amongst the group and provide an indication of the current input owner.

Although the X protocol multiplexor used for this prototype system does not support a late connection, the shared workspace enables participants to join and leave a session at any time.

5.1.3 Sharing a document

This module also provides a transparent mechanism for sharing documents. Once a connection is established the end-user can share a document by selecting it from a file menu. The workspace module determines what type of file it is and then binds the file type to an appropriate application program. The program is then execute the X protocol multiplexor with the appropriate display arguments. Since each participant is required to run the shared workspace module in order to participate in a collaborative session, each person is able to share documents with the group. This module also provides an interface to third party audio/video conferencing tools such as Xerox Parc s network video application (NV). Audio or video connections may be established simply be selecting the audio and/or video tools. These applications are then deployed by the shared workspace interface with the correct connection details. The modularity of the approach ensures complete flexibility and long term support. Replacement communication tools can be inserted transparently.

The shared workspace provides a means for distributing the user interface of an Xclient application amongst members of a design team. While this provides the necessary level of functionality required to support various design activities such as the creation of a CAD drawing or the use of a simulation or visualisation package, the workspace in itself does not provide a means for recording design intent. This is the domain of the data management module which is coupled to the application domain.

5.2 Implementation of an application domain

The application domain needs to be able to interact with the shared workspace component and the data management component. As described above, in this implementation any X window client can interact with the shared workspace with no changes made to the application itself. There are two options for interfacing an existing application with a data management tool: 1. no change is made and the application updates its own data file; 2. the application is integrated with a global database. While it is desirable to opt for the integrated method it is assumed that a mixture of methods will exist in a collaborative environment such that a CAD tool will manage its own data files while interfacing with a global database to provide persistent semantic information.

AutoCAD is the CAD application implemented as part of the application domain. In addition to the representation of the drawing that AutoCAD maintains, a database representation of design semantics is maintained by the data manager. In order for AutoCAD to interact with the data manager it was necessary to customise the application to support dynamic data exchange between the data management tool and to provide the end-users with a mechanism by which to assign additional semantic information to a graphical entity. Information is stored and retrieved to and from the database in a variety of ways through the implementation of the following commands.

(1) dbinit: this initialises a pointer to a particular database

(2) classify: enables a designer to classify any graphical element/s on the drawing by associating the selected objects with a data object in the database. An opportunity is provided to create a new class or instance of object for association.

(3) identify: enables semantic information associated with a graphical entity to be retrieved from the database. If the graphic element is part of a larger object, the composite object is highlighted. For example, a line may be selected and using the retrieval command the user may discover that it is a part of structural column.

(4) search provides a convenient mechanism to retrieve all instances of a particular class of object - search DOORS all

(5) would highlight all doors in a model. To limit the search it is possible to qualify it with a convergence term of the form:

search DOORS where WIDTH < 1000mm and FLOORLEVEL = 1

would highlight all those doors that are on the first floor and satisfy the width requirement. Structured queries are also possible. For example, using a command of the form:

search FURNITURE where ROOMID = "directors office"

would highlight, on the drawing, all furniture in the specified room. Another example might be:

search DUCTING where size < 300mm

(6) annotate: provide a mechanism for associating additional, unstructured semantic information with a set of entities. Annotations may be provided using any tool such as a sketching package, text editor, voice recorder. These are then stored in the database as a multimedia (large) object using the annotation tool.

(7) annotations: retrieves the annotations of a selected group of entities. The annotation data is presented for editing using the tool that used initially to create it.

(8) modify: enables the user to change a specified property of an object. For example:

modify DOOR.name="directors door' finish = "mahogany"

The Postgres monitor is available as an application tool that gives the users direct query access to the object-oriented database. Having both the CAD application and the database monitor application open allows the users to view the graphic representation of the design and the semantic representation at the same time. The commands implemented within AutoCAD, as described above, access links between the graphic and semantic data. The users can select a graphic symbol in AutoCAD and see the associated semantic information in the Postgres monitor window. The AutoCAD commands that provide this facility are: classify, identify, search, and modify.

The annotation tool allows the users to associate mixed media representations with the semantic data without using AutoCAD for the graphic representation. The users can associate a bit mapped image with an attribute in the database, or possibly associate a sound file or video segment. The annotation tool stores the image, sound, or video segment within a large object in Postgres. The commands implemented to facilitate associating and viewing annotations are annotate and annotations.

5.3 Implementation of data management

A major consideration in the development of a collaborative design environment is the need to record the design decisions made during a session. The design decisions should, by consensus, be made on the basis of the currently available project information and should contribute to that information. Alternatives in collaborative environments include:

(1) recording the session on video

(2) recording a history of commands entered into the shared workspace

(3) recording versions of data in a database.

The approach taken in this implementation is to record the design decisions in a shared representation of the design in a database. The database contains a semantic data model of the design that is linked to the graphical representation of AutoCAD. Figure 12 provides an illustration of the use of the semantic data model. Both physical attributes, such as width, height and finish, and semantic information are represented for a graphical entity representing a door.

While this additional information is stored as a semantic data model (Bjork 1991) and may be as detailed or as abstract as necessary to convey information to another designer it is intended as a design journal rather than a product model and chronicles the progression of design solutions and revisions. The database is capable of transparently handling many kinds of data thereby providing a flexible and comprehensive environment for designers to work collaboratively. Additional semantic information may be provided as an unstructured annotation or more rigorously using object classification by associating a graphical entity with an object in a database.

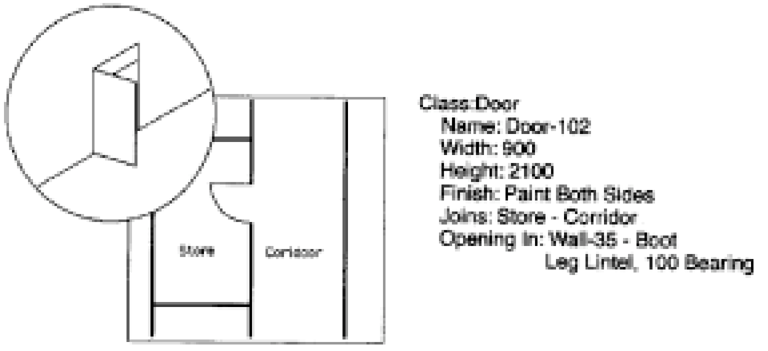

Figure 12. Data model for a door.

The database is maintained by a database management system; in this implementation an object-oriented DBMS called Postgres is used. The data preparation tool considers a transaction as illustrated in Figure 13. Data preparation requires a minimal set of attributes to be associated with the elements of the database entities: author, time, data, and possibly an unstructured annotation - the who, when, what and why of a design decision, represented by the four slots in the transaction frame. The data preparation tool waits for a number of conditions to be satisfied in the application domain before packaging and committing the transaction frame to the database handler. Here the general protocols of the data preparation tool are translated into the more specific transaction language of a given database management system.

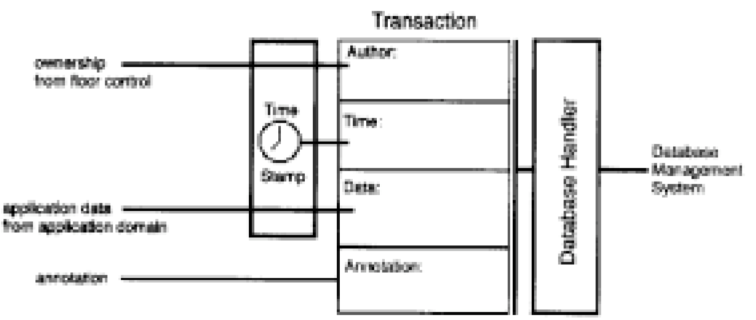

Figure 13. Data preparation in a synchronous collaborative design environment.

The first condition to be satisfied in this process is the ownership of the input at the front-end. Whenever the input is exchanged between the participants the data preparation tool is notified of the new owner. Ownership also indicates who is generating the data. When data is generated by the application it is intercepted and passed to the data preparation tool. A time stamp is obtained from an internal register and the transaction information is completed. The composite information is then committed to the database via the database handler.

Annotations are a special case of data. Annotations consist of mixed media such as text, graphic, audio/visual material which are stored in the database as multimedia objects. Each object contains a pointer to the annotation data and a symbolic reference to the application that was used to produce it. When retrieving an existing annotation the original tool used to create it will be deployed automatically.

6 Prototype Design Environment

The framework described above has been used to implement a collaborative CAD environment. The shared workspace is implemented using a multiplexor for X windows. The application domain consists principally of 3 tools: AutoCAD, a Postgres database monitor, and an annotation tool. The data manager comprises data transaction utilities in a Postgres database management system. The desktop of one of the users is shown in Figure 14.

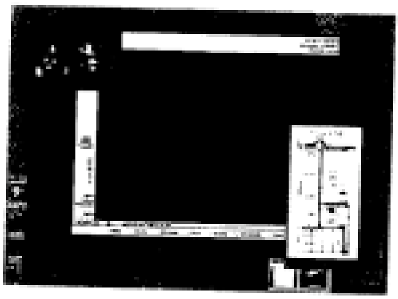

Figure 14. The implemented collaborative CAD environment.

The illustration also shows a multimedia annotation, in this case a static image of a cross section through a doorway. Other information such as video material describing the site or product information may also be associated with the evolving building description.

7 Conclusions

Current groupware supports casual collaborative design activity by providing tools that enable participants to share documents and sketch revisions in a conversational manner. These tools do not provide persistent storage of the design decisions made during a collaborative session owing to the ephemeral nature of the medium - the state of a session only lasts as long as the session itself.

The shared workspace metaphor provides a means to participate in collaborative design ventures by offering a framework within which complex design data can be created, manipulated and visualised using a variety of tools and media that are already familiar to the participants.

Given that a model for collaborative design must provide transitional support between synchronous and synchronous activity, it is necessary to extend the metaphor of a shared workspace to include persistent storage of design semantics to take account of the transient and dynamic nature of group membership within a design team. This is achieved by means of a data management component which is coupled to the application domain. The data management component of the system supports the persistent storage of design semantics in an unstructured and opportunistic manner. While such an approach does not hinder the design team, further studies are required to establish the nature and structure of design semantics that are required to support collaborative design decision making.

The current implementation of the shared workspace includes a floor control component which is used to enforce an interaction protocol. Further research is required to investigate social interaction during computer supported collaborative design to determine the appropriateness of the current solution and to establish alternative interaction protocols. This work would influence future models of the collaborative design process.

In summary, given that a model for collaborative design includes a shared workspace, an application domain, and a database management component; the requirements for each of these major components needs to be identified to explicitly support asynchronous collaboration while also supporting individual design activity.

8 Acknowledgements

This project is supported by the Australian Research Council. Some of the ideas presented were developed in collaboration with Mary Lou Maher. The author acknowledges the Key Centre of Design Computing for the resources and people that have influenced this work. The permission of the Managing Director of Telstra Corporation to publish this work is grateful acknowledged. The views and opinions expressed in this work are those of the author and do not necessarily imply Telstra policy or future service offerings.

9 References

(1) Bjork, B.-C. Intelligent Front-ends and Product Models, Artificial Intelligence in Engineering, 1991, 6(1),46-56.

(2) Condon, C. The Computer Won't Let Me: Cooperation, Conflict and the Ownership of Information, in S. Easterbrook (ed.), Computer Supported Cooperative Work: Cooperation or Conflict?, Springer-Verlag, London, 1993 pp. 171-185.

(3) Easterbrook, S., Beck, E. E., Goodlet, J. S., Plowman, L., Sharples, M., & Wood, C. C. CSCW Systems, A Survey of Empirical Studies of Conflict, in S. Easterbrook (ed), CSCW: Cooperation or Conflict? Springer-Verlag, London, 1993, pp. 52-66.

(4) Kraut, R. E., Egido, Carmen, Galegher, J.Patterns of Contact and Communication of Scientific Research Collaboration, in J. Galegher, R.E. Kraut, & C. Egido (eds.), Intellectual Teamwork, Social and Technological Foundations of Cooperative Work, Lawrence Erlbaum Associates, London, 1990 pp. 149-171.

(5) Latta, S. Beyond Communication: Building Collaborations, Access Magazine, 1992

(6) Maher, M. L., & Rutherford, J.H. (1993). Video Conferencing and CAD, Proceedings of the 27th Annual Conference Australian and New Zealand Architectural Science Association, pp. 113-118.

(7) McFarlane, C. M. (1991). Xmux - A System for Computer Supported Collaborative Work, Conference Proceedings of the 1st Australian Multimedia Communications Applications and Technology Workshop, 15.

(8) Newton, P.W., Wilson, B.G., Crawford, JR., & Tucker, SN. (1993). Networking CAD in the AEC Sector: An Introduction, in P.W. Newton (ed.), Networking CAD, Telecom Research Laboratory, CSIRO Division of Building, Construction and Engineering, Melbourne, pp. 19-33.

(9) Nye, A. (1990). Introduction to the X Protocol, X Protocol Reference Manual, 2nd Edition, O'Reilly & Associates, 0:3-36.