# A computational model for synchronous collaborative design

**M. Saad and M. L. Maher**
Key Centre of Design Computing
Department of Architectural and Design Science
University of Sydney NSW 2006
Australia

**Abstract**

Designers engaged in a team design project need to establish, develop, and maintain a shared environment in which they can collaborate. Design then emerges from the interaction between the participating designers. In this paper we consider issues involved in the development of computer-based design environments in which teams of design professionals can collaborate, focussing on the need for visual and underlying representations which can support multiple interpretations. We consider the environment as providing a shared workspace which facilitates both communication and progression of design ideas, concepts, and drawings. In the environment presented here, the shared workspace has two focusses: the workspace that designers see and interact with, and the workspace that provides an underlying computer-based representation for persistent memory. A representation is presented that can support the representation of semantic design knowledge that needs to be communicated during collaboration, as well as the data needed by computer programs for displaying the geometry.

## 1 Introduction

Design is rarely an activity that is commenced and completed by an individual. The more common design environment is one in which teams of designers work together towards a final solution. Design research, until recently, has focussed on supporting individual activity in design. Computer-aided design (CAD) is used in the design professions as a tool for visualising and documenting a design solution. The mode of use of a CAD system is currently limited to a single-user interface. This interface provides the technology for one person to view the drawing and make the changes in geometry or view. Implicit in this technology is the assumption that using CAD is an activity that is limited to an individual. Distributed CAD currently means that many *individuals* can access the CAD data, such as a centralized file server, not that more than one person can work simultaneously on the same drawing, where each person sees the same view. In this paper, we present another perspective on distributed, collaborative CAD, one in which synchronous collaboration is accommodated through a multi-user interface to a more comprehensive CAD environment.

Recent developments in artificial intelligence, distributed systems and computer-supported collaborative work as well as symbolic representation of design schemas, provide an opportunity to develop CAD systems that support collaborative design. In this paper we review the relevant recent developments and identify what these developments offer for collaborative design. Then we present a model for computer-supported synchronous collaborative design, focussing on the issues related to the development of shared workspaces.

# 2 Artificial Intelligence in Collaborative Design

The recognition that much of human problem solving and activity involves groups of people has lead research in artificial intelligence (AI) to investigate concurrency and distribution in AI. Distributed AI deals with modelling problem solving as a collaborative effort among various agents. Research issues in DAI include modelling communication, control, negotiation, and problem solution space (Bond and Gasser 1988). Many of the prototypes that have been proposed, are based on the blackboard architecture for cooperation which is the medium through which all communication between specialists (also known as knowledge sources) takes place. However, despite its generality and flexibility, the blackboard framework is communication intensive and is not truly concurrent, since they run under the supervision of a global scheduler or a collection of integrated control knowledge sources. This framework could be used to identify conflicts and resolve them rather than support a synchronous collaborative design environment.

## 2.1 Distributed and Integrated Design Environments

Existing CAD systems offer powerful drawing capabilities and realistic representations of the design. The design data in CAD systems is represented as a set of graphic or symbolic objects (e.g. lines, polygons, text, etc). Attributes are attached to these objects to represent non-graphic information. Based on the nature of design as a collaborative work, distributed and integrated CAD have become important issues. *Distributed* CAD implies that the CAD data is available across a distributed network. *Integrated* CAD implies that the CAD data can be read by more than one computer program, for example CAD systems from a variety of companies and other types of computer programs such as analysis tools.

The common approach to distributed CAD is the use of distributed databases. The hardware and software technology is being developed to support the communication and multi-user access to an electronic form of design data. The electronic form of the data ranges from shared files to shared DBMS access. Modelling of the design data provides the basis for establishing the semantics of the shared data. The assumption in distributed CAD is that access to the design data occurs through a predetermined CAD system or DBMS.

The efforts towards CAD integration have resulted in the development of various data exchange standards, as well as many "non-standard" symbolic models. The aims of the data exchange standards have varied from exchanging drawing data to exchanging product models. DXF and IGES are results of efforts on the exchange of graphics databases between CAD systems which deal with geometric information such as lines, circles, etc. The lack of design features in data exchange standards such as topology, relationship, etc, has shifted the focus towards product data exchange standards. The focus on product modelling (e.g. PDES, STEP, etc) has placed an emphasis on 3D solid modelling of geometry and the essential features that can be given to models as well as on non-geometric data and modelling the product as a set of related concepts or objects.

Although a representation of non-geometric information can be stored in a CAD database, most CAD systems do not use this information. Design data is extracted from the database and fed to another system that can reason or execute computations with such data. Interfaces exist to manage the exchange of information between various systems. To deal with the diversity of computer-based design programs, integrated CAD systems provide a basis for data integration. One approach is the use of relational database management systems (RDBMS) in which data is stored as tables, and accessed and modified using a relational query language, e.g SQL. Another approach is the use of object-oriented database management systems (OODBMS), based on object-oriented programming techniques, with utilities found in database management systems, that support encapsulation and complex objects. OODBMS are still in the early stage of development. However, the object-oriented approach shows significantly more promise from a semantic expressiveness perspective. This approach has been applied in various research projects (e.g. (Sriram et al. 1991), (Sauce and Powell 1992), (Poyet et al. 1990), (Law et al. 1991), (Fenves et al. 1990), etc).

To provide support for team design, emergence needs to be recognised and supported by the representation of the design solution. Emergence in design occurs when a new property that was not explicitly represented or intended is found in a design description. In order for a computer program to support emergence in design, two issues are relevant: representation of design intent and recognition of emergent properties. Design intent can be defined as the implicit design knowledge which leads to design decisions at any stage of the design process. An understanding and representation of design intent is required in a collaborative design session to allow the remote communication between designers and to extend the electronic representation of the design solution beyond the geometry or drawing data. Current research on explicitly representing design intent includes representing evolving artifact and design objectives (Ganeshan et al. 1991), representing the purpose or behaviour of design elements (Rosenman and Maher 1992) and representing the user/performance requirement at the design level (De La Garza and Oralkan 1992).

The *recognition* of emergent design properties is necessary in a computer environment that supports design since emergence is an important and common aspect of design, particularly in visually-oriented design domains such as architecture (Mitchell 1989), and in synchronous collaboration because each designer, as an individual, may see different things in a design drawn by his or her collaborator.

## 2.2 Computer Supported Cooperative Work (CSCW)

Generally, the main concerns of computer-supported cooperative work (CSCW) are about the study and theory of how people work together, and how the computer and related technologies affect group behaviour. Research issues in this area range from discovering models for interaction to applications development. Researchers are concerned with modelling group communication ((Pankoke-Babatz 1989), (Bannon and Schmidt 1991),(Smith et al. 1991), etc), the concept of information sharing ((Hennessy 1991), (DePaoli and Tisato 1991)), the requirements for activity management (Benford 1991), and capturing and representing group decision rational ((Lee 1990), (MacLean et al. 1991)).

In contrast, various techniques have been applied to enhance communication, coordination, and collaboration such as networking, concurrent processing, and windowing environments. These technologies help in producing CSCW applications which enhance teamwork. The current applications of this area are primarily in group meetings, long distance conferences, software development, game playing, and shared drawing systems. Relevant research in CSCW includes architectures for multi-user applications ((Patterson 1990), (Crowley et al. 1990)), models for a shared workspace (Ishii and Miyake 1991), and multi-user drawing tools (Dourish 1992). The application of this research to the design of physical objects is not well developed. The distinctions made by computer-supported collaborative design include the need to share design drawings, ideas, and rationale, and to allow for emergence to occur amongst the collaborators, where other applications focus on sharing text or graphics in the form they were entered into the computer. In applying this technology to design, the concerns are in the sophistication of the drawing primitives and display, the representation of design information beyond what each person sees on the screen, and the use of underlying representations of the design that support emergence.

# 3 A Model for Computer-Supported Synchronous Collaborative Design

Collaborative design can be defined as a group of designers working as a team on a shared representation of design requirements, drawings, and documents. Synchronous collaboration indicates that the collaboration occurs when all members of the design team work on the same documents/information/problem at the same time. This kind of collaboration requires extensive interaction between designers in a single and/or various domains. On the one hand, architects may, for example, collaborate on the development of a design concept (single domain). On the other hand, architects and structural engineers may collaborate to identify a solution to the interface of the architectural plan and the structural support system (various domains). The broad nature of design collaboration implies that computer-support for such activity must provide flexibility in the communication of design data and ideas.

Collaborative design involves many kinds of knowledge from different domains. Designers require different views of the design and have substantially different interests regarding the development of the design solution and its associated representation. Multiple levels of abstraction are needed to deal with the diversity of knowledge. In terms of computer support, different ways of interacting with other designers and design tools are needed to support the diversity of design.

We approach the development of computer support for synchronous collaboration through the development of a shared workspace, as illustrated in Figure 1. The shared workspace is the medium through which communication between the participants in the collaborative design occurs. The representation of the shared workspace is a focus for the development of computer-supported synchronous collaborative design. A shared workspace not only provides flexible and effective visual communication but also provides a medium in which one designer can understand another's model/design where design specialists do not necessarily have a shared vocabulary.

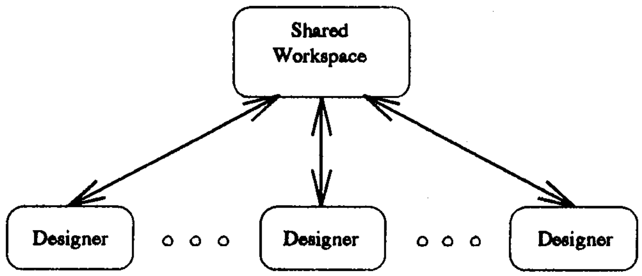


Figure 1: Coordination through shared workspace.

Therefore the underlying representation of the design elements in the shared workspace must be shared. This implies that the model illustrated in Figure 1 is too simplistic.

Pursuing this further, a shared workspace for design has two meanings in the context of computer-supported design: the workspace that human designers view and interact with, and the shared representation of the design problem that the computer uses for persistent memory and interprocess communication. Current research in the development of data exchange standards, in symbolic schemas for integrated design environments, and graphical representations that support shape emergence provide the basis for the computer-based representation. The technology available to support a shared representation includes distributed databases and networked workstations. Here we are concerned with the representation of geometry that supports semantic representation and, beyond the geometry, to include such considerations as function, behaviour, versions, general design knowledge, etc.

The shared workspace that human designers interact with is a visual one. Design information must be represented in the form designers currently use for communication, i.e. drawings, sketches, notes, diagrams, equations, graphs, etc. Many computer programs used in design already provide a visual interface. The concerns here are how to implement existing visual interfaces as a multi-user interface and what additional components are needed when people share a visual interface that was originally developed for a single user.

The shared underlying representation has many components: the representation of geometry, both in a form that a CAD system can use to produce a visual image on the screen; the representation of geometry that can support emergent form; and the representation of the comprehensive design solution with information about schemas and intent.

A model for collaborative design in which existing applications, such as CAD, modelling programs, analysis programs, knowledge-based systems, etc. can be shared by more than one designer is shown in Figure 2. The five components of this model are:

1. *Session Server.* Start up application process that is typically in charge of setting up the collaborative design session.
2. *Coordinator.* A special application process that embodies data management and control between applications and the shared workspace.
3. *Applications.* Existing applications and development tools that support the representation and the progress of design activities.

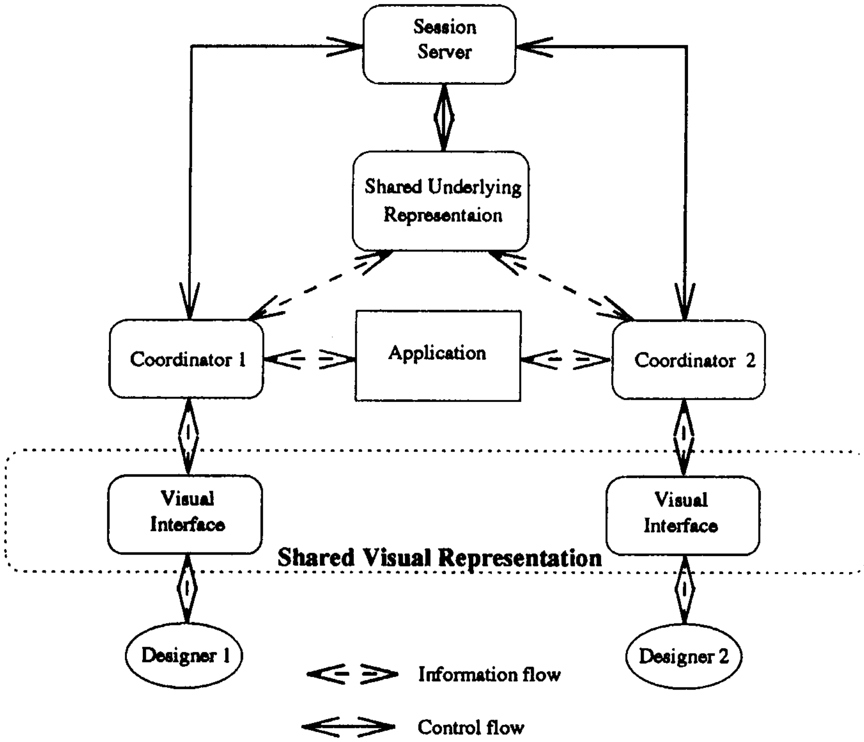


Figure 2: A Model for a Multi-User Synchronous CAD System

4. *Shared Visual Representation.* Visual sharing of design elements, with various support for design concepts and solutions.

5. *Shared Underlying Representation.* Generic sets of objects with the models and processes to support symbolic functionality of design, as well as specific design decisions which represent design elements and their relationships.

In the remainder of this paper we focus on the shared visual representation and the shared underlying representation, as these two components form the shared workspace.

## 3.1 Shared Visual Representation

The visual representation forms the most critical part of a synchronous collaborative design environment. It is the part where real-time interaction between designers and the visual representation of their design occurs. Design, in general, is represented by sketches and drawings. These are the visual symbols which are used by designers and they are usually communicated and saved on a drawing surface. When designers are using computer applications to visualise their ideas, an extension of the still 2D drawing surface to 3D and multimedia becomes available as a way to extend the visualization facilities. A multimedia approach to collaborative design enables designers to communicate at a distance through audio devices (talk to each other) and develop video or animation sequences in addition to the 2D and 3D models now available in

CAD systems. Another consideration is the partitioning of the workspace for the individual designers as part of a design team. The workspace can be partitioned into public vs private, graphics vs text, data vs knowledge base, and direct communication through text or voice as illustrated in Figure 3. Each partition can provide an interface to the underlying computer-based representation of the design and/or to the other individuals in the design team.

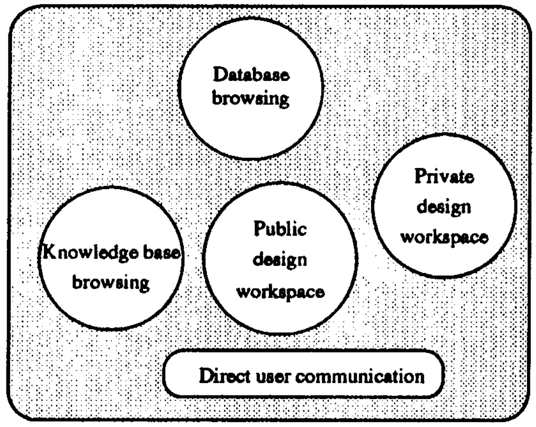


Figure 3: Illustration of a shared visual interface

The public design workspace of the proposed architecture is designed to support communication and sharing graphical images among individual designers. In many ways this part of the workspace is like the shared drawing packages currently being developed, however, we extend such drawing surfaces by providing a persistent memory of the design solution and include the representation of design semantics. This component of the shared workspace supports concurrent design by allowing participants to sketch/draw simultaneously into the common shared workspace. Designers are allowed to edit and process the graphical information of the shared workspace dynamically. The public workspace provides a set of graphics and symbolic objects that can be used during the collaborative design session. These objects provide a tool box for developing a design solution and visualising the effect of design decisions.

The private design workspace allows an individual designer to develop a personal design idea that is not ready to be shared. The visual environment would be similar to the public design workspace but it would only be visible to the person that owns it. Mechanisms for adding new information from the private workspace to the public workspace and vice-versa allow the design ideas to move from public to private or private to public.

In addition to a shared graphical workspace, database and knowledge base browsers provide a designer with access to data that is not geometric. The browsers provide a convenient medium for searching the underlying representation. They also provide access to multimedia elements that are attached to design objects.

A collaborative design session requires direct communication between designers. The direct user communication component provides a medium for coordination, negotiation and cooperative development of design ideas. These include electronic talking systems, video, and voice.

### 3.1.1 Supporting Synchronization in Multi-user CAD systems

The model so far describes a shared environment with different levels of abstractions in terms of supporting collaboration. Since CAD is an important tool in the design profession, we chose to focus on ***collaborative*** CAD in which synchronous collaboration is accommodated through a multi-user interface. A synchronous multi-user CAD system has been implemented with AUTOCAD as the basic CAD system. Multiple designers at different workstations can interact with the drawing at the same time. The system has an event driven mechanism that replaces the command driven interface of AUTOCAD. Each application process receives all the participants' input events. This results in a more responsive interface than in standard CAD systems. Specifically, if one designer is involving in adding some graphical objects to the drawing, the others designers can continue to draw or modify other design objects at the same time.

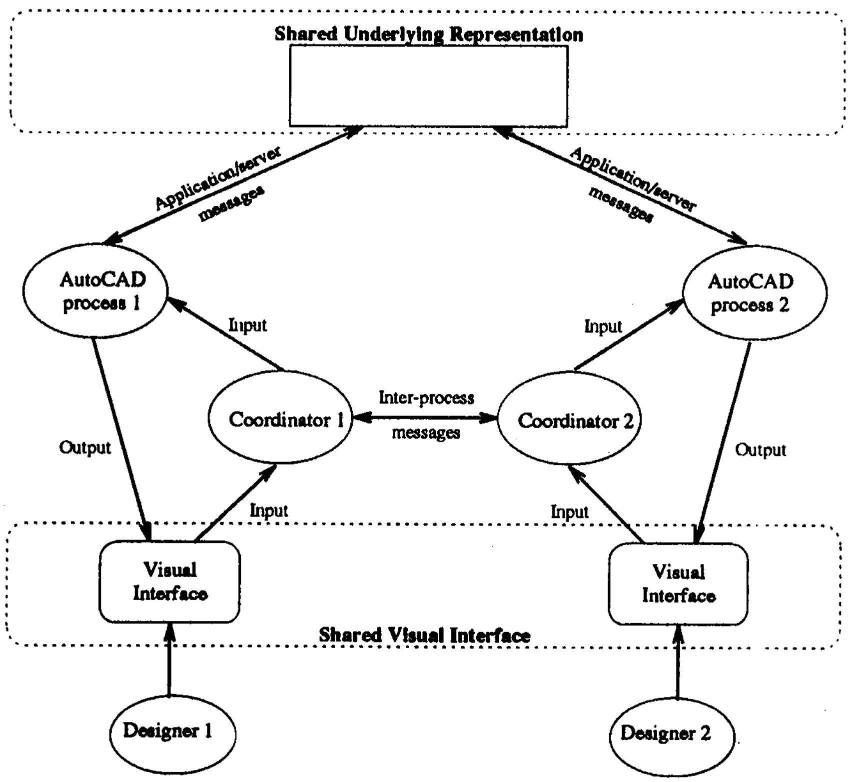


Figure 4: A synchronous multi-user CAD system

As illustrated in Figure 4, a coordinator is inserted between AUTOCAD and the input devices (i.e. keyboard, mouse, etc.). The coordinator translates the event from the input devices into an AUTOCAD command form which in turn executes it and takes action, it then translates the same event into an independent inter-process communication form and distributes this to the other controllers. When *Designer 1* issues a command through the shared visual interface, ***coordinator 1*** first sends this command to AUTOCAD *process 1*. Second, it distributes network message to the ***coordinator 2*** which in turn filters it to AUTOCAD *process 2*. Both AUTOCAD processes execute the current command and display the result on the designers' workstations.

This mechanism simplifies the process of handling events while maintaining synchronization across multiple copies of the application.

### 3.1.2 Implications

The current system has shown how a synchronous multi-user CAD system can be developed from a single user CAD system. It allows designers to create and manipulate the visual data dynamically. The system also allows designers to share a design schema through a common representation. The implication of the implemented system is to provide a basis for moving towards providing richer representations. While the current technology offers support for synchronous collaboration, the use of an existing, commercial, single-user CAD system has implementation problems concerning the development of group interaction. These include: concurrency control and functional features in CAD databases. To maintain the synchronous collaboration among distributed designers, designers must interact with the CAD system through the coordinator, in which a limited number of commands will be supported. When functional features can be embedded within the graphical objects, this allows us to model the structure and the behaviour of design elements without dealing with the actual command that the CAD system provides. Additional issues have also emerged such as the model and process that support multiple interpretation of visual images, shape emergence and the development of a multi-user interface for group CAD systems.

## 3.2 Shared Underlying Representation

The shared underlying representation is the data stored for both persistent memory and use by the CAD system to display and manipulate the visual representation. We discuss the underlying representation in two parts: the geometric information for display and manipulation of visual images on the screen, i.e. the visual data, and the design information for reasoning about the relationships between function, form and behaviour of the images being manipulated, i.e. the design schemas.

### 3.2.1 Representation of Visual Data

The representation of visual data provides the basis for display and manipulation of visual images. In current CAD systems the representation is stored in persistent memory as files. Those CAD systems that use a data exchange standard store the data in files with a standard format, for example, AutoCAD stores the visual data in a DXF format. Also many facilities have been embedded in most CAD systems to manage graphic information. For instance, symbols and layers are used to describe and store a user-defined object. An object may have multiple sets of attributes that can be used to support some form of analysis. The limitation introduced in this representation is that the geometric data is stored according to the primitives used to create and manipulate the visual image. Those support systems does not support: the communication of design ideas, the representation of design intent, and the recognition or manipulation of an emergent form. Those are some of the necessary features that support collaboration between designers and their machines.

### 3.2.2 Representation of Design Schemas

The representation of design schemas defines a space of design elements in the shared workspace. These elements represent the design entities that are shared by the designers in their collaborative session by including information about function, structure (geometry and topology), and behaviour. The entities can be classified into types of design elements, where general information can be represented. The types then provide information related to the design domain, such as floor types used in building design, and the entities represent the information specific to a particular project, such as the service core design elements in the St George Hotel. The integration of specialized and generalized information is illustrated schematically in Figure 5.

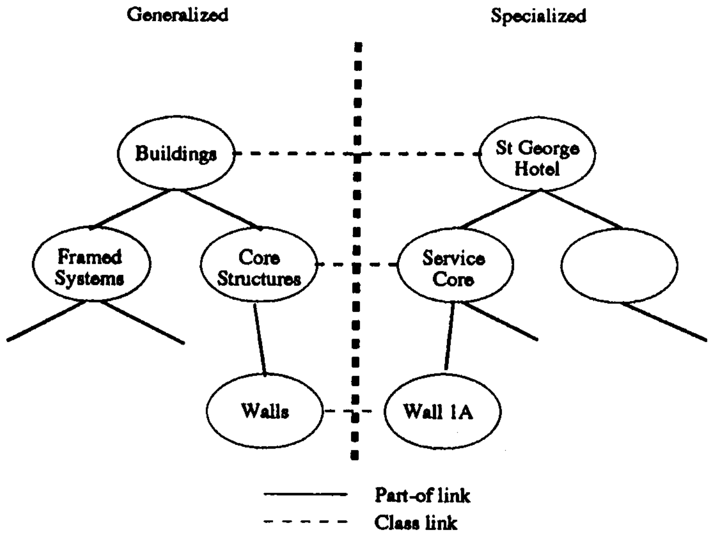


Figure 5: A model for a representation of design schemas

An object-oriented approach is adopted in our model as it provides the semantic expressiveness needed for representing both generalised and specific design knowledge, where the class-instance relationship is made explicit. Objects can also be organised into classes according to their properties, which defines the function, structure and behaviour of all objects that belong to the class. Therefore, the representation of design schemas consists of a set of objects which exist in a knowledge base and a set of objects which are created for a specific project by the session server and each coordinator. These objects are linked to the visual data in order to maintain a consistency between the graphical images being manipulated on the screen and the design schemas associated with each visual image.

A schema similar to the design prototype as defined in Gero (1990) is used to organise the classes and instances of design elements in the underlying representation. A design prototype represents a class of design elements according to the function, behaviour, and structure of the class. The use of design prototypes permits manipulation of design information, i.e. data and knowledge, at multiple levels of abstraction.

### 3.3 A Layered Representation

In order to understand and support design activity as a collaborative process, it is essential that explicit design semantics be represented and accessible as well as the graphical data. A layered representation for coupling the visual data with the semantic design knowledge is proposed. For example, to support emergent form, an alternative representation based on infinite maximal lines (Gero and Yan 1992) could be used as the shared representation, and the graphic data used by the CAD system, such as a DXF file, as the representation associated with one designers view of the visual image. Figure 6(a) introduces two levels of visual data representation, one where the visual data is not associated with the primitives used to create or manipulate the image, the infinite maximal lines level, and the other level represents an aggregation of visual data into primitives for manipulation, the DXF level. The same layered architecture can be used to couple the mass architecture as the basic primitives used for building design with the solid modeller as the tool for visualising and manipulating the visual data (Figure 6(b).

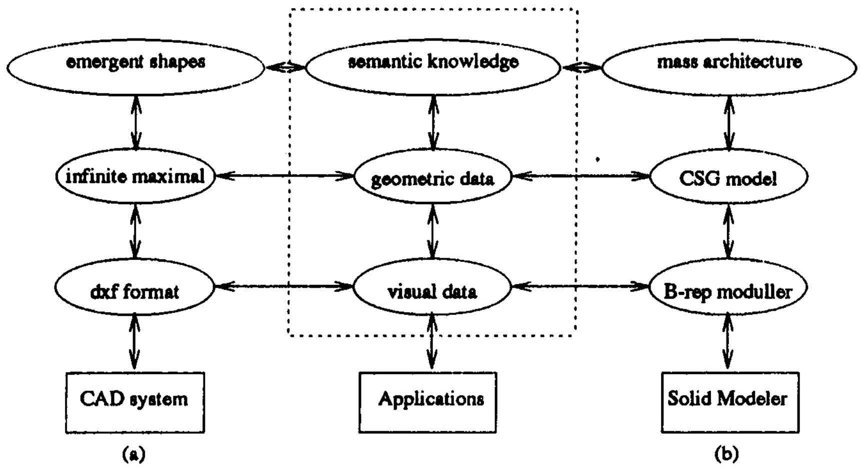


Figure 6: Layered structure for supporting multiple representation.

Another aspect of layered representation is the ability to represent facts about design object depicted by the visual system, rather than the visual data itself (i.e. drawings). The data structures of graphical information are linked to design knowledge. For example, architect may build a description of a house, consisting of a number of spaces that represent an abstract or conceptual unit, which in turn consist of a number of solids that represent the visual units. This structure provides the opportunity for representing and communicating design ideas. Designers will be able interact with and build on design ideas created by others.

### 3.4 A knowledge-based representation

To be useful for collaborative design applications, both non-geometric properties and the relationships between design elements must be modelled. The knowledge-base is the system software component making it possible to construct such a conceptual model for design. In this implementation, the design objects are modelled in the context of mass architecture. Mass architecture is closely related to the definition of interior spaces and their relationships within the building environment. Figure 7 shows an example of a hierarchy of design classes.

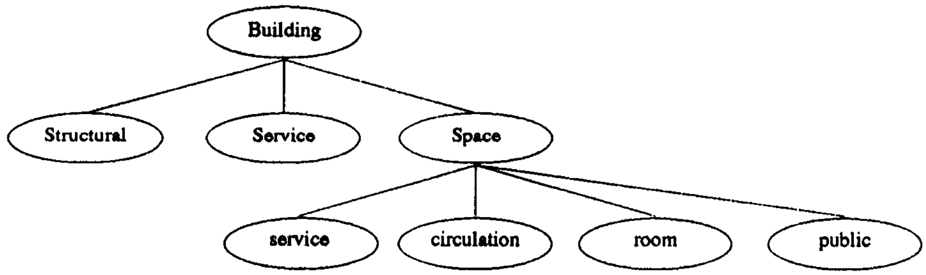


Figure 7: Design classes hierarchy: building spaces

Each class is characterized by the following properties:

**administrative attributes** Representation of objects' identity and integrity such as timestamp, version, consistency checking.

**functional attributes** Representation of the use or purpose of an object, the explicit representation of design intent.

**geometrical attributes** Representation of geometry and its associated features such as shape, topology, enclosure, relationships.

**display attributes** Application specific or graphical-based information for visualising design objects.

Figure 8 shows a class definition of a room.

This formal representation provides a basis for supporting team design because it integrates function, management, geometry, and display. When instantiating a specific design from a design class, design intent will be, by default, the values of the functional attributes. These values can be changed or augmented by designers. The links between function and display attributes will facilitate understanding and collaboration of design process.

## 3.5 Implications

This model provides a useful computer-based framework that supports persistent memory for synchronous collaborative design environment. The advantages of using such framework is with the open architecture which made it possible to represent design concepts with the same underlying structure. Since design concepts are stored within the persistent memory as homogeneous objects, new interface tools of the shared workspace can be easily added to cope with the evolving technology. While the knowledge-based approach seems efficient for maintaining the semantics of the design, it is unlikely to be as efficient in advanced graphical processing as a CAD system is. The need to maintain the semantic integrity of the design must be supported. Mapping between semantics and graphics is clearly important. The current research in design data exchange standards seems relevant toward integrating both representations. Another point that must be considered in a synchronous collaborative design environment is the version management which enhances parallelism of design as well as capturing design rational through recording of design activity.

**ROOM: is-a SPACE**
**Administrative Attributes**
modified-at:
modified-by:
version-number:
shadow-number:
instantiation-methods:
access-methods:
notify-methods:
**Functional Attributes**
types: {family, bed, dining, kitchen, ...}
activity-types: {reception, sleeping, food-preparation, general-utilities, ...}
**Geometric Attributes**
types: {primitive, composite, emergent, ...}
enclosure-type: {walls, glass, roof truss, ...}
topological-relations: CSG tree
boundary-list: {solid, face, edge}
shape: shape-id
position: X Y Z
dimensions: Width Length Height
orientation: Rx Ry Rz
room-area: range [1 to 30sq-m]
room-volume: range [2.4 to 150cb-m]
**Display Attributes**
representation-methods: {B-rep, ...}
display-methods: {wire-frame, surfaces, contours}
display-list: {vertices, coordinates-values}

Figure 8: Representation of building objects as Spaces

# 4 Conclusions

Research and developments in CAD, AI in design, CSCW, and design emergence provide the basis for a multi-user model of a collaborative design environment where a shared workspace is provided to support synchronous design team activity. While the shared visual representation provides the basis for visualising design elements, the shared underlying representation provides a persistent memory of design information, ideas, and intents. Various technologies are now available to explore the implementation and implications of computer-supported collaborative design. While the technology is now accessible and becoming more affordable, how this technology is applied to support synchronous collaborative design is not well developed. Even though it is possible now for a design office to implement networked graphic workstations, within a single location and across cities, the nature of the communication across the network is currently limited to predetermined graphic and non-graphic files and distributed databases. These forms of communication do not occur synchronously. Also, shared drawing surfaces are becoming possible. Currently, such systems do not support persistent memory and do not provide the visualisation techniques in current CAD systems. How shared drawing surfaces can be extended to accommodate emergent form is still under development.

## References


Bannon, L. and Schmidt, K. (1991). CSCW: Four Characters in Search of a Context, *in* J. Bowers and S. Benford (eds), *Studies in Computer Supported Cooperative Work: Theory, Practice and Design*, Elsevier Science, Amsterdam, pp. 3–16.

Benford, S. (1991). Requirements of Activity Management, *in* J. Bowers and S. Benford (eds), *Studies in Computer Supported Cooperative Work: Theory, Practice and Design*, Elsevier Science, Amsterdam, pp. 285–297.

Bond, A. and Gasser, L. (1988). An Analysis of Problems and Research in DAI, *in* A. Bond and L. Gasser (eds), *Reading in Distributed Artificial Intelligence*, Morgan Kaufmann, CA, pp. 3–35.

Bowers, J. and Benford, S. (eds) (1991). *Studies in Computer Supported Cooperative Work: Theory, Practice and Design*, Elsevier Science, Amsterdam.

Crowley, T., Milazoo, P., Baker, E., Forsdick, H. and Tomlinson, R. (1990). MMConf: An Infrastructure for Building Shared Multimedia Applications, *in* ACM (ed.), *Proceedings of the Conference on Computer-Supported Cooperative Work*, ACM, New York, pp. 329–342.

De La Garza, J. M. and Oralkan, G. M. (1992). An object space framework for design/construction integration, *Building and Environment* **27**(2): 243–255.

DePaoli, F. and Tisato, F. (1991). A Model for Real-Time Co-operation, *in* L. Bannon, M. Robinson and K. Schnidt (eds), *Proceedings of the Second European Conference on Computer-Supported Cooperative Work*, Kluwer Academic, Dordrecht, pp. 203–217.

Dourish, P. (1992). *shdr: a shared drawing program*, Rank Xerox EuroPARC, UK.

Fenves, S., Flemming, U., Hendrickson, C., Maher, M. and Schmitt, G. (1990). Integrated Software Environment for Building Design and Construction, *Computer Aided Design* **22**(1): 27–36.

Ganeshan, R., Finger, S. and Garrett, J. (1991). Representing and reasoning with design intent, *in* J. S. Gero (ed.), *Artificial Intelligence in Design '91*, Butterworth-Heinemann, Oxford, pp. 737–755.

Gero, J. S. (1990). Design Prototypes: a Knowledge Representation Schema for Design, *Artificial Intelligence Magazine* **11**(4): 26–36.

Gero, J. S. (1992). Creativity, Emergence and Evolution in Design, *in* J. S. Gero and F. Sudweeks (eds), *Preprints Conference on Computational Models of Creative Design*, Design Computing Unit, University of Sydney, pp. 1–28.

Gero, J. S. and Yan, M. (1992). Discovering Emergent Shapes using a Data-Driven Symbolic Model, *Technical report*, Design Computing Unit, University of Sydney, Australia. Submitted to CAAD Futures '93.

Hennessy, P. (1991). Information domains in CSCW, *in* J. Bowers and S. Benford (eds), *Studies in Computer Supported Cooperative Work: Theory, Practice and Design*, Elsevier Science, Amsterdam, pp. 299–311.

Ishii, H. and Miyake, N. (1991). Toward an Open Shared Workspace: Computer and Video Fusion Approach of Teamworkstation, *Communication of the ACM* **34**(12): 37–50.

Law, K. H., Wiederhold, G., Siambela, N., Sujansky, W., Zingmond, D. and Harvinder, S. (1991). Architecture for Managing Design Objects in a shareable Relational Framework, *International Journal of Systems Automation: Research and Applications* **1**: 47–65.

Lee, J. (1990). SIBYL: A Tool for Managing Group Decision Rational, *in* ACM (ed.), *Proceedings of the Conference on Computer-Supported Cooperative Work*, ACM, New York, pp. 79–92.

MacLean, A., Bellotti, V., Young, R. and Moran, T. (1991). Reaching through Analogy: A Design Rational Perspective on Roles of Analogy, *in* S. P. Roberston, G. M. Olson and J. S. Olson (eds), *Proceedings of the Conference on Computer-Supported Cooperative Work*, ACM, New York, pp. 167–172.

Maher, M. L., Gero, J. S. and Saad, M. (1993). Synchronous support and emergence in collaborative CAAD, *Technical report*, Design Computing Unit, University of Sydney, Australia. Submitted to CAAD Futures '93.

Mitchell, W. (1989). Creativity, Emergence and Evolution in Design, *Preprints Modeling Creativity and Knowledge-Based Creative Design*, Design Computing Unit, University of Sydney, Australia, pp. 263–285.

Pankoke-Babatz, U. (ed.) (1989). *Computer Based Group Communication: The AMIGO Activity Model*, Ellis Horwood, England.

Patterson, J. F. (1990). Rendezvous: An Architecture for Synchronous Multi-User Applications, *in* ACM (ed.), *Proceedings of the Conference on Computer-Supported Cooperative Work*, ACM, New York, pp. 317–328.

Poyet, P., Dubois, A. and Delcambre, B. (1990). Artificial Intelligent Software Engineering in Building Engineering, *Microcomputer in Civil Engineering* **5**(3): 167–205.

Rosenman, M. and Maher, M. L. (1992). Formal representations for communicating design intent, *Technical report*, Key Centre for Design Quality, University of Sydney, Australia.

Sauce, R. and Powell, G. H. (1992). Object-oriented approaches for integrated engineering design systems, *Computing in Civil Engineering* **6**(3): 248–265.

Smith, H., Hennessy, P. and Lunt, G. (1991). An Object-Oriented Framework for Modelling Organisation Communication, *in* J. Bowers and S. Benford (eds), *Studies in Computer Supported Cooperative Work: Theory, Practice and Design*, Elsevier Science, Amsterdam, pp. 145–158.

Sriram, D., Wong, A. and Logcher, R. (1991). Shared Workspaces in Computer Aided Collaborative Product Development, *Technical report*, Intelligent Engineering Systems Laboratory, MIT, MA.