A collaborative analysis tool for visualisation and interaction with spatial data

A Collaborative Analysis Tool for Visualisation and Interaction with Spatial Data

Tina Manoharan[a,*], Hamish Taylor[a], Paul Gardiner[b]
[a]Heriot-Watt University, Computing & Electrical, Edinburgh EH14 4AS, UK
[b]Heriot-Watt University, School of Management, Edinburgh EH14 4AS, UK
[*]E-mail: ceetm2@cee.hw.ac.uk

Abstract

A collaborative virtual environment system is described that is designed to support location-independent shared analysis of spatial data and urban planning proposals. The system seeks to extend the physical workplace of participants into the virtual environment, while preserving traditional textual and verbal communication and cooperation mechanisms. The systems aim is to improve productivity, quality and achieve more transparency in the planning process. The architecture of the Collaborative Urban Planner or CUP system is described and some experimental results that demonstrate urban development control tasks performed within this environment are presented. An application scenario offers a vision of future urban planning practice using CUP. The scenario also diagrammatically demonstrates virtual settings and scenes that could become everyday meeting places for remote planners, architects or engineers assessing proposals and discussing possible alterations to designs.

1. Introduction

The potential of collaborative virtual environment (CVE) technology for computer supported cooperative work (CSCW) is widely acknowledged with a variety of organizations and institutions increasingly finding novel and innovative uses for this technology. These uses encompass the fields of design, entertainment, medicine and engineering [2,4,10,14]. Applications of CVE technology have evolved into successful architectures like DIVE [6], MASSIVE [9], COVEN [14], MAVERIK [12], DEVA [15] and DeepMatrix [17]. This promising new technology will clearly benefit from further evolution through its application to challenging problem domains.

This paper presents the design and development of Collaborative Urban Planner (CUP) a prototype system that will aid shared analysis of urban planning proposals by visualization and interaction with spatial data. Sharing the virtual experience can improve the collaboration among planning officers and other stakeholders in a planning project such as applicants, architects, engineers and the public (*see* Figure 1). These participants, who may be geographically separated, can use the system to explore alternative designs independent of time and place. Planners must work against tight time limits and consequently the system embeds appropriate spatial analysis tools to enable rapid determinations of the acceptability of individual proposals. The approach intends to improve productivity, increase public participation and achieve more transparency in the planning process. It is envisaged that the quality of the

decision-making process can also be improved by virtue of the technology's ability to support collaboration and thus reduce misunderstandings in the negotiation of a proposed development. Better informed decisions will also increase the confidence that the public has in the urban development process.
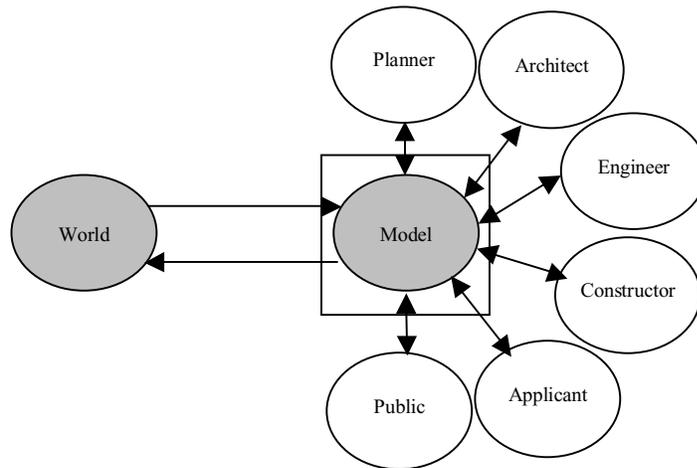
Figure 1: Collaborative urban planning in real world via shared virtual world

2. Application scenario

The application scenario for CUP is based on four major task domains integrated into the system: visual assessment, interactive exploration, collaboration and information management. The scenario is based on UK planning practice, on which the system was initially modelled, but the concepts are applicable for urban planning processes in general [1,21]. CUP is envisaged as a powerful tool that will introduce a new method of collaboration among stakeholders in an urban project with positive changes to contemporary planning practice (*see* Figure 2). This web-enabled tool will also facilitate the globalisation of urban planning services.
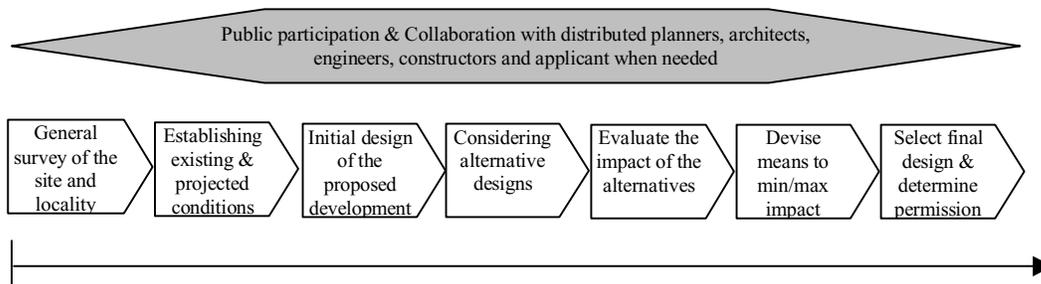
Figure 2: Collaborative urban planning cycle

Consider a case where an international firm puts forward a proposal to construct a branch office in Edinburgh, UK. The use of CUP anticipates an altered planning process where the applicant also submits a computer model of the proposed office building, developed by their architects. The planning committee arranges a meeting

with necessary planning officers to discuss issues relating to the application before granting or rejecting planning permission. At this stage the system can be used as follows. Three or four members can log on to the system at a few workstations while the others sit around a discussion table (*see* Figure 3). Any change made to the VE by one user is immediately communicated to all other workstations. In addition contents from a workstation can also be projected on to a larger screen. At first, planning officers can perform a general survey of the site and locality by navigating through the VE. Once existing and projected conditions are established, a logged in member can insert the supplied model of the proposed development. Thorough environmental impact assessment of this initial design can be done using appropriate analysis tools provided by the system [*see* Manoharan *et al* 13 for details on the system's built-in analysis tools]. For example, development control tasks such as privacy checks, daylight and shadow checks or visibility checks can be performed on the model (*see* Figures 9a-11b).
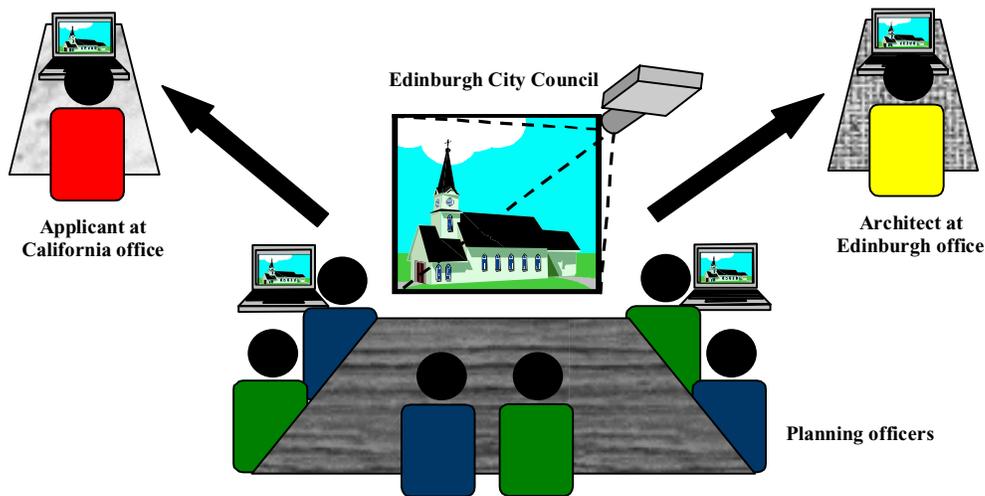
Figure 3: Example collaboration scenario using CUP

If this initial design is not acceptable to the planners they can consider alternatives. Suggestions for changes to the design raised by the attendees can be immediately executed on the system. If advice is required from other experts such as city architects or engineers, the planners can rapidly collaborate with them by asking them to log on to the system from their office workstation. Voice and text communication methods provided by the system, can be used by the participants for discussion. Public participation is also possible at every stage of the planning cycle. The public can log on to the system and view alternative designs suggested by the committee. If they have objections they can lodge them rapidly by directly communicating with the planners or by saving their comments within the system. The planning officers can consider and act on these responses as appropriate.

After evaluating the impact of the alternatives, decisions regarding the proposal can be made. These decisions are debated and illustrated within the CVE and communicated to the applicants situated at their head office in California. If these changes are not acceptable to the applicants they can communicate their feedback to the planning committee along with alternatives more suitable to their needs. In this

3

way, distributed parties can collaborate within the system until an appropriate final decision is made. Finally, if the proposal is accepted, the model gets added to the planning database of the town and will become a part of the official model.

In future years, these virtual settings and scenes could well become everyday meeting places for remote planners, architects or engineers assessing planning proposals and discussing possible alterations to designs.

3. The CUP system

3.1 High level overview of approach

Opening CUP's client applet in a web browser results in a login screen that allows the user to input their name, password and also choose their role in the planning process. Currently supported roles for participants are planner, architect, engineer, constructor, applicant and member of public. Each type of role is associated with a set of functionalities appropriate to that role. Actions that are unavailable to a particular user role are always greyed out. The user is represented in the world with a specific avatar depending on their role. Every user can identify the other users in the world with the avatar representation and the name of the user displayed over the avatar's head. When the user successfully logs in, the GUI allows the user to select objects that should form the VE. Selected objects are loaded to form the user's visual interface. In addition there are also six panels supporting various planning actions (*see* Figures 7 & 9a). This interface serves as a means of initiating tasks and providing feedback to the user. After initiating a task, the user can interact with the VE using the mouse or keyboard. The user can navigate the world and perform required operations on objects after selecting them.

3.2 System architecture

CUP is an extension of Geometrek's DeepMatrix 1.1, an existing multi-user VE system based on two open technologies Java and VRML [17]. The main components of CUP are the Java3D client, the Java server and a MySql database (*see* Figure 4).
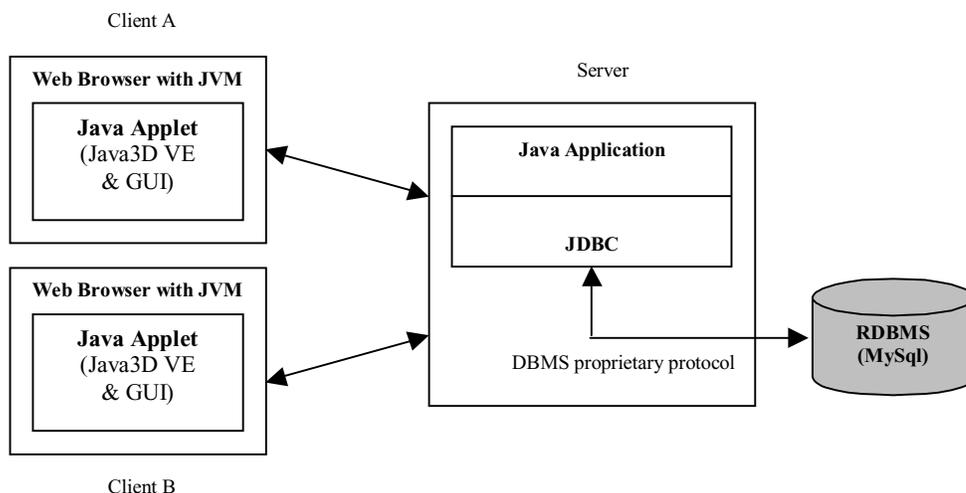
Figure 4: Architecture of the system

The client component has been implemented completely whereas the server component has been extended basically in two ways: (1) to support persistence and dynamic modification of the world, (2) to integrate information sources seamlessly.

3.2.1 The client component

The client is redesigned and extended as a Java JApplet with a Java3D and Swing GUI embedded into an HTML page. When the user opens the web page, the applet is transferred from the server to the client where it is interpreted and executed by the JVM provided by the JRE1.3 plug-in installed in the browser. The plug-in is required for using Swing in the applet when running in a browser.
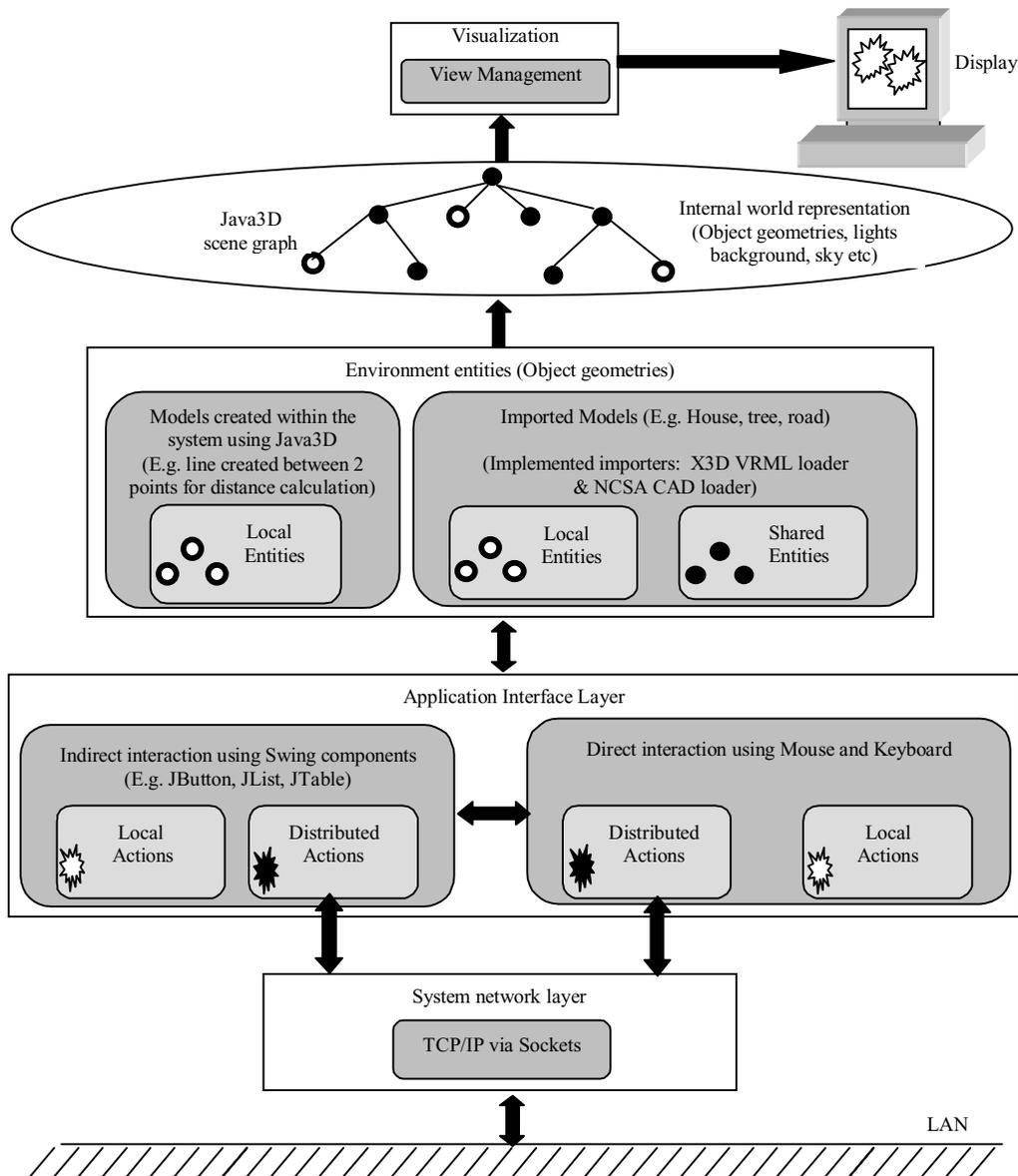
Figure 5: Internal organization of the client application

The 3D visualization was rebuilt and upgraded using Java3D because Java3D supports highly interactive VEs that require changes based on complex user interaction [20]. Although this can also be done with VRML and JavaScript, Java3D can provide a simpler and more elegant solution with the logic and visual interface implemented within the same technology. Java3D also provides better programming control than VRML. Loaders are available for VRML, CAD and other popular file formats to import models into a running Java3D program. Current web browsers like Netscape can support Java3D by installing the Java3D runtime environment into the JRE installation. The plug-in must be pointed to this JRE for executing Java3D in the browser.

The client application's user interface supports visualization and interaction (*see* Figure 5). An internal world model with pre-defined local and shared VRML and Java3D objects forms the visualization part of the user interface, representing planning information as a VE. An example of a local Java3D model would be the line generated when performing distance calculation or visibility check connecting the two end points. Buildings, streetlights or trees in the world are examples of shared VRML models. The capability to import shared VRML and CAD objects dynamically was incorporated using X3D's VRML loader and NCSA's CAD loader. Users can perform actions at a client that can be either distributed or local. All distributed actions performed at a client are immediately transmitted to the server through the network layer to be communicated to other clients. For example, viewing detailed information of a particular object is a local action whereas rotating an object is a distributed action.

The basic structure of this internal world representation can be depicted by a Java3D scene graph (*see* Figure 6). The scene graph contains visual objects, lights, sounds, and behaviors that handle interactions or modifications to objects in the scene. The root node is split up into three subgroups each having their own functionality within the system. The group 'world interaction' contains tools to support navigation, distance measurement, visibility determination and animation. One or more global light sources are added to light the scene. These lights and the background are placed under the 'world realism' group. The geometry nodes and interaction behaviors for the different objects are added under the 'world objects' group.
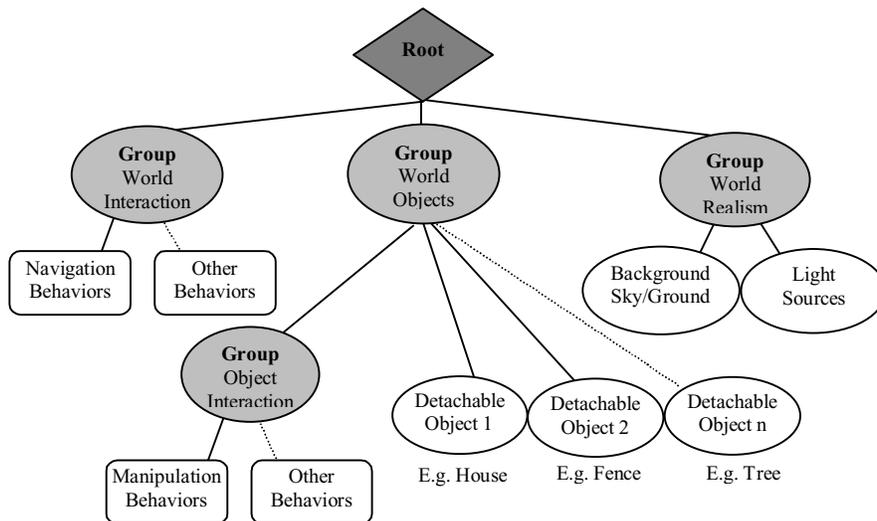
**Figure 6: Scene graph of the system**

The second part of the GUI, which is below this 3D visualization, consists of a set of panels supporting various planning tasks (*see* Figures 7 & 9a). These panels are shown by default but can be hidden when not required or when using a pop-up menu. Some operations like object manipulations or navigation can be performed by direct interaction using the mouse or by initiating them from the panels. On the other hand some tasks like distance calculation or visibility check are initiated only by using the components on the panels or the items on a pop-up menu. In addition the panels also serve as a means of providing feedback to the user. After initiating a task from the panel, the user can interact with the scene using the mouse or keyboard. For example, to calculate the distance between two points the user has to first click the 'Points' button on the 'Navigate' panel and then selecting the two points by clicking on the scene with the mouse. Next, by clicking on the 'Distance' button the system draws a line connecting the two points and displays the distance in meters over the line and also in a text field provided on the 'Navigate' panel. On the other hand an object can be selected by directly clicking on the object with the mouse or by using the 'Select' button on the 'Object' panel or pop-up menu. The system provides feedback by creating a bounding box around the selected object and also displaying a message such as "object 1 selected by user 1' on the status bar of the 'Object' panel.
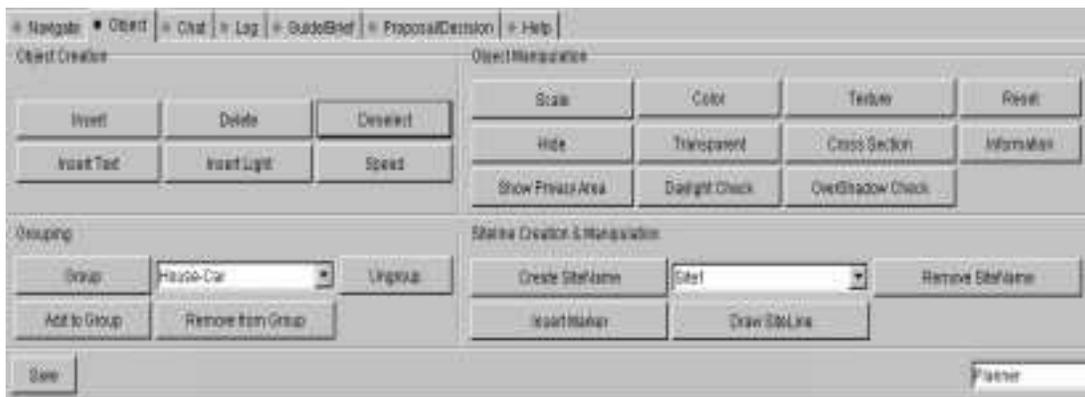
Figure 7: Six panels forming the GUI in addition to the VE

3.2.1 The server and database components

The server is a multi-threaded Java application that communicates with the client via sockets (*see* Figure 8). The server listens to the TCP/IP port and spawns a new thread (client request handler) for each client connection request it receives. This client request handler is responsible for any further communication with the client. In addition it also multicasts information from a client to a set of clients they are designated for. DeepMatrix's client-server communication interface was extended to support a larger range of message types including control and data requests. The server logic layer is responsible for maintaining temporary and permanent information about all shared objects in the world and also for sustaining concurrency control using Java's synchronization techniques.
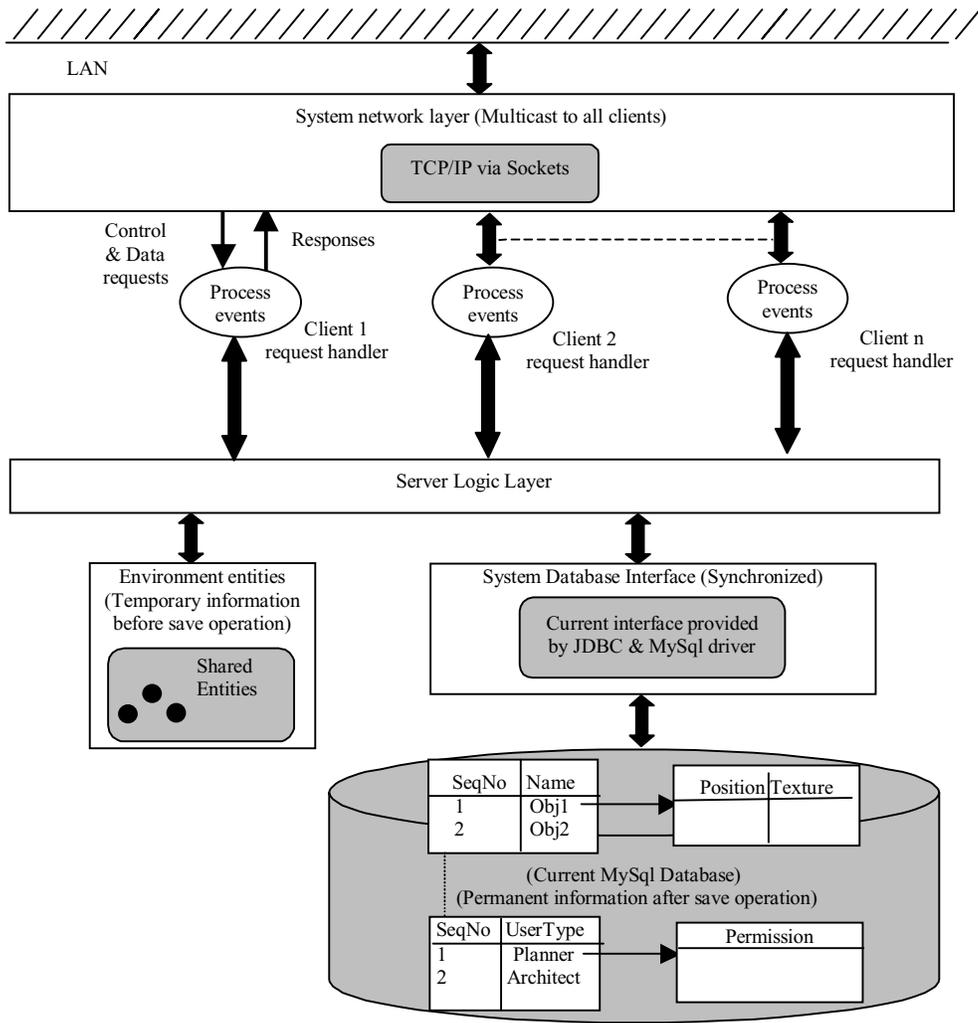
Figure 8: Internal organization of the server and database application

Another important extension made to the original DeepMatrix server was to provide transparent data persistence using Java DataBase Connectivity (JDBC) technology to access a MySql database. Clients can request information from the server, which extracts the queried data from the database and sends it to the client. A backend database supports persistence of data and provides a better way to store and manage large amounts of planning information. User security details and data relating to every object in the VE are stored in the database. Other planning information such as design guides, briefs, proposals, decisions and public comments are also stored. Java's synchronization techniques are used to maintain data consistency. Having one point of access to the database also helps to coordinate data access and maintain data integrity.

The client currently runs on a 550MHz P3 PC with 256MB, WinNT and OpenGL on a Diamond Fire GL 1000 Pro graphics card. It renders scenes with 10,000 x 10,000 square meter flat terrain and about 13,000 Shape3D nodes (here a Java3D Shape3D node refers to a polygon without texture). The objects used are realistic models of the city of Darmstadt in Germany. The server currently runs on a 133MHz SGI with 32MB

and IRIX 6.5. The DeepMatrix server has a practical limit of 8-10 users assuming that not everybody moves and performs distributed tasks at the same time [17]. The scalability for this system, which is based on DeepMatrix, is yet to be tested. However, in most cases for a planning scenario as shown in figure 3, there might be only about 10 simultaneous users working on the system assuming 1-2 planners, 1 architect, 1 engineer, 1 constructor, 1-2 public and 1 applicant.

4. Experimental results

Three examples of development control checks that can be performed with the CUP system are presented below. The scenario used is the analysis of a proposal to construct a new house and garage in a residential area.

4.1 Correct placement of the proposed house and privacy check

This is an example task where the user, e.g. planner, initially inserts the computer model of the proposed house and garage as supplied by the applicant (*see* Figures 9a & 9b). The location of the house and garage can be adjusted by first selecting the model and then rotating, translating or scaling it as required using the mouse (*see* Figure 9c).

Figure 9a: Proposed site for the example house and garage
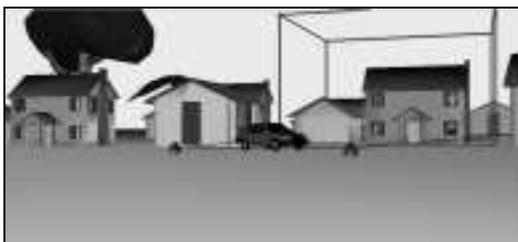

Figure 9b: After inserting the proposed house and garage

Figure 9c: After scaling, rotating and translating the house and garage

Privacy intrusion checks can be performed to make sure that the house is not placed too close to its neighbouring houses (*see* Figure 9d). For example, in UK there is a requirement of 9 meters boundary distance around the house. A backdrop with a

digitised image of the existing surrounding can also be applied to give a more realistic view of how the house might affect the environment in reality (*see* Figure 9e). These techniques can help to analyse the impact of the proposed house and garage on its surroundings and find the most suitable location for it.
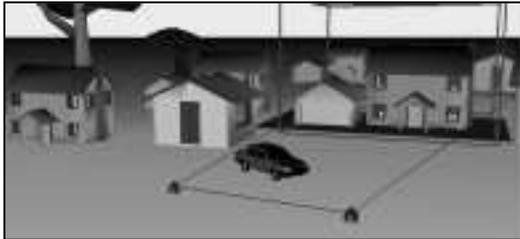

Figure 9d: Privacy area check   **Figure 9e: Inserting backdrop**

4.2 Daylight and overshadow check

According to the basic UK guides to day lighting, anything within a 22.5 ^{0} angle from a point 2 meters above the floor of the house is going to affect the day lighting of that house. Based on this guide the system generates a plane showing the rays falling on the house when information about the position of the sun and direction of its rays are input (*see* Figure 10a). If this plane intersects another object, it indicates that this object possibly blocks sunlight to the house (*see* Figure 10b). This is a crude way but it would immediately show planners if there was likely to be a problem. The same technique can be used in the reverse to check for overshadow i.e. whether this house casts a shadow on any other neighbouring houses. Again, alterations to the initial design such as reducing the height of the house or repositioning can be considered.
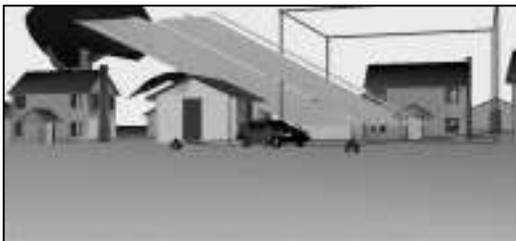
**Figure 10a: Plane showing   Figure 10b: Plane showing
uninterrupted sunlight   interrupted sunlight**

4.3 Visibility checks

By selecting two points in space and initiating a visibility check, the system draws a line connecting the two points and also determines the intersections of this line with the objects in the world. If there are no intersections the system displays "Point 2 is visible from Point 1" in the text field provided on the 'Navigate panel' else it displays "Point 2 is not visible from Point 1" (*see* Figure 11a). In addition the system also moves the user's viewpoint to the first point and aligns their view in the direction of the line to give a better impression of what it might look like in reality. Visibility analysis can be useful to planners to make sure that the house does not obstruct the view to important landmarks from critical locations. A similar check to determine what

objects are hidden behind the house can also be done by making the house partially or fully transparent (*see* Figure 11b).

Figure 11a: Visibility check from point 1 to point 2
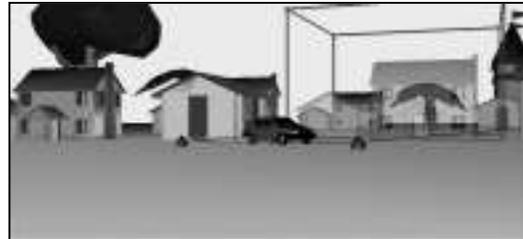
Figure 11b: Objects behind are visible when the house is made transparent

5. Conclusion

The aim of this research has been to consider how the utilization of CVE technology might actually help to improve contemporary urban planning, particularly the activities of development control and decision-making during the permit application process. The paper introduces the architecture of the CUP system that is being developed to provide a collaborative, interactive and location-independent working environment for several different types of participants generally involved in a planning project. The paper has also presented an application scenario, which offers a vision of future urban planning practice using the system. Rapid and cost-effective solutions to various development control tasks are provided by different analysis tools embedded within the system. The interface has been designed to be easy to use, intuitive and efficient. Some experimental results that demonstrate urban development control tasks performed within this environment are also presented.

The research has considerably benefited from a number of meetings and semi-structured interviews held with planners from the Edinburgh City Council (ECC), Edinburgh World Heritage Trust (EWHT) and Edinburgh College of Art (ECA). The application scenario will be tested with selected user groups, consisting of planners from ECC, EWHT and ECA, to influence the further development of the methodology and approach, and to measure its usability and efficiency.

CUP is a prototype that demonstrates the viability of using CVEs for urban planning. To take this conception further will require building upon this basic prototype to develop a production system that provides a more comprehensive solution for planners to perform a wider range of urban planning tasks within it. It also requires consideration of the relationship between planning aids such as CUP and the management systems that store the city models it utilises. A virtual city model service can be also be used to support a diverse set of alternative applications such as virtual tourism, information centres, traffic simulation, air pollution modelling, historic reconstruction and virtual heritage. The authors wish to investigate the prospects of integrating spatial databases held within a GIS with aids like CUP to make use of GIS's storage and information processing capacities. Techniques such as culling, dynamic loading and region partitioning will be looked into to increase scalability. Load sharing between servers to increase the number of users is another area that is going to be considered. Potential to use radiosity computations within particular regions for specified light sources will also be examined [23]. Currently the system

provides detailed object information with hyperlinks to additional information and photos. The benefit of adding other multimedia content such as demo videos and audio will be investigated. It will also be useful to identify the added advantages of including video-conferencing for improving communication among users.

References

[1] Allmendinger, P., A. Prior, and J. Raemaekers, Introduction to Planning Practice, Wiley, 2000.

[2] S. Benford, J. Bowers, L. Fahlen, J. Mariani, and T. Rodden, "Supporting Co-operative work in Virtual Environments", The Computer Journal, Oxford University Press, 1994, Vol.37, No.8.

[3] V. Bourdakis, "On Developing Standards for the Creation of VR City Models", The 19th ESAADE Conference, Helsinki, August 2001.

[4] W. Broll, E. Meier, and T. Schardt, "The Virtual Round Table- A Collaborative Augmented Multi-User Environment", in Proc. of the ACM Collaborative Virtual Environments (CVE 2000), San Francisco, ACM: New York, 2000, pp.39-46.

[5] T. Broyd, "The Impact of IT on Design and Construction", in Proc. Of ICE, Civil Engineering, May 2000, paper 12136, pp.409-415.

[6] C. Carlsson, and O. Hagsand, "DIVE- A Platform for Multi-User Virtual Environments", Computers and Graphics, 1993, Vol.17, No.6, pp.663-669.

[7] V. Coors, U. Jasnoch, and V. Jung, "Using the Virtual Table as an Interaction Platform for Collaborative Urban Planning", Virtual Reality & 3D GIS, Computers & Graphics, Elsevier Science Ltd., 1999, Vol.23, pp.487-496.

[8] M. Dodge, S. Doyle, A. Smith and S. Fleetwood, "Towards the Virtual City: VR & Internet GIS for Urban Planning", Virtual Reality & Geographical Information Systems, Birkbeck College, 22nd May 1998.

[9] C. Greenhalgh, and S. Benford, "MASSIVE, A Collaborative Virtual Environment for Teleconferencing", ACM Transactions on Computer, Human Interaction (TOCHI), ACM Press, 1995, Vol.2, No.3, pp.239-261.

[10] C. Greenhalgh, and S. Benford, "Boundaries, Awareness and Interaction in Collaborative Virtual Environments", in Proc. of the 6th International Workshop on Enabling Technologies: Infrastructure for Collaborative Enterprises, Cambridge, Massachusetts, USA, 1997.

[11] B. Huang, and H. Lin, "GeoVR: A Web-Based Tool for Virtual Reality Presentation from 2D GIS Data", Computers & Geosciences, Elsevier Science Ltd., 1999, Vol.25, pp.1167-1175.

[12] R. Hubbold, J. Cook, M. Keates, S. Gibson, T. Howard, A. Murta, A. West and S. Pettifer, "GNU/MAVERIK: A Micro Kernel for Large-scale Virtual Environments", Presence: Teleoperators & Virtual Environments, Feb 2001, Vol.10, pp.22-34.

[13] T. Manoharan, H. Taylor and P. Gardiner, "Interactive Urban Development Control with Collaborative Virtual Environments", to be published in Proc. of Virtual Systems and Multimedia Conference, VSMM2001, IEEE Computer Society, 2001.

[14] V. Normand, and J. Tromp, "Collaborative Virtual Environments: The COVEN Project", in Proc. Of the Framework for Immersive Virtual Environments Conference, FIVE'96, Pisa, 1996.

[15] S. Pettifer, J. Cook, J. Marsh, and A. West, "Deva3: Architecture for a Large Scale Virtual Reality System", In Proceedings of ACM Symposium in Virtual Reality Software and Technology 2000, ACM Press, Oct 2000, pp. 33-39.

[16] S. Pettifer and J. Marsh, "A Collaborative Access Model for Shared Virtual Environments", In Proceedings of IEEE WETICE 01, IEEE Computer Society, June 2001, pp.257-272.

[17] G. Reitmayr, S. Carroll, A. Reitemeyer, and M.G. Wagner, "DeepMatrix – An Open Technology Based Virtual Environment System", The Visual Computer Journal, 1999, Vol.15, pp.395-412.

[18] P. Selman, D. Davidson, A. Watson, and S. Winterbottom, "GIS in Rural Environmental Planning: Visual and Land Use Analysis of Major Development Proposals", Town Planning Review, 1991, Vol.62, No.2, pp.215-223.

[19] S.J. Simoff, and M.L. Maher, "Analysing Participation in Collaborative Design Environments", Design Studies, Elsevier Science Ltd., 2000, Vol.21, pp.119-144.

[20] H.A. Sowizral and M.F. Deering, "Projects in VR", Sun Microsystems, IEEE Computer Graphics and Applications, May/June 1999.

[21] Thomas, K., Development Control – Principles and Practice, UCL Press, London, 1997.

[22] A. J. West and R. J. Hubbold, "System Challenges for Collaborative Virtual Environments", In D. Snowdon and E. Churchill, editors, Proceedings of Collaborative Virtual Environments '98. Springer-Verlag 2000, Keynote Speech.

[23] Distributed Ray Tracer for Java3D, http://www.dcs.kcl.ac.uk/teaching/units/cs2grs/graphics_projects_01_02.html