A Study on the Distributed Collaborative Architectural Design System

Seung-Hoon Han
Senior Researcher, Convergence Laboratory, KT, Korea

Abstract
The computing resources of an organization or project team were often spread across many different platforms and in different locations. Recently, there has been a growing interest in distributed CAAD (Computer-Aided Architectural Design) integration due to the needs of direct collaboration among project participants. The potential for the integration of information is expected to have a tremendous impact on architecture and the construction industry. The aim of this research is to provide a new paradigm for a CAAD system by combining research on integrated CAAD applications with recent collaboration technologies. To accomplish this research objective, the proposed system has been designed and a prototype implemented to produce enough guidelines to foster interest in the development of future CAAD systems on the Internet. To this end, two different scopes of implementation are evaluated: first, global architecture and the functionality of a distributed CAAD system; and, second, the association of an architectural application to the system.

Keywords: collaborative design, distributed collaboration, design process, integration, Internet

1. Introduction
The needs of collaboration among design participants spread in different locations is emphasized in design stages in order to not only save time and provide places to meet and talk, but also to save cost for those events as well: This is being realized by the Internet, which helps support a networked, integrated real-time multi-user environment. As the base for collaboration activities moves from physical places to cyberspace, the methods of connecting every participant by means of a computer technology have been desired and considered.

The increased complexity in technical support systems and the fragmentation resulting from the addition of many more individuals to a building project have made the design process far too complicated. In addition, the lack of collaboration in knowledge or information representation strategies has made the communication during the design and construction phases more difficult.

Efforts have been made to encourage the development of collaborative design systems to enable their use in the design phases. The Internet is a solution to allow the design systems to be used simultaneously with various interactions, and ICT (Information and Communication Technology) has been an essential concept for supporting collaboration in architectural design.

Internet is becoming the optimal tool for collaboration among participants in architectural design and construction projects because of the low connection costs and wide availability. Such collaborations will include the exchange of project drawings and various forms of project materials and general distribution of project information through the Internet. One way or another, the existence of the Internet and the wealth of related technology will change the way architectural design and construction are practiced today.

Distributed object computing has the potential to change the information landscape of a broad range of business practices. As integrated computer systems offer the capability to improve the effectiveness and efficiency of management processes in practice, their use is likely to increase the information flow and the quality of communication among project participants in the collaborative design process.

This study will investigate how architects can collaborate using distributed technology, especially in the early design stages which mainly involve their cognitive work, and aims to gain insight into the advantages and shortcomings of such an approach. From this insight, it is envisioned that specific recommendations will be made for future development of distributed collaborative architectural design systems. Through this paper, the architecture of a distributed collaborative architectural design system is investigated and some experiments that examine design workflow tasks performed within the environment are presented. An application scenario offers an alternative for design evaluation practice using the proposed system. The scenario will also include a demonstration of virtual settings and scenes that may be cooperative.
places for remote participants, such as planners, designers, architects or engineers discussing possible alterations to designs.

2. Towards Distributed Collaboration

In 1963, Steven Coons wrote a visionary paper titled "An Outline of The Requirements for a Computer-Aided Design System" which suggested two main trends that a CAD application would face in order to support various design tasks, and set the research agenda in CAD for the next generation.

Currently, CAAD systems have adequately satisfied several demands. They have dramatically improved the accuracy and consistency of working drawings, enabled designers to visualize their design ideas in three-dimensions, allowed the analysis of designs through data exchange and integrated databases, and even allowed the designers to evaluate designs based on comparisons to previous cases and/or the formalization of grammars.

There is, however, a consensus that CAAD systems have not yet achieved their full potential. First, most systems employ a single-user approach to solving architectural problems which fails to grapple with the fact that most design work is done through teamwork. Second, current systems still cannot support early design stages which involve client briefing, data collection, building program formulation, and schematic design generation.

The focus on collaboration is due to recent technological advances, both on the hardware and software sides, which, coupled with a maturity in the knowledge-base of group behavior and dynamics, have made it possible to support and enhance collaborative processes through technological interventions (Jabi, 1996).

Coons, first of all, emphasized the role of Human-Computer Interaction (HCI) with which the future CAAD systems would be pursued. Although there have been many research projects in HCI to improve communication between human and systems like CAAD systems, one of the current issues in interface design is how to manipulate interaction with non-geometric data. Also, the rapid growth of the Web and its business and academy-oriented opportunities portend a new need for information interaction and visualization.

An architect uses non-geometric data in making design decisions both for administrative purposes and to manage the design process. In practice, a design is produced with synthesis and analysis of information. Such information can range from site data to legislation, standards and product information. Because most of this architectural information pertains to a building model, information about architectural design needs to be related to the building model as a core model. Additionally, these building model data need to follow commonly accepted exchange standards to be adapted to systems in other firms as well as to be reused in future projects with flexibility and generality that Coons emphasized in his paper: "The computer should also be able to furnish information about standard processes" (Coons, 1963).

The building design process has changed significantly in the last years. Generally, it is a matter of fact that the technological developments in every field of science have an influence on the society and therefore on the design and the design process itself. Architectural specialists are considering especially for the influence of the rapid developments of ICT in architectural design (Sariyildiz et al., 1997).

The Internet has evolved as an excellent resource for the AEC (Architecture, Engineering, and Construction) disciplines, as it allows quick, efficient, and widespread communication to those who can access it, sharing everything from design information to project participant communication. Companies who previously marketed and sold CAAD products are now diversifying and offering services and other resources related to all aspects of the design industry.

Just as the earliest CAAD applications were relatively unsophisticated in their capabilities of making the drafting process of designers easier, these online services are currently in an early, formative period. The CAAD and AEC industries are relatively beginner on the Internet and so such services have strong as well as weak points. Already established CAAD companies enhanced the features of their offerings with innovation coupled with know-how, while newly joined enterprises to the industry realize of the needs of the industry viewed from new perspectives. Considering the consummate growth and widespread utilization by the AEC industry, the Internet and these Web-based services will be the greatest area of growth and development in the CAAD industry (Park, 2001).

Another new strategy for collaboration is proposed to empower designers in the architectural field with an innovative process, which comes from the utilization of distributed computing based on the Object-Oriented (OO) approach. OO design applied to CAAD development lends favorably to the expected nature of distributed objects, which can considerably cut down decision-making procedures by providing cooperation between them; developments in CAAD technology has led to modular objects and eventually to their distribution. Distributed technology allows the designer to extract valuable information associated with the objects distributed online, not only values such as simple dimensions, but also other user-defined values from which reasonable updates and modification can be made.

Web services are appearing that cater to the AEC industry's need to collaborate efficiently and methods
of implementing Web-enabled collaboration are arising. Recent peer-to-peer, distributed approaches are becoming a major trend of collaboration, although they have not been commercialized in the architectural profession yet. This approach provides a basis for all work to be done, concerning everything from project information to application without having to worry about obsolete or non-common hardware, software or unneeded personnel.

Concepts and tools such as HCI, Data Exchange Standards, OO Programming and Web technology have all emerged from work on conceptual data models and network computing, and are apt to foster the development of a new paradigm that will enable researchers to take a new approach to CAAD. Indeed, the development of CAAD software applications, the development of new modeling methodologies and the definition of standards for information exchange create opportunities for achieving distributed system integration.

Therefore, the opportunity is seen to implement a solution which will provide both objects of basic usability to designers and the ready accessibility of those objects in the form of programmed applications over the Web, and will thus be manifested in the CAAD-enabled distributed system.

3. Proposed System Architecture

This paper is concerned with the design and realization of distributed collaborative virtual environment using Distributed Multi-User Technologies (DMUTech), a working prototype of 3D computer-generated design environment, which actively supports collaboration between distributed participants. The approach taken in this system reflects both the management of interpersonal communication and the utilization of connection in distributed systems.

Consequently, a distributed collaborative design system, named ARCH:DMUVR here, to support more efficient communication and interaction among multiple participants in the design process, is proposed. The scope of development and functions of this application is narrowed to focus on its major contribution to the distributed system - a tool to test the integration performance of the proposed system.

The role of the proposed system here is to help architects make better design decisions with real-time presentation, communication, collaboration, feedback, and evaluation, especially in geometric aspects of the building design. The following key characteristics of the distributed systems have been reviewed and prototyped in terms of computation to gain successful implementation of the proposed distributed system:

1) Presentation: There is a notional world or space presenting design proposals, which is the virtual environment, generated from the core design model, activated as the abstract model, and visualized by VRML plug-in for the Internet browsers.

2) Representation: Every client is represented or embodied within the virtual environment using avatars and is visualized to other users by means of this embodiment to enhance access and comprehension. Each user is autonomous and able to move independently around the virtual environment. Building design components which are objects in the virtual environment are extracted from the scene-graph, and represented by a separate Graphical User Interface (GUI) called 3D Building Object Editor for the system.

3) Communication and Collaboration: Participants can communicate and collaborate in many different ways through different communication channels within the computational and networking domains. A chatting window, for example, is provided for textual intercommunication using a specific communication channel, and a 3D Building Object Editor is used for distributed collaborative design among distributed participants through a different communication channel.

4) Decision-Making: The proposed system is also concerned with improving the support for collaborative decision-making based on observations of critical issues in agreement and negotiation drawn from the discipline of social science.

When a participant in the design team attempts to update a design proposal, for instance, an agreement must be reached from all other connecting users via a networked agreement procedure in the system.

5) Evaluation: The most important concept for evaluation is observations of and reflections on the effectiveness and shortcomings of the distribution aspects of the developed system. A dynamic Web-based system evaluating application is implemented for the purpose of those observations, and its data are reviewed by experts for future development.

Since there is diversity in realizing communication between different design processes, such as between different participants who have different knowledge-bases, the main communication option is direct peer-to-peer communication using CORBA (Common Object Request Broker Architecture). In addition, the central server which already controlled a few other communication channels cannot become a bottleneck in this way.

For distributed virtual environments (DVE), objects have a graphical representation (scene-graph), an internal state and a behavior usually defined by program code. Such objects have to be added or removed from a scene in real time, their behavior has to be tracked in real time and their implementation has to be distributed immediately on different computers in the network. Some objects can be controlled by other objects and they should be able to share information.
Further requirements result from security considerations, e.g., protecting a scene from vandalism. An object can grant or deny access to its data and behaviors. For this purpose, it could provide user rights. In a process called authentication, the identity of users and their objects has to be checked. When several objects try to access a shared resource, a conflict can arise. Such conflicts can be avoided through transaction mechanisms or they can be resolved by conflict solution strategies towards decision-making (Diehl, 2001).

A VRML browser usually allows two primitive network operations: hyper-links and inclusion of media stored on different servers in the network. DMUTech is used for all aspects of network communication in multi-user worlds which have not been provided by the VRML browser (Roehl et al., 1997).

The VRML browser and the Java applets can communicate via the EAI (External Authoring Interface). But, for the transmission of time-uncritical, large-scaled messages among browsers, it would be better to use CORBA and its network protocol, IIOP (Internet Inter-ORB Protocol). These browsers are now capable of communicating through an Object Request Broker. The programmer no longer needs to write code for sending messages to other hosts, but simply calls methods of the objects, which actually exist at other hosts where the methods get executed.

For textual interaction between browsers, which is actually interacted by participants, IIOP is too slow, and TCP (Transfer Control Protocol) is more appropriate for their communication.

4. System Implementation and Experiment
ARCH:DMUVR can be started at two different levels; one is an expert system component accessed by designers who need to perform an initial task of building site analysis and selection. The scope of development and functions of this GIS knowledge-based application will be narrowed to focus on its major contribution to the distributed system to test accessibility and usability of the prototype, as not only the entry level to be integrated into the main system, but also as a stand-alone, specific-purpose application.

The other access level is the main entrance to ARCH:DMUVR for all other normal participants using the Design Visualization Interface. This interface is the way in which ARCH:DMUVR is visualized to all connecting participants. It is anticipated that the system will normally be used with 3D graphics, and this is basically reflected in the design of a 3D GUI and has been supported by VRML browsers as shown in Fig.2. At the top of this main window is the 3D graphical view of the building design visualized by CosmoPlayer. An abstract VRML model is launched into the interface for the previously selected building model, and clients' avatars are bound to the built VR environment. This shows a view into the virtual world as the graphical client of ARCH:DMUVR. At the bottom of the window is the communication interface which makes it possible for users to communicate with each other about the building model through textual.
exchange and share information. The user connecting to the system requires an authentication.

The Design Visualization Interface interacts with a Java application which provides tools to connect the authoring functions for the built VR environment, show the world information such as building layer and user connection properties, and link supplementary features of design evaluation applications.

Specific building component data to be used in the application server can be obtained either continuously from the client's local computer using middleware via IIOP, as long as the connection with the client is maintained, or discretely by packages of data through the database repository at reasonable time intervals. While connected to the server application, the client can receive real-time mapping results from the server.

ARCH:DMUVR has a feature to bring data into the 3D Design Object Editor, which is a GUI support of the collaborative design environment; that is, CoDesign on the menu bar of the initial ARCH:DMUVR system calls the Application Launcher, which shows available applications distributed on the Internet and allows the user to launch those applications.

Whenever an application is requested from the user by choosing an item from the Application Launcher, it searches the Internet using the naming service provided by CORBA, and connects to the local computer which owns relevant applications and data. All loaded geometric data from the local computer can be mapped to the 3D Design Object Editor, and transformed to 3D graphic components. Modified data using this editor are passed to the Design Visualization Interface, examined above, for a VR presentation, and dispatched to all participating users for collaboration.

In multi-user environments, avatars play an important role as the virtual representation of a user. It is located at the viewpoint of the representing user from which she or he looks at the scene. The shape of the avatar determines how the user is seen by other users. If a user navigates through the scene and moves the viewpoint, the avatar also moves in the views of the other users.

Clients are able to participate in the collaborative virtual environment and interact with other clients through avatars. In decision-making stages, the role of the avatars is more important. For example, if a conflict occurs between different experts during the decision-making process in adjusting the dimension of the building component in a design proposal, and if they want to investigate the target component at the same viewpoint, they are easily able to move to the point of interest by recognizing the other avatars in the scene. The position of the moving avatar is immediately notified to all other avatars, and its changes are instantly reflected in the scene. This way, one avatar who suggests the meeting at a specific location in the scene, can move first to the spot and lead other avatars to the same spot; this will help the users collaborate smoothly and make better decisions.

On the other hand, 3D geometric building data transferred from the client's local database are mapped to the 3D Design Object Editor and visualized in the Design Visualization Interface. The Design Object Editor is a specific application that is owned by a client and distributed on the Internet. Currently, ARCH:DMUVR can connect six different distributed applications and databases categorized by building components, such as slabs, columns, walls, windows, doors, and lighting; it is assumed that each application belongs to a different knowledge-based client. Those applications control the components of the scene-graph in the Design Visualization Interface, and linkage between the two interfaces is maintained by the application server, which is an owner of the application.

This 3D GUI is the way in which design proposals are authorized in 3D graphics. The building objects can
be drawn using 3D graphic algorithms at the central canvas, and can be modified by altering the properties at the right-side in this interface. Altered data using this editor are transferred to, not only the central server to dispatch them to all clients for their modified VR visualization, but also to the application server from which the 3D Design Object Editor was originally launched, in order to pass those data to other clients' Design Object Editor interfaces and to change their drawings simultaneously. The current version of the 3D Design Object Editor supports collaboration only in geometric aspects of the building design due to limitation of the building property data.

Supplemental functions for design manipulation are also provided at the bottom of the interface. Such tools are not for collaboration, but for personal operations between current single user's 3D Object Editor and the VR Visualization Interface. Those actions are network-independent and not broadcast to other clients. For example, as shown in Fig.4., when a user wants to remove the upper slab from the scene for a better view, this action can be requested at the 3D Object Editor, and the results are shown directly in his visualization interface. The axes on the canvas can also be toggled individually.

One of important requirements for operating ARCH:DMUVR is the security consideration. During collaboration, the scene-graph must be protected all the time from any possible bad behaviors or accidental changes. For this purpose, at the middle of the design update procedure is the Agreement Interface. This interface plays an important role in mediation of different opinions arising from the decision-making process. When a client attempts to update a design proposal with the 3D Object Editor, this action is immediately notified to all other clients who can grant or deny access to data and its behavior by responding to the Agreement Interface.

The Agreement interface uses the voting system, although the rule of getting agreement can be considered in various forms. Users' responses are anonymously collected in the server machine, and the server returns the voting result to all clients. Once the applicant for the design update acquires the votes for agreement from half or more of the respondents, the design proposal can be modified. This networked agreement procedure will help the design participants collaborate in a reasonable, positive manner.

The participants in the system not only can communicate through the interface and leave their annotations, but can construct their evaluation criteria for major characteristics of a building presentation which is the most crucial aspect of a collaborative environment.

Evaluations consist of scoring evaluation criteria associated with a score value, and leaving a comment for each criterion in a text field. Scores and comments entered in the form are sent to the server-side and saved in the system database with a client ID and a date written. A later appendix from other clients can be filled out through the same interface and added immediately to the database as new records. Then, all evaluations stored will be retrieved and displayed on the result form in chronological order. This evaluation

Fig.4. Interaction between 3D Object Editor and VR Browser

Fig.5. Flow of Design Update with Agreement Interface
result will allow the clients to review the building presentation and the design objects (Han and Turner, 2001).

Many experts in the architectural professions including designers, scholars, doctoral students, project managers, and architectural technicians, have experimented with this prototypical system that presents the possibility of a distributed system for architectural purpose. It has, as the result, been proven that the distributed object approach to integration has benefits when we consider the current technological and economic state of architectural collaboration.

Instead of integration being achieved through static models that define the structure of shared information, the collaboration models were able to be distributed through a network to be easily accessed and modified from multiple users in different locations.

While experimenting and simulating ARCH:DMUVR in collaboration with various professions in the architectural field, the following have been observed:

1) ARCH:DMUVR is a prototypical product of the distributed system model. As such I think it has been successful. It has provided a demonstration platform to explore the concepts of collaborative design.

2) A framework of distributed object computing environments using CAAD and GIS system is usable, since it has been helpful for holding distributed meetings in diversity of professionals over networks.

3) The current focus on peer-to-peer communication, based on sharing computation with other machines in the network, is promising. It has been particularly satisfying to see other experts deriving new insight from its use and enjoying involvement in the virtual design process with the system.

4) The current version of ARCH:DMUVR is a minimal prototype implementation to check the usability of the system in collaboration phases in the design process. A fully functioning prototype needs to be built to test the applicability to the actual AEC industries.

Especially, most participants in the experiments have emphasized the needs of evaluation tools for quantitative variables in the building model. The other application is, consequently, being developed for evaluating quantitative factors that are relative to building performance for the modified design proposals. This kind of evaluation uses objective variables and rules for measuring building performance, and contains many arithmetic operations.

Design Evaluating Applications for this purpose are built as distributed applications, which are located at and called from distributed experts’ computer systems connected to the Internet. CORBA technology is applied to the implementation of this evaluator integrated in ARCH:DMUVR; The Design Evaluating Application remains ongoing and will be the subject of future work.

The role of the Design Evaluating Application in the system is to help architects make better design decisions with immediate feedback. It is proposed as a discrete application, which will run on a remote server and be executable on the Internet. This results from the assumption that in the future, bigger expert applications will be built on a server domain and will provide evaluation results to clients.

As a part of ARCH:DMUVR, this application should satisfy the following criteria:

1) Immediate feedback needs to be served. As an end-user interface, the system needs to have immediate feedback and wait for input from the clients before moving to the next stage in the design process. The Internet-based application receives input from a local database system and passes the result either back to the database or directly to ARCH:DMUVR. When the result is not a final one and needs to be approved by the designer, the direct passing of the result to the main system is more desirable, and eliminates unnecessary data storage in the DBMS. Otherwise, results go to the local databases first, and then are passed to other system components through the connection mechanism, hence improving the system integrity.

2) The internal evaluating process needs to be both hidden and transparent. Although in most of the cases, the data flow and computational intermediate steps need to be hidden from the user interface (to avoid complexity), there might be some cases where architects need to request detailed-computation information for deeper analysis. For those cases, the optional transparency of the application process should be provided to the end-user in a readable manner.

3) An application viewer needs to be set up as a communication interface between application service expert and the online client. Along with the information architects get from the CAAD model by direct mapping from the server application, this viewer will provide any relevant extra information available from the remote application experts.

Fig.6. Variables for Evaluation
5. Conclusion and Discussion

This paper is concerned with the design and realization of a distributed architectural collaborative design system, ARCH:DMUVR, which seeks to actively support communication and collaboration between distributed users. Through this paper, the following concerns are addressed to help architects, researchers, and other related professionals working on multi-user cooperative virtual reality for architectural design:

1) Architectures for Distributed Virtual Environments (DVE);
2) Network requirements of those systems;
3) Implementation and use of collaborative Virtual Reality (VR) system; and
4) Models of interaction and communication in distributed collaborative architectural design systems.

ARCH:DMUVR realizes and extends a geometric model of interaction providing a conceptual and computational basis for managing interactive communication and collaboration. Three major perspectives have been combined for this study:

1) The user's perspective within the everyday social world: considering the kinds of interaction and control that are agreeable to all participants in the system;
2) The machine's perspective: considering the implementation and management of the system that are affordable to all currently available hardware and software settings to support this interaction; and
3) The network's perspective: considering the form and requirements of the communication that facilitates the interaction.

ARCH:DMUVR was designed specifically to be a distributed collaborative architectural design system, and some of its key features are:

1) A full implementation, presentation, and representation of the building model and its geometric properties that control all interaction;
2) Rich computer-mediated communication via combinations of text and three-dimensional (3D) graphics;
3) A stable, compatible implementation running on various platform settings; and
4) Support for multiple collaborative environments and their own applications linked via distributed computing protocols.

This paper also proposed that what is needed for effective computer-aided collaborative design is an integrated application environment, using distributed collaboration technology, which is accessible and comprehensible to all the professionals in the building design team, which not only allows the sharing of information but also the sharing of understanding, and which facilitates the development of design tools for different aspects that can be plugged into it, and details the additional solution to a shared building representation for the envisioned collaborative design environment. This approach will promote the use of computers on-site, in new terms of Network-Aided Architectural Design (NAAD).

By using the features of the CORBA distributed computing framework, an efficiently integrated CAAD application system can be built on the Internet. Ultimately, it enables, by experimenting with the prototype distributed collaborative design system, the testing of practical levels of system implementation and functionality to draw potential uses and collaborative tasks in the architectural professions.

References

1) Coons, S. (1963) "An Outline of the Requirements for a Computer-Aided Design System." Proceedings of the Spring Joint Conference. May 1963. Detroit, Michigan, USA.
2) Diehl, S. (2001) "Distributed Virtual Worlds – Foundations and Implementation Techniques Using VRML, Java, and CORBA." Springer. Berlin, Germany.
3) Duerk, D. (1993) "Architectural Programming: Information Management for Design." Van Nostrand Reinhold. New York City, New York, USA.
4) Eastman, C., Jeng, T. and Chowdbury, R. (1997) "Integration of Design Applications with Building Models." in Proceedings of CAAD Futures 1997. Kluwer Academic Publishers. Eindhoven, The Netherlands.
5) Greenhalgh, C. (1999) "Large Scale Collaborative Virtual Environments." Springer. London, Great Britain.
6) Han, S. (2004) "A Working Prototype of Distributed Collaborative Architectural Design System." Ph.D. Dissertation. The University of Michigan. Ann Arbor, Michigan, USA.
7) Han, S. and Turner, J. (2001) "The Architectural Approach to Virtual Reality Support of Multi-user Environments." in Proceedings of CAAD Futures 2001. Kluwer Academic Publishers. Eindhoven, The Netherlands.
8) Jabl, W. (1996) "An Outline of the Requirements for a Computer-Supported Collaborative Design System." in Open House International, vol 21, no 3.
9) Khemlani, L. and Kalay, Y. (1997) "An Integrated Computing Environment for Collaborative, Multi-disciplinary Building Design." in Proceedings of CAAD Futures 1997, Kluwer Academic Publishers. Eindhoven, The Netherlands.
10) Novitaki, B. (1994) "Architect-Client Design Collaboration." in Architecture. June 1994.
11) Park, H. (2001) "Distributed Representation of an Architectural Model." Graduate School of Design. Harvard University. Cambridge, Massachusetts, USA.
12) Sariyildiz, S., Volker, H. and Schwenck, M. (1997) "Improving CAAD by Applying Integrated Design Support Systems and New Design Methodologies." in Proceedings of CAAD Futures 1997. Kluwer Academic Publishers. Eindhoven, The Netherlands.