Breland, J. and Shiratuddin, M.F. (2008) ***A proposed framework for collaborative design in a Virtual Environment.*** **In: International Joint Conferences on Computer, Information, and Systems Sciences, and Engineering (CIS2E 08), 5 - 13 December, Virtual conference.**

International Joint Conferences on Computer, Information, and Systems Sciences, and Engineering (CIS2E 08), December 5 - 13, 2008

A Proposed Framework for Collaborative Design in a Virtual Environment

Jason S. Breland & Mohd Fairuz Shiratuddin
School of Construction, The University of Southern Mississippi, Hattiesburg, MS 39406, USA

***Abstract*-This paper describes a proposed framework for a collaborative design in a virtual environment. The framework consists of components that support a true collaborative design in a real-time 3D virtual environment. In support of the proposed framework, a prototype application is being developed. The authors envision the framework will have, but not limited to the following features: (1) real-time manipulation of 3D objects across the network, (2) support for multi-designer activities and information access, (3) co-existence within same virtual space, etc. This paper also discusses a proposed testing to determine the possible benefits of a collaborative design in a virtual environment over other forms of collaboration, and results from a pilot test.**

Keywords: architecture, collaborative, design, virtual environment

I. INTRODUCTION

A typical construction project requires multi-disciplinary collaboration and expertise of the design team to materialize the owner’s intent and vision into constructible designs. The construction industry is constituted of “high level of complexity, uncertainty, discontinuity, with many restrictions” [1]. Typically, the level of collaboration among design team members varies depending on the complexity of a project, and the organizational arrangement the design team members are bound to.

A construction project is a complex endeavor and its success is most likely only when different professionals collaborate especially from the early stages [3]. During a study of computer mediated architectural design, three forms of collaboration were observed [2]; “mutual collaboration” which involves designers equally working together on the same aspect of the task; “exclusive collaboration” is when designers working on separate aspects of the same problem with occasional time for consultation; and “dictatorial collaboration” when by appointment or naturally, emerge a “designer in charge” who makes all the design decisions.

Collaboration among the design team members is limited due to the nature of current design execution which is linear in nature [4], [5], [6] & [7] and restrictions of 2D design tools used to create and communicate the designs [8], [9] & [10]. Each designer performs his/her own discipline-specific element of the design, then passes the design to the next discipline-specific team member (such as a structural designer’s starts designing after the architectural designer has completed the architectural design of the facility). The final design of a construction project is complete when each team member collaborates in terms of completing their specific task in furnishing every part of the facility’s design.

The computer has been found to play an active role in collaborative design because it provides designers with the visualization support for 3D models, assistance in generating alternative designs, and a platform for design critiques [2]. It can be further argued that early collaboration with the support of computer technology benefits designers and also the consequent processes in construction project designs that are often complex [2],[3]. Computer technology such as virtual environment (VE) provides virtual spaces that can be crafted to support collaborative work and social interplay.

As such, we propose a Collaborative Virtual Environment (CVEs) that enhances collaborative ability for designers, to a more desired level of interaction whereby allowing them to work on the same design within the same space, without interfering any one's assigned task. We define CVE as a VE that supports synchronized multi-participants activities; participants are able to co-exist within the same virtual time and space, and able to see what others are doing in the VE. A CVE that is specifically for architectural design would be a feasible solution to enhance collaboration among designers such as architects and engineers. Not only a CVE allows for real-time 3D viewing and displays of designs, but also permits multiple designers to work on the same design that can be part of a larger project.

We have developed a working prototype i.e. the Collaborative Design Tool in a Virtual Environment (CDT-ve) utilizing the Torque 3D Game Engine from GarageGames. The prototype is capable to stand on its own as a working model. However, there are still a few features which have minor problems, and several new features we would like to add. Our end goal is to provide a robust framework to support architectural design activities. In this paper we describe some of the key components of the framework, a proposed testing procedure and brief results on a pilot study we undertook.

II. MAIN COMPONENTS OF THE FRAMEWORK

Unaltered Torque 3D Game Engine (TGE) uses a fairly straight forward code structure. The computer hosting the VE holds the current environment's data as well as performing collision calculations, and items and characters' tracking. Each visible object within the host's VE is then transmitted to every client's computer as an 'invisible' or 'ghost' object. We have made modifications to the original TGE to allow for real-time collaborative design in a VE.

The TGE supports multiplayer online gaming, thus, the networks speed is always a top priority. The data sent out from the server to clients is optimized to not only reduce the number of packets that need to be sent but also to avoid cheating on the client side. Information could be gleaned from the update messages to give an unfair advantage such as player locations, etc. With a collaborative design environment, the client computers are programmed into having the same information available to them as the host. Object names, properties, and other useful information such as attached data should be sent out to the clients. In addition, the data structures that hold the objects on client computers is different and had to be altered to hold additional information and allow access and manipulation. Fig. 1 shows the changes made to the message system (shown in red). These changes leave the server ultimately in control so that it can ignore changes if necessary as well as simplifying the synchronization process.

The Editor's code was modified as well. Every time an object is manipulated in the Editor, a different command will be executed depending on if the CDT-ve is running as a host or a client. If it is running as a host, the original code usually executes. For every possible manipulation of the VE, new commands had to be written to access the expanded message system. Thus we have almost doubled the Editor's code which handles object manipulation in the VE. For every movement, rotation, scaling, creation, deletion, copying, pasting etc., new code had to be written. Different object types are often handled differently as well as different code for a single or multiple objects. In summary, there exist a myriad of different scenarios when manipulating objects in the Editor. Each scenario had to be independently considered and expanded to include alternate code when the application is running as a client.

The Host and Client Design Update Model

Fig. 1 also shows the additions and alterations made to the prototype CDT-ve (shown in red). Clients can now enter the real-time design environment. As clients create, manipulate, or delete objects, commands are sent to the hosting computer where the objects are altered per the client's request messages. The altered objects are then updated to all clients. If a ghost object (client side object) is being moved, rotated, or scaled, it is immediately updated by the client's real-time design environment. While a ghost object is being moved, rotated, or scaled, the client who is currently moving the object will ignore updates from the host for that object until manipulation is complete. If the host and a client are simultaneously moving an object, the client has precedence.

The Server Structure

The server structure of the CDT-ve remains largely unchanged, (we leave the server in control) only the message system however has doubled in size to allow more detailed communication between servers and clients. A dedicated server is not necessary as the application itself is all that is needed for hosting and to start collaborative work with other users. Whichever computer chooses to host, the design will act as the central hub for all information including the design file. Based on the client-server architecture shown in Fig. 2, there can never be an occasion when clients will need to communicate directly with another client. The messages will always pass through the host. The host also acts as a moderator when it comes to resource control.

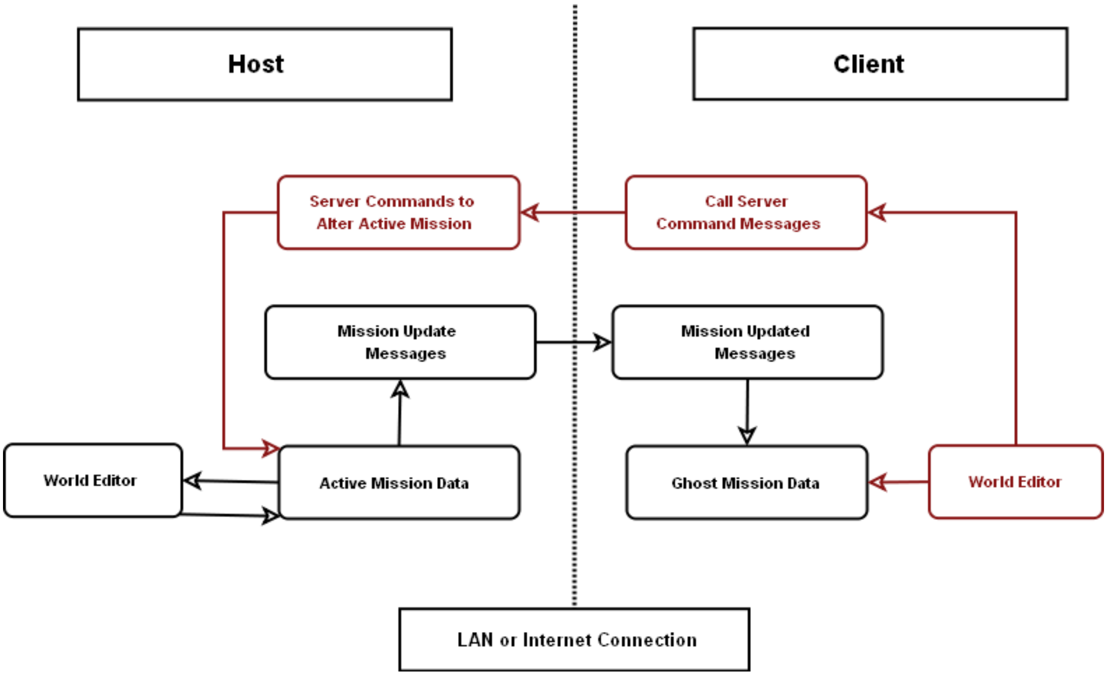

Fig. 1. The messaging update system (additions show in red)

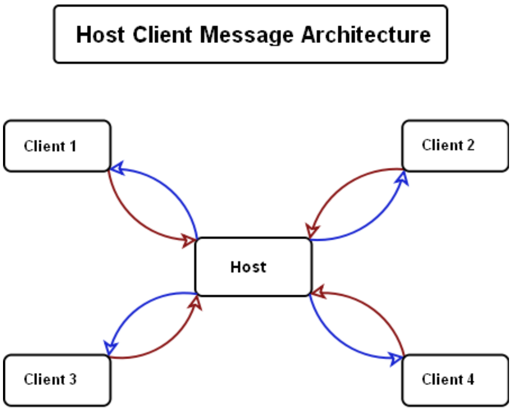

Fig. 2. The CVE client-server architecture

The host user is required to assign a password when the design file is first hosted. This ensures that design resources are secured and once authenticated; the resources are automatically uploaded to the client if the client is missing them. As a form of version control and resource management, if a client wishes to save the current design that everyone is working on collaboratively, the host will receive a message asking for confirmation and permission to save for that particular client. This action occurs every time a client wishes to save, and a negative response from the host denies the client saving privileges for that moment in time.

III. THE FRAMEWORK SUB-COMPONENTS

In addition to the primary collaborative functions of the CDT-ve, there are many sub-components to the prototype that are for the most part handled automatically. Though they are passive, they are still the key features to the collaborative design process of the CDT-ve. Some of the key sub-components are discussed below.

Resource Request

As a client PC joins a host PC's VE, objects such as the construction site (or terrain), buildings, audio cues such as sounds, and even different weather conditions are constructed on the client's computer as ghost objects. If a resource is called and is not found in the client PCs' resources, the host will then send that resource to the client and it will exist as a permanent object to the client's computer permanent pool of resources.

Avatar Updates

The manipulation of a client's avatar within the VE is an essential part of the collaborative process. There are several steps that must occur between the client pressing the keyboard command to move forward and the host updating that action to all users (see Fig. 3).

- Client issues command to move.
- Client performs local collision test.
  - If no collision occurs new position is updated on local machine
  - If collision does occur a new altered position is generated to avoid collision
- Client sends new position vector to Host
- Host performs collision test
  - If no collision occurs new position is updated to all users
  - If collision does occur altered position is updated to all users
- If Client's position differs from that sent by the Host, it is overwritten

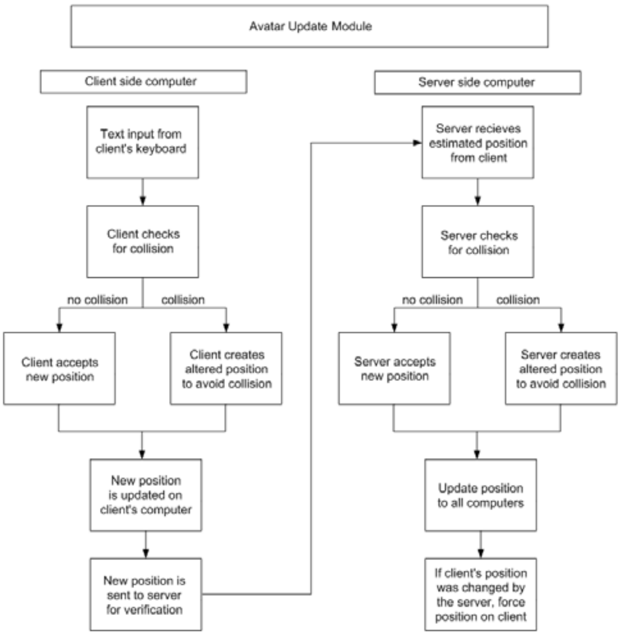

Fig. 3. Avatar Updates Model

Design Saving

When a client attempts to save, a dialog box appears on the host's computer asking if the host will allow the save. If the host denies the request, then nothing happens. If the host allows it, a recursive function is called to build the DesignFileGroup folder as it appears on the host's computer identically on the client's side including subfolders and mission info. The save file dialog box then appears for the client PC to name the design and save the file.

Texting

The ability to clearly communicate among remote designers is an important aspect within collaborative design environment. Texting is the only available communication tool at the moment. However, voice-over-IP (VOIP) programs such as TeamSpeak and Skype can be used along with the tool. The texting model is shown in Fig. 4. When a client sends a text message, the server will ensure that all clients, except the one who sent it receives the message along with that users identification profile.

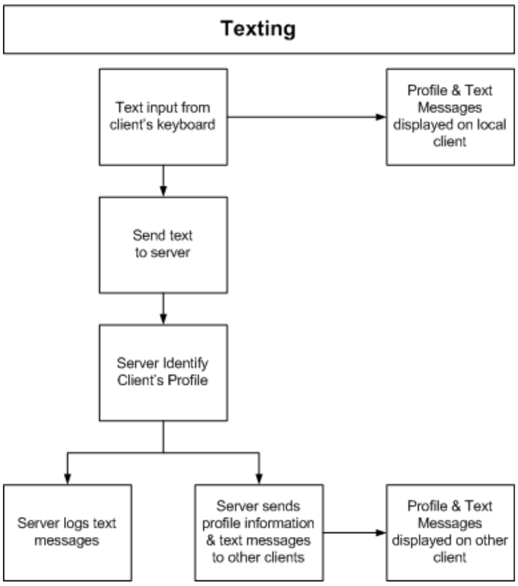

Fig. 4. Texting Model

IV. PROPOSED TESTING & PILOT TEST

Subjects will be broken down into groups of one, two, or three. The subjects who are by themselves will serve as a control group. Subjects who are in groups of two or three will be randomly divided into two sets of testing conditions. We will call these "conditions A" and "conditions B". Subjects under conditions A will be working face to face with each other and use a single non-collaborative computer with our prototype installed. Subjects under conditions B will be working in separate rooms each with a collaborative computer equipped with the prototype and a third party voice-over-IP program (see Fig. 5).

All groups will be presented with specifications and 2d plans, and shown briefly how to use and navigate the prototype. They will then be asked to construct the commercial building that appears in their plans using the prototype collaborative virtual environment. Note that only groups working under conditions B will actually use the collaborative capabilities. All virtual resources necessary to complete tasks will be provided. The prototype program will automatically record the actions of each subject, as well as log technical data (such as number of moves, rotations, etc). The conversations of subjects under conditions B will be recorded, and video of those under conditions A will be recorded. The data will then be analyzed to determine the effects of working collaboratively in a virtual environment as opposed to working face to face with a single interface into the environment.

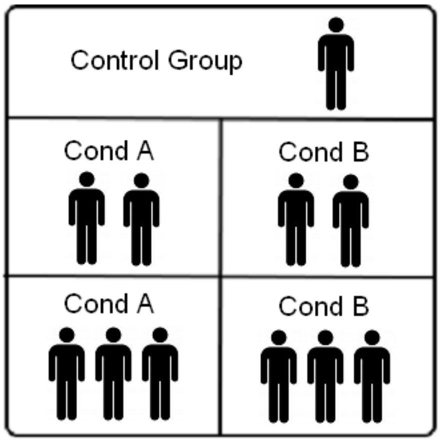

Fig. 5. Proposed testing condition

Pilot Test

We have conducted a brief pilot test using two subjects. First, one user constructed the VE by himself. Then he worked together collaboratively with the second subject to construct the VE. Subjects were in the same room and communicated verbally as well as with visual cues in the VE. Lastly the second subject constructed the VE by himself. The order the subjects constructed the VE was staggered like this to reduce the bias created from users more quickly finishing a VE they have constructed before. The results can be seen below in Table I.

TABLE I

RESULT FROM THE PILOT TEST

	Total Time	Move Count	Rotate Count*	Delete Count	Total Edits Per Minute
Sub. 1 solo	1:50	218	14	13	2.35
Sub. 2 solo	1:28	225	12	8	2.92
Sub. 1 collab	0:50	189	13	24	4.78
Sub. 2 collab	0:50	217	8	3	4.72

The results indicated in Table I was very promising. Total time was greatly reduced while manipulations per minute increased. It should be taken into account though that total time spent in the VE is not an appropriate measure of productivity as a couple of minor crashes set back subject 1 (solo) by 10 to 15 minutes and the collaborative group by 5 to 10 minutes. This is why we have included the total number of manipulations of the environment per minute: to show productivity over time. Many more variables were tracked using an embedded persistent script which automatically tracked and logged the two user's movements but only the commands directly related to the construction of the test VE have been included. Subjects reported a surprising increase in their mood as well as their work speed which they credit wholly to the presence of another individual VE. They were encouraged to work harder because someone was actively depending on them. Subjects also reported that they had an idea of the other subject's location and work area through the experiment even though each of them was in free camera mode (invisible and flying). We plan to research the psychological implications of group work and space perception in greater detail in the future.

ACKNOWLEDGEMENT

The authors would like to thank all the team members at GarageGames for their work on the Torque 3D Game Engine.

REFERENCES

[1] H.R. Hiremath, and M.J. Skibniewskib, “Object-oriented modeling of construction processes by Unified Modeling Language,” *Automation in Construction*, vol. 13, pp. 447-468, 2004.

[2] M.L. Maher, A. Cicognani, and S. Simoff, (1996). “An Experimental Study of Computer Mediated Collaborative Design,” *Proceedings of the 5th International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises (WET ICE'96)*, p 268, 1996.

[3] H.C. Howard, R.E. Levitt, B.C. Paulson, J.G. Pohl, and C.B. Tatum, (1989). “Computer integration: reducing fragmentation in AEC industry,” *Journal of Computing in Civil Engineering*, vol. 3(1), pp. 18–21, 1989.

[4] R.S. Spillinger, *Adding value to the facility acquisition process: best practices for reviewing facility designs*. National Academies Press, 2000, http://www.nap.edu/catalog.php?record_id=9769#toc

[5] W.E. East, T.L. Roessler, M.D. Lustig, and M. Fu, “The Reviewer’s Assistant system: system design analysis and description,” Champaign, IL: U.S. Army Corp of Engineers Construction Engineering Research Laboratories. (CERL Report No. FF-95/09/ADA294605), 1995.

[6] M. Fu, and W.E. EastEast, E. W, “The Virtual Design Review,” *Computer-Aided Civil and Infrastructure Engineering*, vol.14, pp. 25-35, 1998.

[7] M.F. Shiratuddin, and W. Thabet, “Remote Collaborative Virtual Walkthroughs Utilizing 3D Game Technology,” *Proceedings of the 2002 ECPPM: eWork & eBusiness in AEC*, Sept 9-11, 2002.

[8] P.S. Dunston, X. Wang, and T. Lee, “Benefit/cost analysis of desktop virtual reality application in piping design,” *Proceedings of the Conference on Construction Applications of Virtual Reality (CONVR)*, pp. 22-31, 2003.

[9] S. Emmitt, *Architectural Technology*. Oxford:Blackwell Science, 2002.

[10] M.F. Shiratuddin, and W. Thabet, “A Framework for a Collaborative Design Review System Utilizing the Unreal Tournament (UT) Game Development Tool,” *Proceedings of the 20th CIB W78 Conference on Information Technology in Construction*. Pp. 23-25, April 2003.