StringCVE: ADVANCES IN A GAME ENGINE-BASED COLLABORATIVE VIRTUAL ENVIRONMENT FOR ARCHITECTURAL DESIGN

Jules Moloney[*] and Robert Amor

University of Auckland
[*](and the Spatial Information Architecture Laboratory, RMIT)
Auckland, New Zealand
j.moloney@auckland.ac.nz and trebor@cs.auckland.ac.nz

ABSTRACT

The University of Auckland has been utilizing a multi-player game engine to develop an application (StringCVE) to coordinate architectural design and critique within a collaborative virtual environment (CVE). The initial emphasis of the research was to provide a low cost but feature rich alternative to commercial Virtual Reality (VR) in order to facilitate virtual design studios for architectural education. This paper summarizes case study feedback from beta tests, reports on current development, and positions the application relative to commercial VR systems and 3D CAD software. We propose that the most suitable use of game engine-based CVE is to support the early stages of design where teams can collaborate and evaluate iterations at a relatively low level of detail. In order for this to be a useful part of the design cycle the easy transfer of data (geometry and embedded information) from the game engine to 3D CAD software is crucial. We describe an integrated project database that allows correspondences between CAD representations and game engine resources to be maintained. This will allow a level of interchange between the game engine and typical CAD software sufficient to enable efficient use of StringCVE in the design and construction process.

Keywords: Architectural design, Game engine, Integrated database, Virtual environment.

INTRODUCTION

The use of the internet to facilitate collaborative design for architectural education and practice – so called virtual design studios (VDS) – has been the subject of almost a decade of experimentation and speculation. (Bradford et al 1994). VDS have usually involved a mix of technology and representational media but recently there have been examples of studios undertaken that use virtual reality (VR) as the primary media (Schanabel et al 2001). This approach allows “designing within the design” (Maher and Simoff 1999) in order to allow communication between design collaborators and critics or clients to occur within the emerging architectural form and the simulated site.

This approach has some promise as it allows a mode of design activity and communication similar to physical collaboration while leveraging the advantages of VR. For architectural design these advantages are summarized below.

- Synchronous design: users can propose and manipulate 3D form in a shared workspace while being physically remote.
- Intuitive decisions: the 3D form can be intuitively developed in a similar fashion to the 'mark-interpret-mark' cycle of hand sketching.
- Design context: evaluation can occur from any point of view within the simulated visual and aural site context.
- Design performance: there is a potential to include embedded information that gives the design team feedback on aspects such as environmental performance, cost, occupancy, etc as the 3D form is being manipulated.
- Asynchronous critique: critics and / or clients can access 'work in progress' and give feedback at the early design stages.

Despite the promise of VR-enabled design collaboration the research in this field has been hampered by the high cost of hardware and software and the consequent lack of case studies involving a large number of designers. In order to address the need for low cost VR for architectural design the School of Architecture at the University of Auckland has been experimenting with game engines from the entertainment sector as an alternative to commercial virtual reality systems. Our interest is with multi-user engines that provide a client server architecture that can be utilized to support collaborative virtual environments (CVE). The distinction between single-user and multi-player engines is important as this determines possible development platforms and also has an impact on collaborative potential.

There has been some use of multi-player game engines for architectural design but in general this has involved implementing the level editing feature that come with games such as Doom, Half Life and Unreal Tournament (Shiratuddin and Thabet 2002). In 2000 the School of Architecture ran a design studio in Half Life and while such an editor enabled excellent graphics and precompiled interactivity scripts (e.g., animated openings) there was no access to the source code to enable application development. By mid-2001 it was decided to investigate options for developing a more educationally orientated application based on available game engines. The key factors in the choice of a development platform besides excellent graphic and sound capability were support for large scale environments, robust multi-user capability on low bandwidth, and of course access to the engine source code to allow new functionality to be developed. In June 2001 Garage Games released the Torque game engine (Garage Games 2003) that met all these criteria and a decision was made to develop an application using the Torque software development kit. Basic functionality was added to enable the use of an application we have named StringCVE which was used for an Auckland design studio. The outcomes of this studio were reported in a case study (Moloney 2002) where the key advantages of using a CVE in comparison to typical architectural modelling and animation software were discussed in terms of:

- Design iterations: fluency of editing encouraged experimentation.
- Participatory critique: reviewers actively explore rather than observe.

From this case study a further set of requirements were established to enable asynchronous modes of communication to facilitate internet-based design critique. This functionality has been implemented using a mySQL database linked to a PHP chat forum to enable 3D browsing of comments and ‘white board’ type mark ups of the designs. (Moloney et al 2003). The current communication modules are summarised in Figure 1.

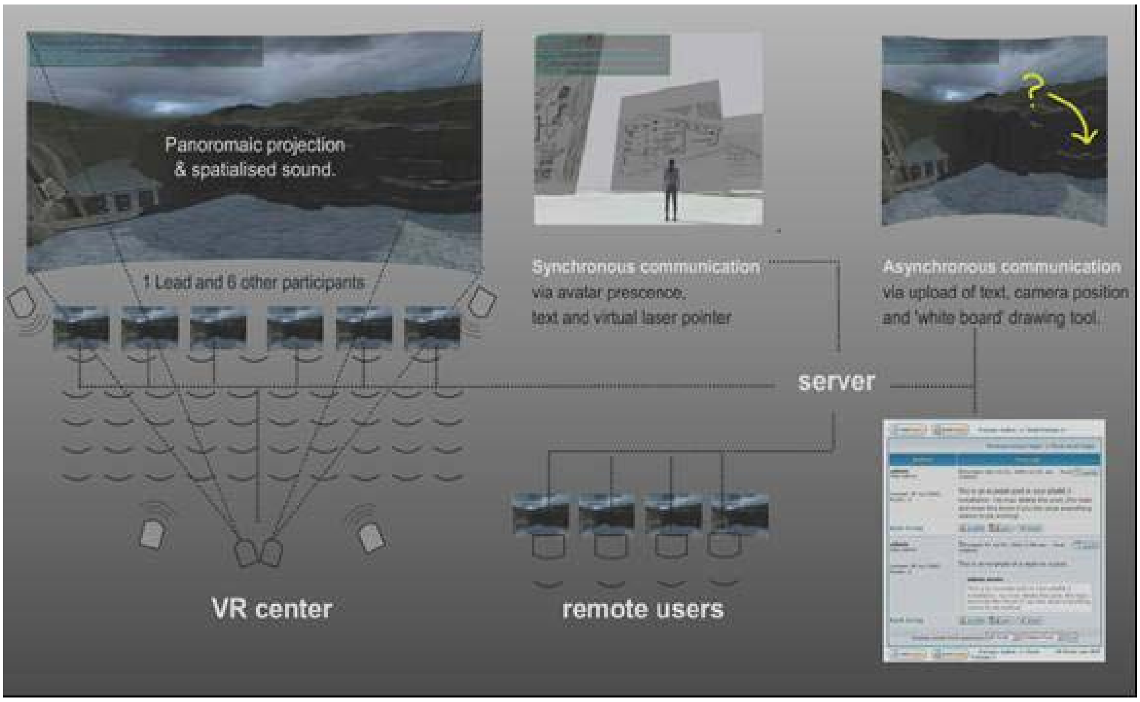

Figure 1. Overview of StringCVE communication modules.

As a result of feedback from beta test users the database functionality is currently being extended to provide support for collaborative design as well as critique. New functionality will:

1. Allow multiple users to access a database server and add new design components to, or modify existing design of, existing projects. On client login all new design components and design edits are accessible
2. Allow any client to manipulate the design (environment and components) and propagate these updates to multiple clients in real time.
3. Provide managed control of the data set: Levels of access would need to be assigned to each user and design; Implementation of ‘save edit’ and ‘design history’ functions are planned with each edit able to be undone (i.e., users can step through design versions).
4. Extend existing asynchronous design critique functionality to allow comments to be recorded with each design iteration

The original emphasis for the research was to provide an application to support design education but interest from professional architects and other commercial users has prompted a review of the development. The aim of this paper is to position the use of game engine-based VR such as StringCVE in relation to the strengths of commercial VR and 3D CAD software and outline key functionality to enable its efficient use as part of a professional design cycle.

BLURRED BOUNDARIES: VR / 3D CAD

Before considering how the capabilities of game engines may be extended it would be useful to position the technology in relation to 3D CAD software and Virtual Reality systems. We note that there are varying types of VR but for the purposes of this discussion we are referring to immersive and distributed VR systems such as CAVERNsoft (2003). However, the aim of the discussion is not to identify particular software attributes or clear boundaries between VR and 3D CAD but to identify the relative strengths of each system as a means to determining the most appropriate development approach for StringCVE. Three key factors are identified for the purpose of this discussion: design context; environmental interactivity; and collaboration.

Design context

In terms of 3D graphics there is no easy distinction between the sophisticated 3D capabilities of software such as Autodesk Viz and VR software. With a standard graphics card such 3D CAD software allows fully textured models to be examined in some detail and in real time. In many respects the distinction between 3D CAD and VR is made in relation to hardware. VR systems such as a CAVE or using head-mounted displays are distinguished by the use of displays that fill the user’s visual field. The aim of such hardware (when used for architectural design) is to distance the user from their current surrounding in order to immerse them in the computer simulation. Within such systems the architectural model is viewed within a simulated *environment* as opposed to typical 3D CAD software where the model is usually viewed in the abstract context of the preview window. This distinction between the context of a typical CAD preview window and VR environments where an attempt is made to place the design in context is a vital distinguishing feature that enables a fundamental shift in design emphasis. Rather than concentrating on architecture as a singular object devoid of context there is the opportunity to view the design in relationship to a simulated physical context, in relation to other design precedents, or in reference to embedded information. A primary motivation for designers to develop projects in VR systems is this potential to make decisions relative to the real time context of a simulated environment. The term environment should be interpreted here in as wide a definition as possible - both the visual and aural context and embedded information that gives the design team feedback on aspects such as environmental performance, cost, occupancy, etc.

Environmental Interactivity

A second factor that distinguishes VR systems is the degree of interaction within the designed environment. We are not referring here to interactivity in terms of design modeling where typical 3D CAD geometry, texture and lighting tools have more detailed and sophisticated interfaces. Such interactive modeling is much more limited in VR but embedded triggers or time-based behaviors enable a type of interactivity not found in 3D CAD. We refer to this form of interactivity as environmental. The interaction can be simple visual cues such as animated openings or day / night light cycles. Such visual interaction can be augmented with 3D located sound to simulate, for example, external noise levels in various parts of the design. Complementing this 'experiential' interaction is the possibility to embed data relative to the model and its status in the environment and interactively display this to the user. Hence interaction in virtual environments enhances both the experience of the design and can simultaneously provide environmental performance and other data such as occupancy levels and cost factors. A 3D CAD system can of course provide similar functionality by rendering animations and attaching data to objects – the distinction is that a VR system is dynamic allowing interaction between user, 3D model, environment, and embedded information in real time.

Collaboration

The third issue we will consider is that distributed VR systems allow the opportunity for multiple users to collaborate in real time. Such synchronous collaboration can occur via text chat, voice communication and through interaction with shared design components. In this way multiple users can develop designs in a real time virtual simulation. 3D CAD systems do not allow for such real time collaboration but there are possibilities for users to collaborate asynchronously. Typically this is through text annotation and the marking up of designs via whiteboard-type functionality. In addition to synchronous or asynchronous communication the capability of the software to support a range of file types is an important attribute affecting collaboration. Being able to import and export multiple file types is especially important given that consultants will often be using different software. In this respect 3D CAD has greater support than typical VR systems.

Figure 2. StringCVE asynchronous communication module.

Levels of augmented collaboration from StringCVE are shown in Figure 2. From left: Comments are embedded within the virtual environment and can be browsed by clicking on descriptions; this 'teleports' the user to the camera position of the user who entered the comment and also overlays a mark up graphic entered by the user; simultaneously the text comment is located and displayed in a standard PHP chat forum; replies to the initial comment can be entered; the project can also be navigated via the PHP comments to bring up other camera views and mark up graphics.

Positioning StringCVE Between VR and 3D CAD

In the brief overview of relative strengths we have identified design context, environmental interactivity and collaboration as key differences between an immersive

VR system and typical 3D CAD. These are summarized in Table 1 with the current functionality of StringCVE

	Design Context	**Environmental Interactivity**	**Collaboration**
VR	Excellent, high detail	Some	Synchronous
StringCVE	Good, low level of detail	Some environmental interactivity	Synchronous and asynchronous
3D CAD	Poor, high level of detail	None	Asynchronous

Table1. Comparison of strengths of design tools.

Inevitably the results of the comparison relate directly to the needs of the original user for which the software was specified. For CAD users the 3D functionality is a development of their primary function as detailed drawing tools for architects and other design professionals to produce construction information. Complex details can be visualized in 3D in order to develop detailed design solutions where the implications of 3D junctions are difficult to perceive in two dimensions. High accuracy and detail are required but there is little need to visualize the full context in which the detail is located. There is also no need for the model to be interactive other than for manipulating the 3D viewpoint while collaboration is limited to a supervisory activity - hence the development of drawing 'mark up' functionality to enable the project leader to monitor design decisions. Typical VR users are harder to classify given the range of applications but in general users require a fully immersive experience with high levels of detail and some interaction. Collaboration in distributed VR is possible but in practice this is limited by the number of systems available - the high cost precludes widespread use as a collaborative design tool. In comparison to VR users, multi-player game users require highly interactive environments with good collaboration possibilities while there is less need of full graphic immersion as the focus is on the game play and interaction. Conversely when compared to 3D CAD, game engines are not designed to support high levels of detail or dimensional accuracy.

In terms of design context, environmental interactivity and collaboration an application based on a multi-player game engine can be positioned between the high end visualization capabilities of VR and the high dimensional accuracy of 3D CAD. The pragmatic advantage of utilizing a game engine to produce a collaborative virtual environment application is that the primary VR functionality is available at low cost (i.e., real-time graphics and sound, environmental interactivity and multi user support). CVE developers should take advantage of this ‘egalitarian’ opportunity but be wary of developing functionality that is well catered for in 3D CAD systems. We propose that the most suitable use of a game engine-based CVE in design and construction is to support the early concept stages where teams can collaborate and evaluate iterations at a relatively low level of detail. This positioning of game engine-based VR as a design communication and evaluation tool will avoid inappropriate use as a surrogate 3D CAD software and will complement VR systems where full graphic and aural immersion is more suitable for representation of final design solutions.

REPRESENTATION AT THE EARLY STAGES OF DESIGN

It can be argued that full immersion in VR systems is in effect a hindrance to collaboration and design thinking at the early design stages. While there are many approaches to formulating ideas in architecture, researchers of architectural pedagogy have identified two methods aptly named “analysis - synthesis” or “hypothesis - test” (Ledewitz 1985). These problem-based methods, where design solutions are a result of analysis or propositional testing of program requirements, are complemented by recent thematic approaches where designers may investigate more abstract ideas at the early design stages. No matter the approach conceptual design is a reflective activity that operates at a lower level of representation than that available in immersive VR. For architecture, design has traditionally been conceived via abstractions and it is probable that high levels of detail and realism are counterproductive to conceptual thinking. As noted by distinguished architect and theorist Stan Allen, “design does not operate on the basis of resemblance, but on abstract codes” (Allen and Agrest 2000). These codes are the ‘shorthand’ of plan, elevation and sectional sketch that enable preliminary form, space and circulation permutations to be studied. Despite the proliferation of design software in design studios such sketching is still commonly used and it is unlikely that the natural fluency and eye hand co-ordination possible with analogue media will be completely replaced. Nor is such replacement of analogue media desirable. We have argued in a previous publication that working simultaneously across a range of media enables the computer to complement drawing by hand and physical models (Moloney 1999).

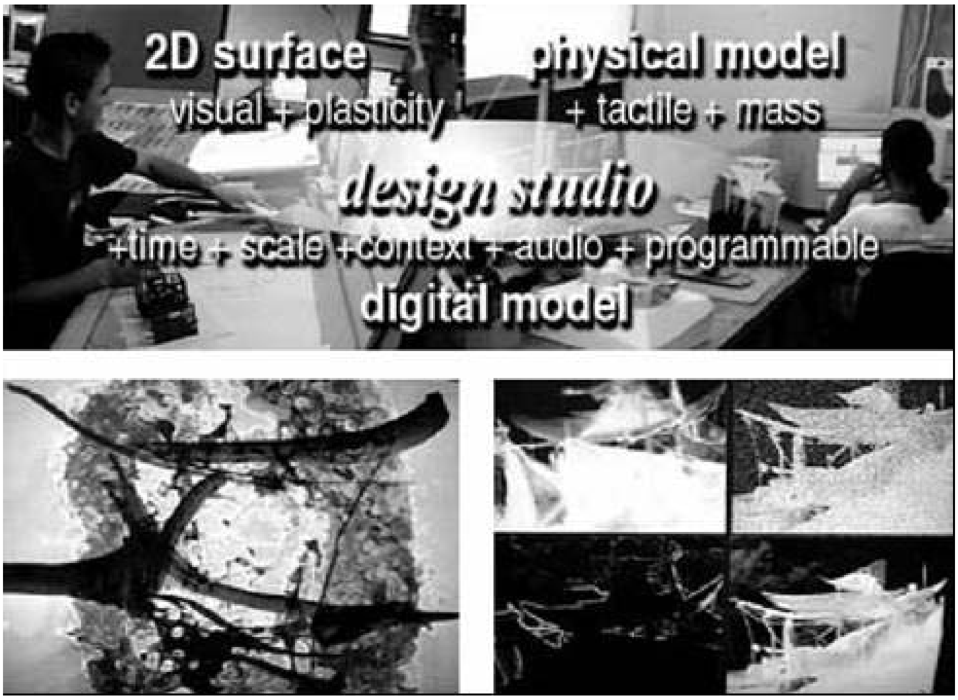

Figure 3. Cross Media working at the early stages of design.

Unfortunately, VR places the designer in a digital vacuum that disables such cross media approaches. Conversely the less realistic graphics and non-immersion of systems such as

StringCVE allow some valuable distancing from the 3D simulation. The ability to sketch on paper, converse with a colleague seated alongside and interact in a screen-based virtual environment with other remote designers allow design iterations to unfold within a combination of the digital and the analogue (see Figure 3). While it is valuable to maintain such continuity with analogue media the possibilities of digital representation to extend the mode of enquiry at the early design stages have yet to be fully exploited. For example, trials of StringCVE have experimented with spatial sound to augment design projects (see Figure 4).

In a recent design studio at the Spatial Information Architecture Laboratory (SIAL) students utilized location recorded sound samples in order to enhance the context by triggering memories of site specific conditions and to allude to functional program (More et al 2003). A second studio will continue these investigations of the relation between visual and aural cognition, a mode of enquiry not easily achieved with CAD software.

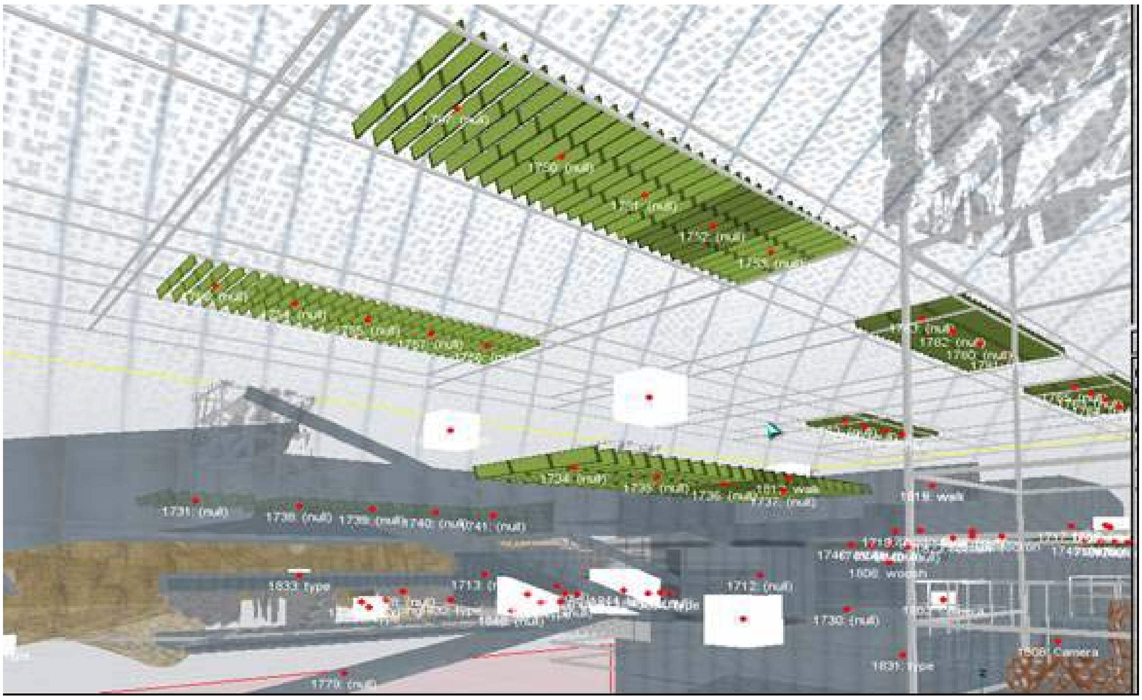

Figure 4. Project from SIAL design studio in edit mode with embedded sound samples.

We are also interested in the capacity for a CVE to augment visual representation with information that gives the designer real time feedback on design attributes such as floor area and environmental performance. This embedded information need not be highly accurate for the early stages of design but should provide approximate values on building performance to enable the designer to monitor the implications of different basic approaches. A precedent for this development is the Pangea software developed by researchers at the Bartlett School of Architecture (Pangea 1996) that linked computer models to a range of 'intelligent' techniques (Neural Nets, Genetic Algorithms, Fuzzy Logic, etc) that communicate design performance to the designer.

DATA TRANSFER BETWEEN A GAME ENGINE AND CAD TOOLS

For a game engine-based CVE to be a valuable part of the design and construction cycle there needs to be more compatibility with standard 3D CAD software. Our current development focus is on the best way to achieve transfer of geometry and embedded data between the two systems. We continue this paper with our proposal for such an integrated project database.

To enable a level of data transfer between tools there are three main approaches which can be taken:

1. A bespoke converter. This has been the favored approach for many CAD tools allowing 3rd party additions. In this approach a converter is built on top of the CAD system working directly with its API (Application Programming Interface) and dealing with the building representation in whatever form is offered by that CAD system. This approach works very well where integration with a single CAD system is required and where the 3rd party software can be updated every time there is a new release of the CAD software (with new data structures and API). However, it is not an open solution, so requires a large effort to develop converters for all the major (and minor) CAD systems that you would like to extract data from for your supplementary design tool.
2. Standard building file transfer. This is the approach pushed by standards bodies such as ISO and IAI (2003). Here a standard data model for building information is agreed upon by the construction industry and all CAD and design tools provide an interface to allow a building model to be transferred in and out of their tool in this form. With the current IFC standard (IAI 2003) a building model is transferred in a format called SPF (STEP Physical File, ISO 1998) which embodies the complete building information within one file. There are other standards for allowing partial transfer of building information, but these are not used in commercial tools. This approach works very well where it is desirable to transfer the complete building representation to another tool to be worked upon. Building model transfer through IFCs is now offered by many of the major CAD vendors (e.g., AutoDesk, GraphiSoft, Bently) However, it does not allow concurrent working on a building model (as the whole building model is within a particular tool at any one time) and it requires each tool to handle the whole representation of a building, even though the tool may only require a small portion of the building model for its particular analysis.
3. An integrated project database (IPDB, Amor and Faraj 2001). This is more of a research approach, where a centralized database is created holding the complete building model in a standardized and structured form (e.g., IFC). The IPDB coordinates and controls all flows of information back and forth between design tools and CAD systems, ensuring that each tool receives only the information it requires. It allows partial building model transfer and can propagate modified information through to tools which need to be informed of updates. However, there are very limited commercial implementations of this technology to date, though the IFC Model Server project (Adachi 2002) may soon provide an open source system to perform this task.

For the StringCVE project we are developing the information transfer mechanism along the lines of the IPDB approach. However, to integrate with the games engine there is one further complication. The representation of a building within the game engine is of such a different form than found within the IFC representation, or an object-based CAD system (e.g., ArchiCAD, AutoDesk Architectural Desktop), that there is no likelihood of providing a bi-directional IFC translator over the top of the game engine.

IPDB Development

To understand the difficulties posed by the respective representations of the games engine and the CAD tool we need to characterize their internal representations. An object-based CAD tool represents a building through a collection of inter-related components, where each component represents some portion of the concrete structure of the building (usually). In older CAD systems this representation was based purely around the geometry of such components, in the object-based systems there are many parameters associated with these components. In the games engine there is no real representation of a building, rather there are a set of resources which are rendered by the engine and, through a human's perception, allow us to intuit the form of a building.

To handle these differences we propose an IPDB system which can hold two disparate representations of a building and maintain the correspondences between them. The CAD representation in the IPDB is an object-based representation utilizing the IFC standards for building information in general. The games engine representation in the IPDB is based on the types of resources which are allowable within a games engine. To allow some transfer of information between the two representations the IPDB maintains a correspondence table which allows us to ascertain what the relationship is between a set of building components within the CAD/IFC view of a building and the games engine view of the building. These correspondences are established at the time that a CAD-based building representation is transformed into the games engine resource format. Each games resource is tagged back to the originating structures in the CAD representation. With this correspondence established it is possible to map a restricted set of transformations within the games environment back to the CAD system.

The transforms which we are supporting are based on the modifications to resources allowable within the games environment. This covers geometric transforms such as rotations, scaling, and translation which transform the associated building components in the same manner as within the games environment. Resources can also be deleted causing the removal of components within the CAD model. We are currently exploring the support that could be offered for adding resources into the games model. This would equate to adding structure and components to the CAD building model. To support this we envisage a precompiled library of resources (like a product catalogue) which could be inserted into the games environment. Each resource available to the games environment would have a correspondence to a set of structures within the CAD representation. Therefore, any addition into the games representation in the IPDB can be matched with corresponding structures in the CAD representation in the IPDB and later replicated through to the CAD system.

With an appropriate user interface embedded into the games environment it will then be possible to perform early design collaboration with simple building alterations and additions being transformed into CAD-level objects able to be fully detailed with the more appropriate CAD tools.

CONCLUSIONS

We have documented the outcomes of early development of StringCVE, a collaborative virtual environment application based on the Torque game engine. The relative strengths of such game engine-based software has been discussed in relation to immersive VR systems and typical 3D CAD software. Multi-player game engines enable low cost, feature rich screen-based virtual environments but compared to 3D CAD the graphics architecture is not suitable for detailed or dimensionally accurate use. The representational requirements for the early design stages have been discussed and we conclude that a game engine-based CVE is best suited for use at this stage of the design build cycle. We propose that a game engine-based CVE offers much potential as a representational and communication tool for the early stages of design. The current objective of the StringCVE project is to provide functionality to facilitate collaboration in an information rich, visual and aural context to enable a range of preliminary design ideas to be developed. Once agreement has being reached the design can then be developed in detail via 3D CAD. The easy transfer and coordination of data (geometry and embedded information) from the game engine to 3D CAD software (and vice versa) is crucial if such design collaboration is to be an effective part of the design process. We have identified an integrated project database as the most appropriate solution for data transfer and have described the general approach to implementing such a database. The IPDB is currently been written for the StringCVE application and will be beta tested in a design education context in early 2004.

REFERENCES

Adachi, Y. (2002) Overview of the IFC model server framework, Proceedings of EC-PPM, Portoroz, Slovenia, 9-11 September, pp. 367-372.

Allen, S. and Agrest, D. (2000) Practice: architecture, technique and representation, ISBN 90-5701-032-1, Amsterdam: G+B Arts International.

Amor, R. and Faraj, I. (2001) Misconceptions about Integrated Project Databases, ITcon journal, http://www.itcon.org/2001/5/, ISSN 1400-6529, Vol. 6, pp.57-68.

Bradford, J.W., Cheng, N. and Kvan, T. (1994) Virtual Design Studios, The Virtual Studio, Proceedings of the 12th European Conference on Education in Computer Aided Architectural Design, ISBN 0-9523687-0-6, Glasgow, Scotland, 7-10 September, pp. 163-167.

CAVERNsoft (2003) Online reference last accessed July 2003 http://www.openchannelsoftware.org/projects/CAVERNsoft_G2/.

Garage Games (2003) GarageGames and the Torque Game Engine, http://www.garagegames.com/, last accessed 14/8/2003.

IAI (2003) International Alliance for Interoperability, http://www.iai-international.org/.

ISO (1998) Part 21: Implementation methods: Clear text encoding of the exchange structure - Product data representation and exchange, ISO-IEC, Geneva, Switzerland, ISO 10303-21.

Ledewitz, S. (1985) Models of Design in Studio Teaching, Journal of Architectural Education, 38(2), pp. 2-8.

Maher, M.L. and Simoff, S. (2000) Collaboratively Designing Within the Design, Proceedings of Co-Designing 2000, Coventry, UK, 11-13 September, pp. 391-399.

Moloney, J. (1999) Charcoal, Bits and Balsa: Cross Media Tactics in the Foundation Design Studio, Architectural Computing from Turing to 2000, eCAADe Conference Proceedings, ISBN 0-9523687-5-7, Liverpool, UK, 15-17 September, pp. 110-115.

Moloney, J. (2002) String CVE: Introducing a Virtual Design Studio Utilizing the Torque Game Engine. Proceeding of the 20th conference on Education in Computer Aided Architectural Design in Europe, ISBN 0 9541183 0 8, Warsaw, Poland, 18-21 September, pp. 522-525.

Moloney, J., Amor, R., Furness J., Moores B. (2003) Design Critique Inside a Multi – Player Game Engine, Proceedings of the CIB W78 Conference on Construction IT Bridging the Distance, Waiheke Island, New Zealand, 23-25 April, pp. 255-262.

More, G., Harvey, L., Moloney, J. and Burry, M. (2003) Implementing Non linear Sound Strategies Within Spatial Design: Learning Sound and Design Within a Collaborative Virtual Environment, Proceeding of MelbourneDAC, the 5th International Digital Arts and Culture Conference, Melbourne, Australia, 19-23 May, pp. 135-141.

Pangea (1996) http://cebe.cf.ac.uk/learning/habitat/HABITAT2/pangea.html.

Schnabel, M.A., Kvan, T., Kruijff, E. and Donath, D. (2001) The First Virtual Environment Design Studio, Architectural Information Management, 19th eCAADe Conference Proceedings, ISBN 0-9523687-8-1, Helsinki, Finland, 29-31 August, pp. 394-400.

Shiratuddin, M.F. and Thabet, W. (2002) Virtual Office Walkthrough Using a 3D Game Engine, International Journal of Design Computing, 4, ISSN 1329-7147. http://www.arch.usyd.edu.au/kcdc/journal/vol4/.