

CONFERENCE THEME: VISUALISATION AND INFORMATION

Full paper

SUPPORTING COLLABORATION AND MULTIPLE VIEWS OF BUILDING MODELS IN VIRTUAL WORLDS

Mary Lou Maher

Key Centre of Design Computing and Cognition, University of Sydney
mary@arch.usyd.edu.au

Mike A Rosenman

Key Centre of Design Computing and Cognition, University of Sydney
mike@arch.usyd.edu.au

Lan Ding

CMIT, Commonwealth Scientific and Industrial Research Organisation
lan.ding@csiro.au

Gregory J Smith

Key Centre of Design Computing and Cognition, University of Sydney
g_smith@arch.usyd.edu.au

David Marchant

Woods Bagot
David.Marchant@woodsbagot.com.au

Andy Dong

Key Centre of Design Computing and Cognition, University of Sydney
andy@arch.usyd.edu.au

ABSTRACT

Collaboration in virtual environments has the potential to support communication and collaboration in multi-disciplinary teams. Current software for documenting and developing models of buildings focuses on supporting a single user who is a specialist in the specific software used within their own discipline. In this paper we report on the research, and a proposed approach, in supporting multiple disciplines in virtual worlds. We emphasise that the views and models are founded on the functional concerns of designers and the concerns are discipline specific. An agent-based approach to representing multiple views is described, along with a virtual world environment that supports communication and collaboration. In order to prototype and develop a collaborative environment, we consider simple models in an inherently multi-user virtual world.

Keywords: Collaboration, Multiple Views, Virtual Worlds, Agents

1. INTRODUCTION

Current software tools for documenting and developing models of buildings focus on supporting a single user who is a specialist in the specific software used within their own discipline. Such tools tend not to be multidisciplinary and not foster collaboration. Collaboration in distributed virtual environments has the potential to support communication and collaboration in multidisciplinary teams if a suitable design infrastructure could be added. Collaborative Virtual Environments (CVE) provide one kind of infrastructure to support communication and collaboration.

A Collaborative Virtual Environment (CVE) is a computer-based, distributed, virtual space or set of places. In such places, people can meet and interact with others, with agents or with virtual objects. CVEs might vary in their representational richness from 3D graphical spaces, 2.5D and 2D environments, text-based environments, to combinations of these (Li and Maher, 2000). Figure 1 shows, by way of example, the prototype CVE that we are developing. To facilitate distributed collaborative designing, a number of features are required. These can broadly be classified as either facilitating artefact modelling or as facilitating communication and collaboration.

CVEs, such as the Active Worlds¹ one shown in Figure 1, facilitate communication and collaboration by providing mechanisms for shared context, awareness of others, visual and textual communication, immersive manipulation of constructed artefacts, and a fostering of community design. What current CVEs do not provide are mechanisms for reasoning about artefacts and the representation of different views or properties by practitioners from different disciplines. This paper reports on the research, and a proposed approach, in supporting reasoning about, and representation of, different views in virtual environments.

¹ <http://www.activeworlds.com>

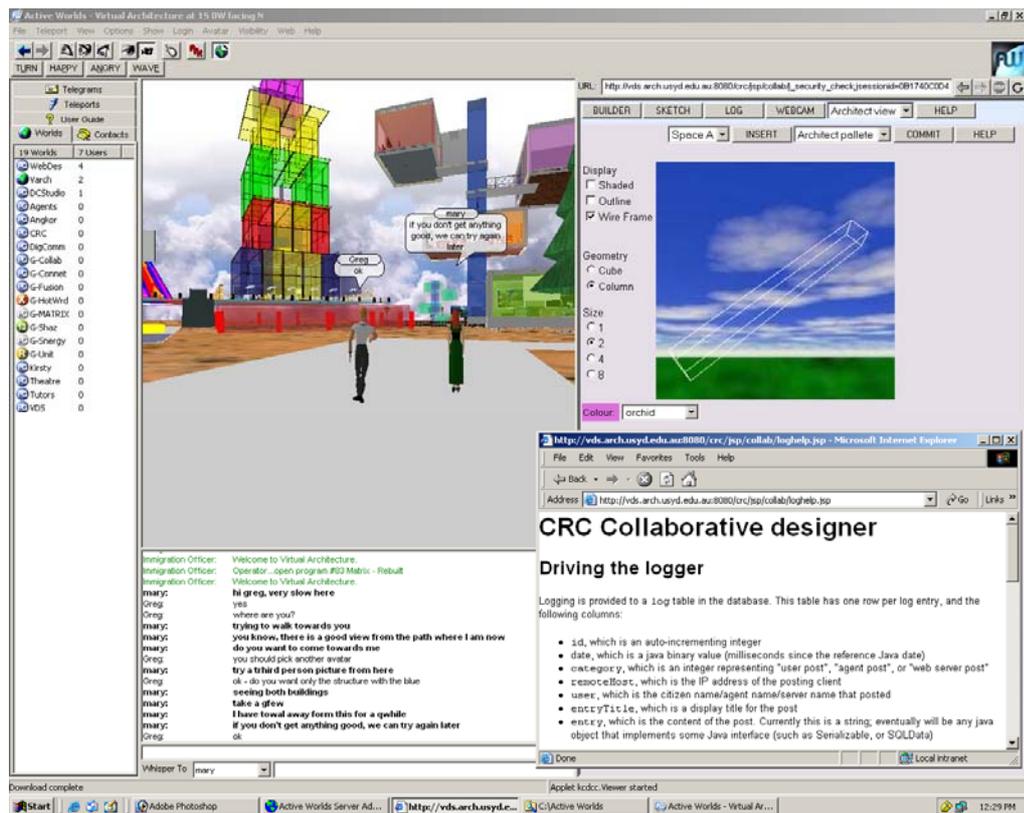


Figure 1. Virtual Worlds and the prototype CRC Collaborative Designer

In this paper, Section 2 considers representing the multiple views of a single design artefact that may be held by practitioners from different disciplines involved in a project. Discipline models allow each discipline to work according to its own concepts and representations, with a common model providing for the integration required between models. Section 3 describes an approach where role-based agents provide the reasoning mechanism that connects the visualization of the objects in the virtual world with the discipline and common artefact models in a database. Section 4 introduces a scenario to illustrate these ideas using simple conceptual design artefacts.

2. REPRESENTING MULTIPLE MODELS

A model of an object is a representation of that object resulting from a particular view taken. Given a design object, such as a building, there are many views that may be taken, leading to different conceptual interpretations. For example, a building may be viewed as a set of activities that take place in it; as a set of spaces; as sculptural form; as an environment modifier or shelter provider; as a set of force resisting elements; as a configuration of physical elements; etc. A building is all of these, and more. For each different view of a building there will be a corresponding model, as illustrated in Figure 2.

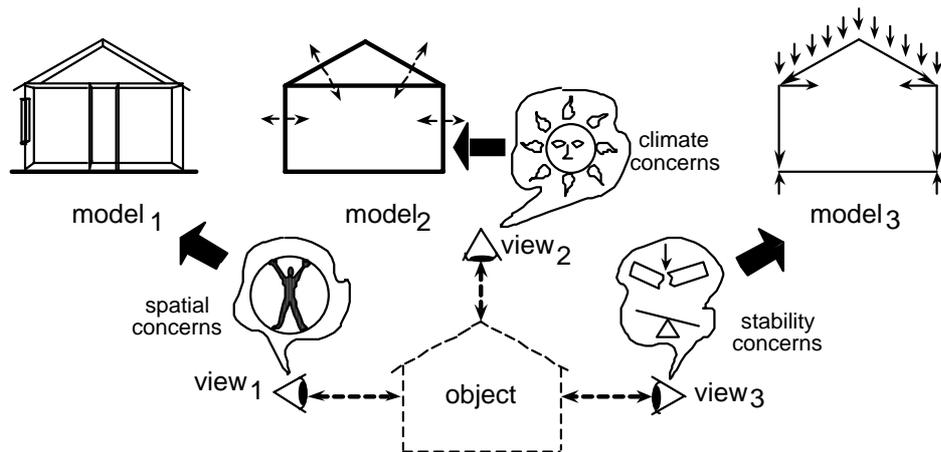


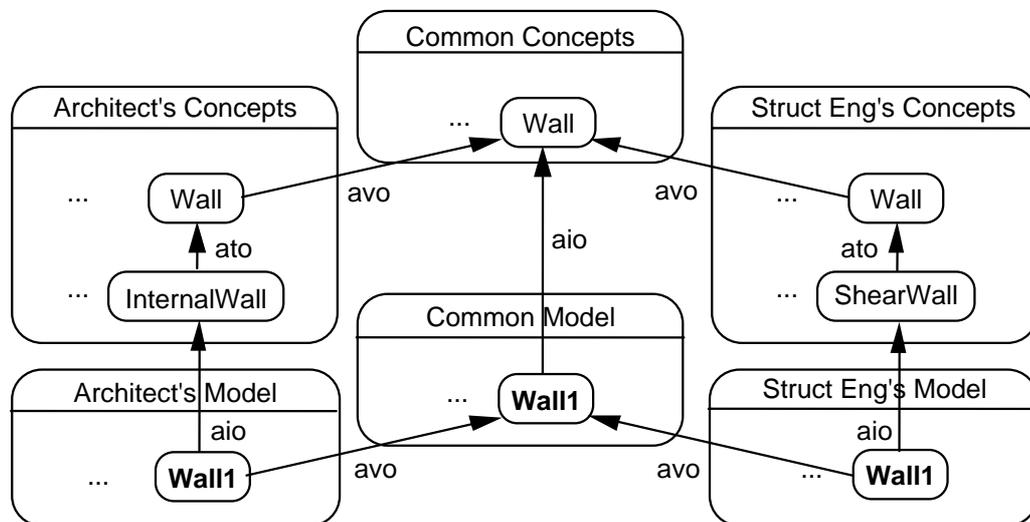
Figure 2. Multiple views and models.

Depending on the view taken, certain properties and descriptions of the object become relevant. The sound insulating properties of a wall are not relevant to a structural engineer's description of that wall. In fact, many walls may not be relevant at all to a structural engineer if they do not either contribute directly to the stability of a building or indirectly by providing a substantial load. The architects will model certain elements such as floors, walls, doors and windows. For the architects, these elements are associated with the spatial and environmental qualities with which they are concerned. Structural engineers, however, see the walls and floors as elements capable of bearing loads and resisting forces and moments. Both models must coexist since the two designers will have different uses for their models. For example, the structural engineers will need to carry out calculations based on their model while the architects may need to ascribe different properties to their separate wall elements. According to Bucciarelli (2003) "There is one object of design, but different object worlds." And "No participant has at any stage in the process, a comprehensive, all-encompassing understanding of the design. No participant has a 'god's eye view' of the design." The engineers may modify some of the properties assigned to elements by the architect and may add some new elements, such as beams and columns. The addition of such new elements may affect the architect's model (and vice versa). Any such decisions taken by the engineer must be conveyed to the architect by making changes in the architect's model as appropriate. It will be shown that such changes in another discipline's model can be done when the change affects a function which is the concern of that discipline.

There exists considerable work using a single model approach based on the construction of a model from 'primitive' elements from which multiple interpretations are derived (Howard et al., 1992; Amor and Hosking, 1993; Clayton et al., 1994; MacKellar and Peckham, 1994). This approach is analogous to the formation of views in database management systems. However, it is argued that this approach is insufficient, since the 'primitive' elements themselves are subject to the views taken by the different viewers and hence different primitive models are constructed by each such viewer (Rosenman et al. 1993; Rosenman and Gero, 1996). Since the basic description of an object differs from viewer to viewer, each viewer may represent an object with different elements and different composition hierarchies. For example, while architects may model walls on different floors as separate elements, bounding various rooms, the structural engineers may model only a single shear wall. So that, not only is the interpretation of the meaning of a design object different from one viewer to

another but, also the description of the structure of the object differs. There exists no single unified model or even a single set of unique elements but rather different descriptions of the same elements and different subsets of these descriptions in different models. The disciplines have their own discipline models and there exist a number of core models which contain the common concepts between disciplines. We refer to this approach as the “common model” approach. It is similar to that taken by Nederveen (Nederveen, 1993; Nederveen and Tolman, 1992) and also Pierra in his work on PLIB (Pierra, 1993; Sardet et al, 1998).

Although each discipline creates and maintains its model, the various models refer to the same object and, therefore, must be consistent. Using the common model approach, this is achieved through the common models using the a_view_of relationship (Pierra,1993; Sardet et al., 1998), as illustrated in Figure 3.



ato = a_type_of aio = an_instance_of avo = a_view_of

Figure3. Communicating across models through the a_view_of relationship

The discipline concepts (classes) hold those properties specific to the discipline. For example, the Wall (architect) concept may have a function that is to bound a space, while the Wall (structural engineer) concept may have optional functions of supporting elements and/or transferring lateral loads. The Wall (common) concept will have the structure properties, such as the material, shape, dimensions and location. Since both the architect and engineers Wall are a_view_of the common Wall, they will inherit properties from the common Wall. When either the architect or engineer creates a wall object (such as Wall1) in their model, a corresponding instance of the common class is created and the discipline object is related to the common object through the a_view_of relationship. Wall1 may be an instance of a Wall class either directly or as an instance of a Wall subclass such as InternalWall.

3. SUPPORTING COLLABORATION AND MULTIPLE VIEWS

Our approach to supporting collaboration and multiple views augments the inherently multi-user platform provided by the Active Worlds platform with additional collaboration tools (such as webcam, sketching whiteboard, etc); an object database; and an agent system to facilitate multiple views functions. The components are shown distributed across a network but need not be. Figure 4 shows a system architecture depicting our approach. The major components are:

- A client browser to a virtual environment provides the CVE experience to the designers with real time rendering to reflect changes immediately.
- A virtual world server provides the interactive 3D world to the CVE, as well as interactive chat.
- A 2D HTTP server enables server-side scripting of the web pages that provides information from the database, access to tools like a webcam, iphone, and sketching board.
- A database that holds the discipline models, the common model, mappings between these models and virtual world objects, and other data to facilitate collaboration as well as provide the basis for translating to and from specific CAD models.
- A multi-agent system to monitor and maintain the information in the servers and the database, providing a unified experience to designers.

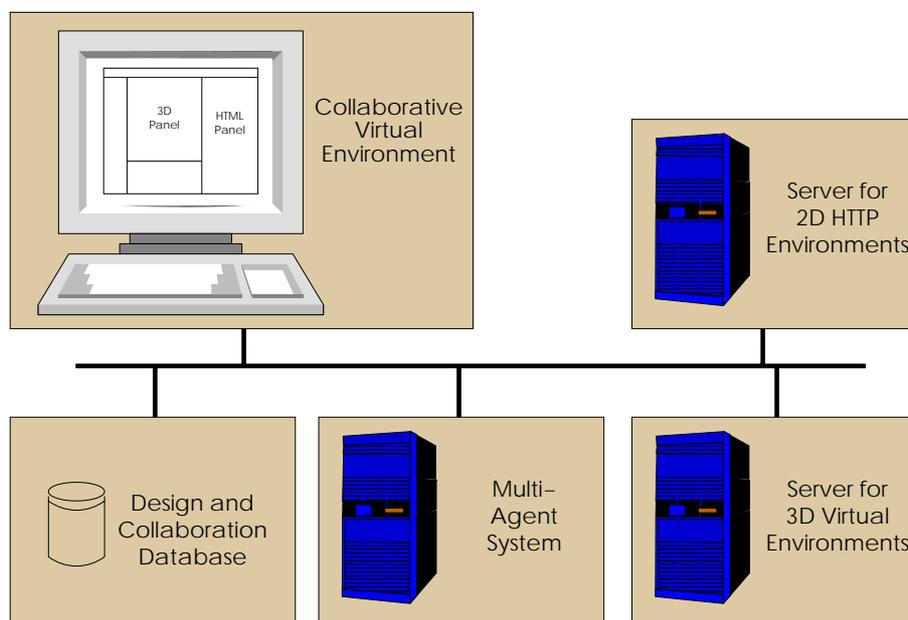


Figure4. System architecture

The primary role of the agents is to construct and maintain multiple views, both abstractly of the design artefact, and concretely of 3D objects instantiated in a virtual world. Haymaker et al. (2000) describe how agents that play roles as filters, constructors or mediators interpret semantic database entities. Our agents play similarly classified roles except that the

database entities that our filter agents interpret and update are the discipline and common models of Section 2.

Filter agents filter the view of the discipline and common models. The filtered view presented to constructor agents changes according to the disciplines of the designers that are currently collaborating and their current interest. Filter agents also provide the interface between the 3D objects from which the 3D virtual worlds are constructed and the database objects that comprise the discipline and common models.

Constructor agents build design artefacts in a 3D world. They insert 3D objects to build artefacts, as well as moving and deleting 3D objects to reconstruct the space of the design artefact. Mediator agents associate 3D objects with designers and their 3D world avatars, handle text chat from designers to agents, handle remote method invocations from the HTTP server, control the work flow between filter and constructor agents, and enable session data to be logged. Data collector agents provide for logging and data collection for later cognitive and data mining analysis or simply as a record of important collaborative sessions.

Figures 5 and 6 show two typical interactions of the agents. Figure 5 shows an architect adding a new wall to an existing building. The architect sets parameters on the 2D build panel and clicks an insert button. This results in a message being sent via the HTTP server to the mediator agent. The mediator decides that construction is required and adds a new 3D object to the world, with selection of the object and its location being parametric. If successful, an identifier of the new object is returned by the virtual world server, which the mediator passes to the filter agents along with the other parametric details. The filter agents use these to update the models in the database. The virtual world server also sends update messages to each client browser to update their 3D views.

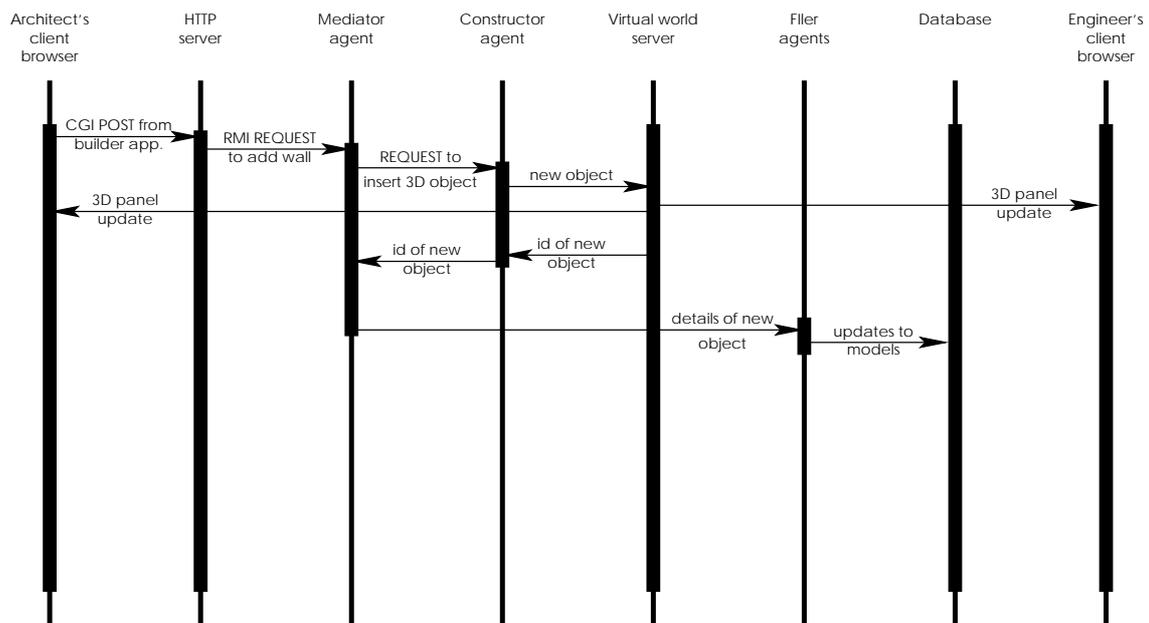


Figure 5. Interaction diagram illustrating an architect adding a new wall to the design artefact.

Figure 6 shows the interaction diagram for an engineer entering the world and changing the view of the building presented from that of an architect's view to that of an engineer's view. At any time the 3D view of the design artefact will be that of one of the disciplines. When a designer requests a change of view the mediator requests that the constructor change each object accordingly. What each change is will depend on the view and the object. For a change from an architect's view to an engineer's view, some spatial objects may be re-rendered such that they are invisible as spatial concepts are generally not of concern to a structural engineer. Similarly, some other objects may become visible. Other objects that are of interest to both disciplines, but to different degrees, may change visual properties such as by changing colour, texture or transparency.

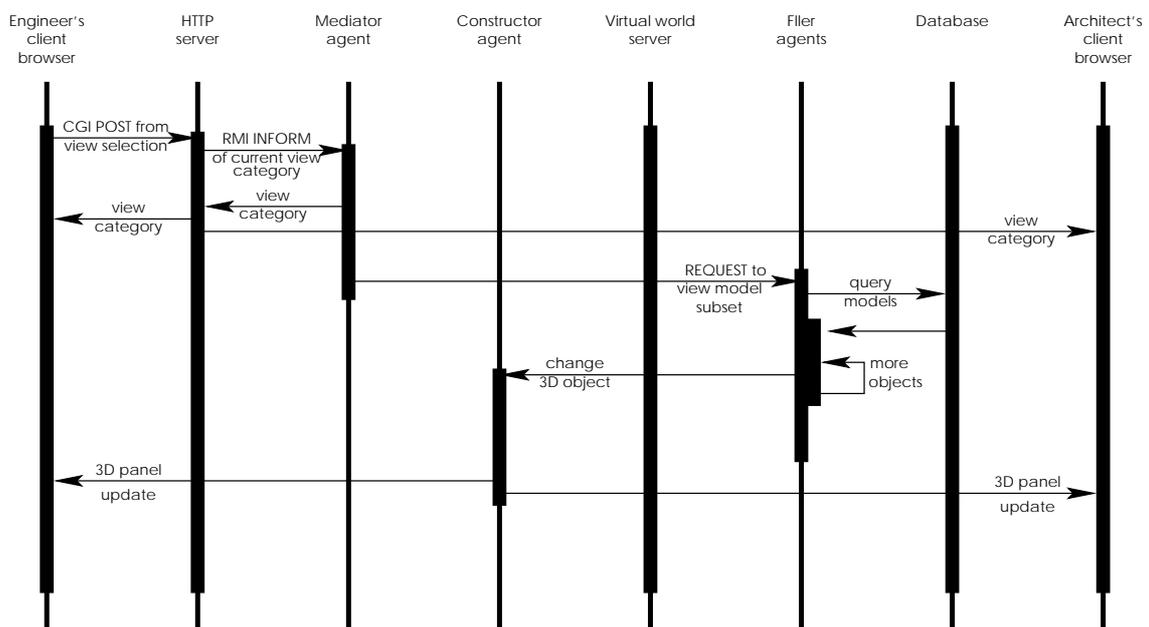


Figure 6. Interaction diagram illustrating an engineer changing the current view from that of the architects.

4. PROTOTYPE IMPLEMENTATION AND SCENARIO

Figure 1 shows one view of the client browser on our prototype implementation. In this implementation, Active Worlds is used as the 3D virtual world server and their browser is the basis for the CVE. The long vertical panel on the left of the client browser shows the 3D worlds known to the 3D world server. Each designer is represented in their current world by an avatar, as seen in the large panel to the right of the worlds panel. This shows a 3D view from the viewpoint of the avatar of the person logged in on this browser. It is this panel, along with sound and chat, that provides the 3D immersive experience. The panel below the 3D view panel facilitates chat. To the right of the 3D view panel is a panel that shows dynamically served web pages that provide more information about the design or run interactive applications.

The right panel window shown in Figure 1 contains row of buttons at the top, each of which launches one application. Currently these applications include

a builder, a distributed sketchpad, an interface to the data logger, a web cam with audio, and help pages. Figure 1 is showing the builder application. Help pages launch as separate overlay windows, as shown in Figure 1.

The sketchpad is an applet that provides for freehand sketching over a background image. The background image may be uploaded and so designers in different locations can sketch over a 3D view of the design artefact. The current sketch and background image are saved in the database such that designers can work on the same sketch.

The web cam page shows live images of the online collaborating designers, plus will provide for audio after the manner of Voice-over-IP (VoIP). This is to facilitate collaboration over a distance using materials not otherwise catered for in a virtual environment, such as showing books, magazines, real world models, and so on.

The agents are programmed in Java and C, and communicate with the HTTP server using Java remote method invocation (RMI). Agents interact with the Active Worlds server using the AW software development kit (SDK) using a java native interface (JNI). The mediator agent connects itself to the system as an RMI server with which to receive messages from the HTTP server. It also registers itself as a callback with which to receive textual chat from designers and avatar events from the virtual world. The mediator interprets these messages and forwards them to the filter and constructor agents. The constructor agents also register themselves as a callback with which to receive 3D object add, change and delete events. Filter agents communicate with the database. Currently queries are in SQL via Java database connectivity (JDBC); future versions may instead use XQuery on a native XML database. Previous research (Maher et al. 2003) has demonstrated how IFCs can be used to populate a database from an ArchiCAD model, and how this database can be used by agents to construct views in Active Worlds. Following from this, our database will hold the discipline models, the common model, mappings between these models and the virtual world objects, and other that facilitates collaboration.

In order to prototype this collaborative environment we consider only simple 3D models, although the database is being constructed with translation to and from existing CAD tools in mind. We illustrate the prototype with a scenario in which an architect and a structural engineer insert simple models to communicate their conceptual design ideas to each other. The architect inserts two zone objects, showing how the two major spaces of the building are connected. These objects are shown in Figure 7 (a). The engineer inserts two wall objects, showing where the major load bearing systems will be located with respect to the spatial elements. The engineer's objects are shown in Figure 7(b). Figure 7(c) shows a view that display both the architect's and the engineer's models.

In this scenario we show the visual interface using different colours and types of objects that indicate models from the different disciplines. The database represents each of these objects and includes properties that record the discipline that is associated with the object, the location of the object and the basic geometric properties of the object. The agents' roles include filtering the objects to be displayed, recording the changes to the database to create new

objects or modify existing objects, and to manage the insertion of new objects into the AW server.

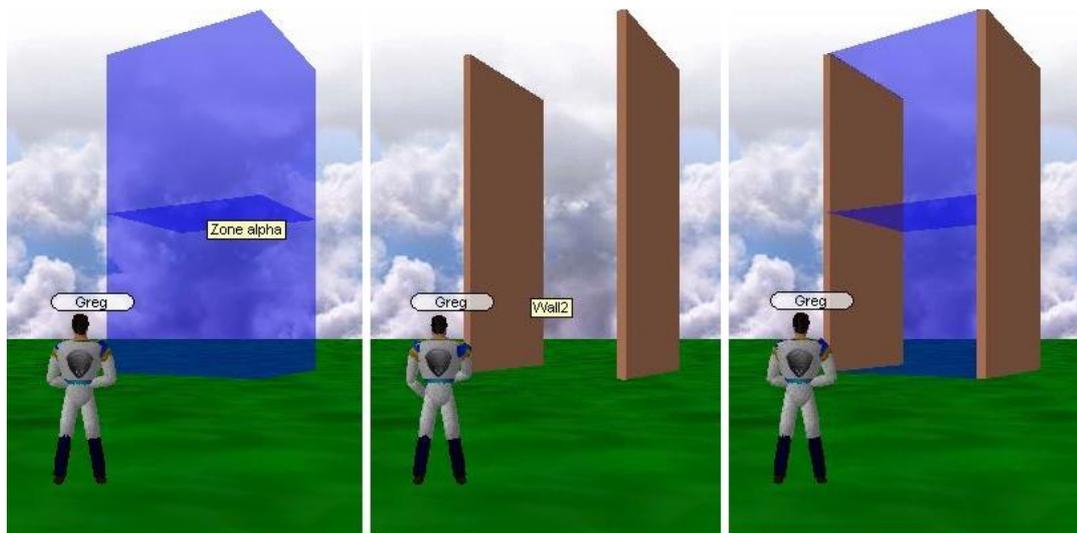


Figure 7. Collaboration between an architect and an engineer. (a) The architect inserts a zone denoted "alpha". (b) The engineer inserts wall objects denoted "Wall1" and "Wall2" (c) Combined view.

5. CONCLUSION

This paper has described research on augmenting existing virtual world platforms to support collaborative multidisciplinary design, with an emphasis on developing a representation and scenario that facilitates communication of function through simple forms. We take advantage of the distributed immersive 3D environment to facilitate communication, enhancing it with a distributed sketchpad, web cam and audio. Supporting multiple disciplines is handled by a multi-agent system that maps between a database of discipline and common models and a 3D view being presented in a virtual world. We describe here a prototype implementation that is the basis for ongoing development and cognitive studies of designers.

REFERENCES

- Amor, R.W. and Hosking, J.G. (1993). Multi-disciplinary views for integrated and concurrent design, *In* Mathur, K.S., Betts, M.P. and Tham, K.W. (eds.), *Management of Information Technology for Construction*, World Scientific, Singapore 255-267.
- Bucciarelli, L. L. (2003). Designing and learning: a disjunction in contexts, *Design Studies*, **24**(3): 295-311.
- Clayton, M.J., Fruchter, R., Krawinkler, H. and Teicholz, P. (1994). Interpretation objects for multi-disciplinary design, *In* Gero, J.S. and Sudweeks, F. (eds.), *Artificial Intelligence in Design '94*, Kluwer Academic Publishers, Dordrecht, Netherlands, 573-590.
- Haymaker, J, Ackermann, E, and Fischer, M. (2000). Meaning mediating mechanism, *In* Gero, J.S. (ed.), *Artificial Intelligence in Design '00*, Kluwer Academic Publishers, Dordrecht, Netherlands, 691-715.
- Howard, H.C., Abdalla, J.A. and Phan, D.H. (1992). Primitive-composite approach for structural data modeling, *Journal of Computing in Civil Engineering*, **6**(1):19-40.
- Li, F. and Maher, M.L.(2000), Representing virtual places - A design model for metaphorical design, *ACADIA2000*.

- Maher, M.L., Liew, P-S., Gu, N. and Ding, L. (2003). An agent approach to supporting collaborative design in 3D irtual worlds, *Proceedings of eCAADe*, to appear.
- Mackellar, B.K. and Peckham, J. (1994). Specifying multiple representations of design objects in SORAC. In Gero, J.S. and Sudweeks, F. (eds.), *Artificial Intelligence in Design '94*. Kluwer Academic Publishers, Dordrecht, Netherlands, 555-572.
- Rosenman, M.A., Gero, J.S. and Hwang, Y-S. (1993). Representation of multiple concepts of a design objects based on multiple functions. In Mathur, K.S., Betts, M.P. and Tham, K.W. (eds.), *Management of Information Technology for Construction*, World Scientific, Singapore, 239-254.
- Rosenman, M.A. and Gero J.S. (1996). Modelling multiple views of design objects in a collaborative CAD environment, *CAD Special Issue on AI in Design*, **28**(3): 207-216.
- Nederveen, S.V. (1993). View integration in building design. In Mathur, K.S., Betts, M.P. and Tham, K.W. (eds.), *Management of Information Technology for Construction*, World Scientific, Singapore, 209-221.
- Nederveen, G.A. van and Tolman, F.P. (1992). Modelling multiple views on buildings, *Automation in Construction*, **1**:215-224.
- Pierra, G. (1993). A multiple perspective object oriented model for engineering design. In Zhang, X. (ed.), *New Advances in Computer Aided Design & Computer Graphics*, International Academic Publishers, Beijing, China, 368-373.
- Sardet, E, Pierra, G, Poiter, J. C, Battier, G, Derouet. J. C, Willmann, N and Mahir, A. (1998). Exchange of Component Data: The PLIB (ISO 13584) Model, Standard and Tools, *Proceedings of the CALS EUROPE'98 Conference*, pp 160-176.