AUGMENTATION OF REAL-TIME 3D VIRTUAL ENVIRONMENTS FOR ARCHITECTURAL DESIGN AT THE CONCEPTUAL STAGE

SUBMITTED: February 2008
REVISED: July 2008
PUBLISHED: December 2008
EDITOR: John Messner

Rabee M. Reffat, *Associate Professor*
Architecture Department, KFUPM, Dhahran, Saudi Arabia
rabee@kfupm.edu.sa and http://faculty.kfupm.edu.sa/arch/rabee

Yaman Khaeruzzaman, Emad El-Sebakhy and I Putu Raharja
Information and Computer Science Department, KFUPM, Dhahran, Saudi Arabia

SUMMARY: *This paper presents a computational system for augmenting real-time 3D virtual environments to support the formation and compositions of architectural designs. The developed IAMVE (Interactive Architectural Modeling in Virtual Environments) system provides more flexibility to architectural designers and develops an edge to multi-user real time 3D virtual environments to be better utilized in the context of architectural design. The features of IAMVE system include: (a) providing a set of 3D architectural objects that are made available within the virtual environment and can be utilized by the user to create architectural design compositions; (b) allowing users to modify both geometrical and non-geometrical properties of these objects; and (c) maintaining interrelationships between these architectural objects using constraint-based rules automated by the developed system in real-time. These rules include connecting the edges of non-parallel walls that have gaps within the distance of 50 cm, maintaining an assembly hierarchy between the wall and its doors and windows, and maintaining an assembly hierarchy between the window and its shading devices.*

KEYWORDS: *3D Virtual Environments, 3D Modeling; Architectural Compositions, User Interactivity.*

1. INTRODUCTION

3-D immersive virtual environments are at the forefront of ‘the physical Internet’ (Scola, 2006). Virtual environments allow users to communicate with each other through avatars, their digital personas, rather than relying solely on text. Virtual environments provide an alternative ‘space’ that supports many human activities including design of artifacts. A large number of architecture and design schools are currently utilizing virtual environments as presentation platforms in their virtual architectural design studios. Virtual Environments can either support teaching in a single studio within an institution or bring together students from several institutions. There are various motivations for engaging architecture students in virtual design studios including presenting an essential learning for practice of the future, exploiting technology in design teaching, researching the nature of design communication and processes, and searching for ways to improve the educational experience of students (Kvan, 2000). Furthermore, virtual environments allow designers to visually walk through, inspect and present the designs in an immersive 3D environment at the proper size and scale. Attributes of a multi-user virtual environment platform include community access, client access and control, and scope for feedback (opinion) in addition to participation. A multi-user platform also provides “traditional”, familiar collaborative Internet technologies, e.g. file and document sharing (Sherman and Craig, 2003). Augmenting, evaluating and studying virtual environments and evolving communities give rise to a collaborative memory space, of benefit to every user adding to and participating in it (Schnabel and Kvan, 2001).

Virtual environments provide powerful communication and navigation environments wherein users can collaboratively design in centralized or distributed environments. Some examples in this field include “Phase X” (Schmitt, 1997) that is a design course at ETH, Zurich which starts using the computer as a medium but in a passive approach. “Phase X” expanded on the idea of paperless studio by building more dimensional computer models, networking the designs and focusing on abstract concepts. The concept of space as an element has been developed in "Sculptor" (Kurmann et al, 1997) to enable the designer to design interactively in real time with the computer in a 3D environment. The combinations of solid and void elements, positive and negative volumes, enabled the designer to facilitate the computer at the early stages of conceptual design. Another application is "roomz" (Strehlke and Engeli, 2001); that is a workplace called "Myscenario" which allows three types of interactions: changing the wall colors, placing objects within the space and creating a path through the space.

The computational value for efficiency, visualization and communication are evident in many advanced CAD (computer aided design), systems in which 3D models can be generated, rendered and animated. 3D virtual environments in design domains have been able to mimic the spatial configuration of physical worlds, changing the role of CAD systems, partly, from being drafting tools to producing the building blocks of these new environments (Maher et al, 1999). 3D Virtual environments facilitate a level of communication and collaboration not readily available in conventional CAD systems. The integration of virtual environments and CAD systems using common data model can make a significant impact on synchronous collaboration and real time multi-user multi-disciplinary modification of building data. Early attempts such as Maher et al (2003) use agents to assist design collaboration in 3D virtual environments. Maher et al (2003) introduced the use of agents for the integration of 3D virtual environments and CAD systems to go beyond passive data transfer whereby agents can respond to changes in the CAD system or 3D virtual environment in the form of an update to the geometry, or as a recommendation to change non geometric information or to propagate changes to other parts of the design. This approach is applied using ArchiCAD as the CAD system, Activeworlds as the virtual environment platform, and the Express Data Manager is used as the central repository for storing the representation of the relevant data model. A multi-agent system is developed to connect the virtual environment to this database to allow active data sharing.

This paper extends earlier work outlined above and specifically targets architectural design at the conceptual stage. The paper introduces a system that replaces the integration of CAD systems and virtual environments by embedding the required modeling features of CAD systems into the virtual environment, tailor them to architectural design requirement, and use constraint-based rules to maintain interrelationships of different design components in 3D virtual environments for architectural design. There are various 3D virtual environment platforms such as Activeworlds, Second Life and Cybertown that can benefit from the proposed work in this paper. Activeworlds has been around longer than many virtual environments, since its beta launch in 1995. It is a virtual environment that is similar to Second Life and Cybertown. Users of Activeworlds can build objects that are visually appealing, but lacking in functionality, especially compared to newer virtual environments such as Second Life. On the other hand, Second Life is a 3D environment with fully customizable avatars and a built-in scripting and building language. Users imagine and create all sorts of objects, including educational simulations and real-world products for sale. Second Life has been hailed by many for its educational potential. Many colleges and universities have purchased islands within Second Life and have built virtual campuses inside. While Activeworlds and Second Life are both virtual environments with similar capabilities, they can not be simply compared with one another, at least not in all aspects. Either of them seems to target and market towards different kinds of people. Second Life heavily emphasizes its economic and social aspects, while Activeworlds is all about building and creation. Obviously one can build/script in Second Life, but it feels like they are slowly drifting away from that in order to cater to a more mainstream audience; the ones that just want to hang out, chat, attend events, etc. Activeworlds licenses its virtual environment technology for private development, which is potentially useful for educational researchers, entertainment corporations, and others. The strongest point in Activeworlds is its building interface; Activeworlds objects, when rotated, follow their axis when they move about on the XZ plane. Second Life completely lacks this simple feature. Also, basic math and centering of axes works perfectly in Activeworlds compared to Second Life. However, parametric modeling with real-time geometrical and non-geometrical modifications is a feature of Second Life and a limitation in Activeworlds. Nevertheless, objects in both Activeworlds and Second Life are generic modeling and are not specifically objects for architectural design. Therefore Activeworlds have been used as a vehicle for implementing the concepts of this paper; however such concepts are also applicable to other 3D virtual environments including Second Life in order to be utilized for architectural design. Hence, the actual research problem addressed in this paper is to augment 3D virtual environments with new features that facilitate using them for collaborative architectural design at the conceptual stage. This paper presents the computational system for developing interactive architectural design compositions within a 3D real-time virtual environment. The features of implemented system includes providing a set of 3D objects for architectural design (walls, doors, windows, floors, ceiling, and shading devices) that are made available within the real time 3D virtual environment, allowing users to modify the geometrical and non-geometrical properties of these objects, and maintaining interrelationships between these objects using constraint-based rules automated by the developed system in real-time.

2. INTERACTIVE MODELING WITHIN 3D REAL-TIME VIRTUAL ENVIRONMENTS

In architectural design, architects are accustomed to investigating objects and spaces in their designs whereby objects play a dynamic role in building design. Current virtual environments provide users with the ability to build their own 3D structures using generic objects that are available in their objects library or imported from CAD systems to augment the existing library. In order to best utilize current virtual environments for architectural design at the conceptual stage, there is a need to embed adaptable architectural objects that can be easily inserted and modified within these virtual environments while developing an architectural design concept of buildings. Furthermore, both graphical and non graphical information about these architectural objects should be automatically updated in an accessible database for reasoning on developed conceptual designs using these architectural objects. This paper presents a system for architectural designs to be interactively composed by designers in real-time 3D virtual environments using a set of architectural objects. This set of 3D architectural objects includes walls, doors, windows, floors, and ceiling and are made available to be placed and manipulated by designers within the real time 3D virtual environments. These architectural objects are connected to each other through relationships that form assembly hierarchies between the parts of a building. For instance in FIG. 1, a room is considered an assembly of parts such as walls, doors, windows, ceiling and floor. Assembly hierarchies allow propagation of changes from objects to their dependent and related objects. For example, when a wall is relocated, then the system automatically updates all its parts (windows and doors) and relocates them accordingly as well as maintaining the relationships between this walls and connected walls; these relationships are predefined in the system. The automatic propagation of changes helps to maintain the integrity of relationships between architectural objects.

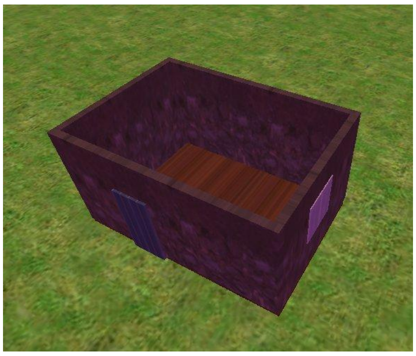


FIG. 1: An example of a design composition of a room using the assembly hierarchy architectural objects including walls, windows, doors and floor.

The proposed work within virtual environments is distinguished from existing Virtual Design Environments, such as ETH (Engeli, 2001), COVEN (COVEN), MASSIVE (Greenhalgh 1995), DIVE (Frécon, Ståhl et al. 1991) by providing an interactive 3D virtual environment with architectural objects that can be used to create architectural designs in real time within these environments without having to have an external CAD system to model such objects. This is in addition to providing assembly hierarchies among these objects as building components, and including non-graphical properties attached to each instance of these objects such as function and material. On the other hand, COVEN and MASSIVE projects are two systems not primarily concerned with real-time (synchronous) multi-user representation of information and dynamic models. They offer document sharing and multiple interpretations of designs but do not enable collaborative designing on a single shared model in real time. The DIVE system focuses on developing support of standardized tools and multi-user applications for networked participation. Collaborative Virtual Environments (CVEs) of the nature of COVEN support a wide range of disciplines, e.g. design, visualization, simulation, training, education and entertainment.

The DIVE technology provides the platform on which the MASSIVE networked virtual environment technologies operate whereby MASSIVE is an essentially a teleconferencing system.

Other relevant work on architectural design in 3D Virtual Environments include: Virtual Environment for Conceptual Design in Architecture (Anderson et al, 2003); and DesignWorld (Rosenman et al, 2006 and 2007). The Virtual Environment for Conceptual Design in Architecture in addition to providing an interactive environment for creating and manipulating geometry, it attempts to adapt and enhance elements from traditional design environments that are rich in imagery and information. The environment supports simple creation, manipulation, copying and deletion of rectangular solids and cylinders in the manner of many 3D CAD modelers. On the other hand DesignWorld is a prototype system for enabling collaboration between designers from different disciplines who may be in different physical locations. DesignWorld consists of a 3D virtual environment augmented with web-based communication tools and agents for managing the different discipline objects. It uses agent technology to maintain different views of a single design in order to support multidisciplinary collaboration. This architecture enables DesignWorld to address the issues of multiple representations of objects, versioning, ownership and relationships between objects from different disciplines.

Both Virtual Environment for Conceptual Design in Architecture and DesignWorld are useful and interesting prototypes of virtual design environments; however both are not providing a platform for architectural designers within which they can create designs from primary architectural objects and maintaining the relationships amongst these objects within a real-time and multi-user 3D virtual environment. Hitherto Virtual Environments for architecture are lacking appropriate tools for 3D design. Navigating and manipulating 3D design requires 3D geometrical primitives as well as a set of 3D design tools which are introduced by the developed IAMVE system presented in this paper. The developed system is given the acronym of "IAMVE"; that is Interactive Architectural Modeling in Virtual Environments.

3. IMPLEMENTATION OF IMAVE

The IAMVE is composed of four components as illustrated in FIG. 2: (a) Virtual Environment, (b) Web-based interface, (c) Constraint-based Rules Program, and (d) MySQL database. The Web-based interface is integrated within the primary interface of the virtual environment. The Web-based interface is the system's interface; hence it is referred to as the IMAVE interface. It provides the user with a set of 3D architectural objects that are used in creating architectural designs with the virtual environment. These objects include floor, wall, window, roof, ceiling and shading devices. The IMAVE is developed using the PHP programming language. The result of development phase of integrating the IMAVE interface within the virtual environment interface (Activeworlds) is shown in FIG. 3. The Constraint-based Rules Program detects designers' actions and records them in the MySQL database. Both Constraint-based Rules Program and MySQL run in the background while the IMAVE and the virtual environment interface run in the foreground.

Designers' actions include all placed architectural objects using the IMAVE interface, and any deletion and modification of the attributes of these architectural objects (both graphically and non-graphically). The Constraint-based Rules Program records all graphical and non graphical information of the architectural objects placed by each designer tagged with his/her ID in which each designer (user) has a designated ID. Also, the Constraint-based Rules Program has a set of predefined rules that the system maintains in the developed designs using the IMAVE interface. These rules are formulated to help the users (designers) to focus on design development while the system takes care of the modeling and geometrical relationships. These rules include (a) connecting the edges of non-parallel walls that have gaps within the distance of 50 cm, (b) maintaining an assembly hierarchy between the wall and its doors and windows, e.g. if a wall is moved, then its doors and windows are moved along with it, and (c) maintaining an assembly hierarchy between the window and its shading devices, e.g. if a window is moved or modified, then its shading devices are moved or modified accordingly and along with it. The Constraint-based Rules Program is developed using the C programming language that utilized the SDK features (software development kit) of the virtual environment (Activeworlds). It communicates with MySQL database to record and modify its contents to maintain the predefined rules. These modifications are then reflected in the virtual environment. Both Activeworlds server and IAMVE are operating on one server machine. Once IAMVE is executed at the server machine, then it interface will be appear to all users whenever they access the Activewrolds server using the Activeworlds client software from any computer through any internet access, but preferably through LAN or DSL connection.

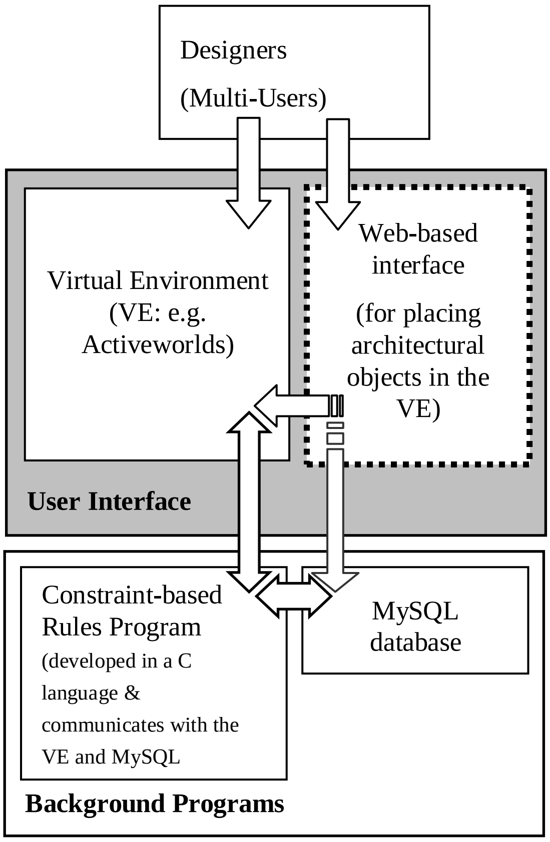


FIG. 2: Framework and components of the IMAVE system.

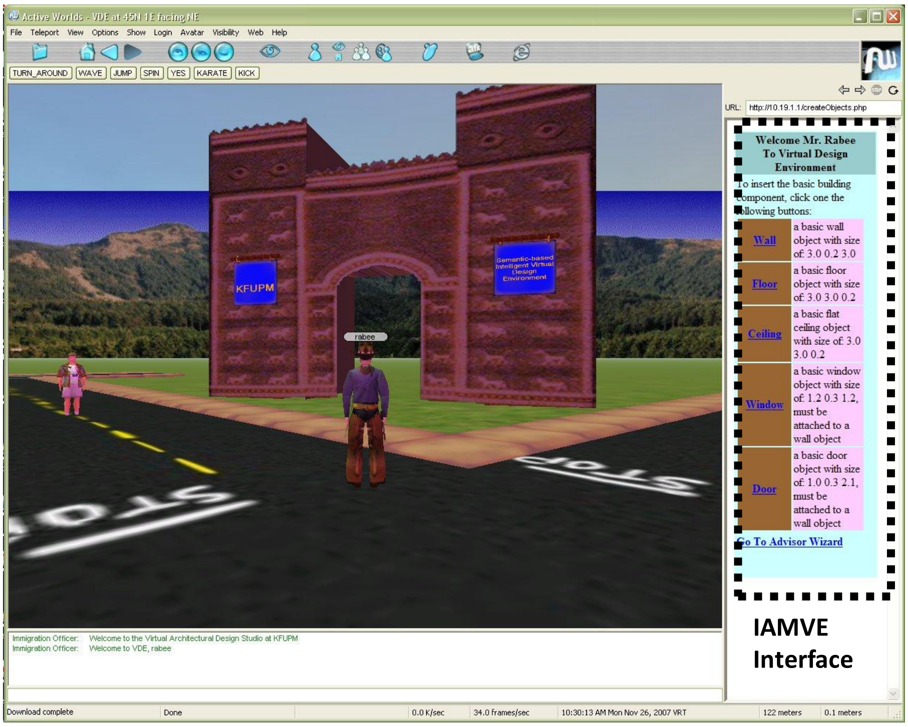


FIG. 3: The IAMVE interface integrated within the conventional interface of the virtual environment platform.

4. FEATURES OF THE DEVELOPED SYSTEM FOR INTERACTIVE ARCHITECTURAL COMPOSITIONS IN VIRTUAL ENVIRONMENTS

The IAMVE system has various features that are articulated in the following sub-sections including:

(a) Provision of placing 3D architectural objects from inside the virtual environment without having to import such objects from an external CAD system.

(b) Provision of modifying the geometrical and non-geometrical properties of each architectural object from inside the Virtual Environment.

(c) Automated assembly hierarchies between architectural objects using constraint-based rules that facilitate the propagation of changes from objects to their dependent and related objects.

4.1 Placing 3D architectural objects from inside the virtual environment

Using the IAMVE system, the user can place architectural objects at the avatar (user's figurative representation in virtual environments) location. These objects include walls, floors, ceiling, doors, windows, and shading devices. Each of these objects has default values that can be altered by the user. FIG. 4 illustrates the placement of walls, floor, ceiling, door, window and shading devices objects in the virtual environment using the IAMVE objects. The Constraint-based Rules Program detects all new architectural objects placed in the virtual environment through the use of IMAVE objects and records their geometrical and non-geometrical properties in the MySQL database as shown in the example in FIG. 5.

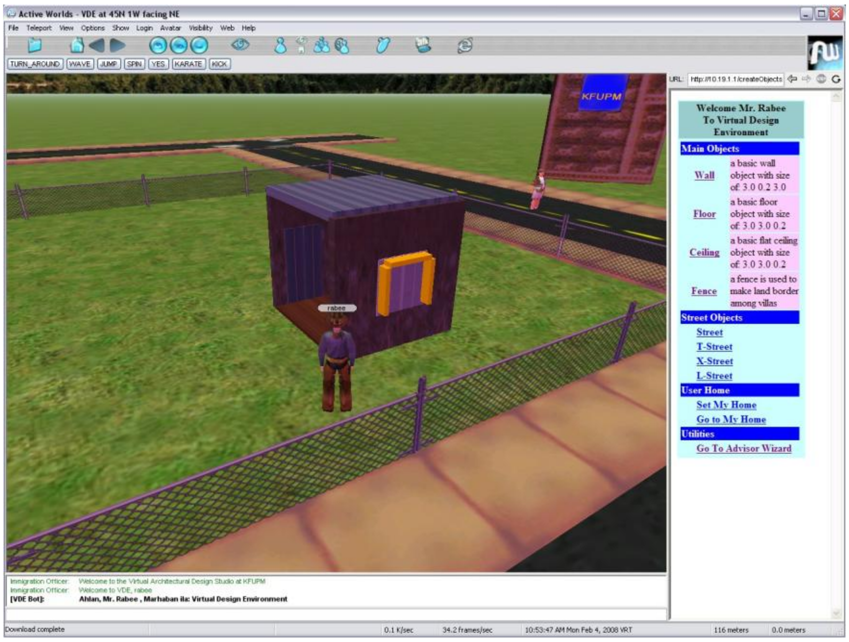


FIG. 4: An example of utilizing the IAMVE system for placing walls, floor, ceiling, door, window, and shading devices objects in the virtual environment.

Query results operations
Print view | Print view (with full texts) | Export

Show : 30 row(s) starting from record # 0
in horizontal mode and repeat headers after 100 cells

awObjectNumber	objectNumber	objectType	Functional	Materials	color	textrure
276841682	10	floor	MEZZANINE	PLYWOOD	NULL	NULL
1427077822	11	floor	ATTIC	HEBEL	NULL	NULL
2101895629	12	floor	FAMILYROOM	BRICK	NULL	NULL
653332108	14	floor	BEDROOM	PLYWOOD	NULL	NULL
-2073065342	15	floor	make a selection	make a selection	NULL	NULL
1730278567	17	wall	make a selection	make a selection	NULL	NULL
310968644	12	wall	make a selection	make a selection	NULL	NULL
-1698446897	16	floor	BEDROOM	PLYWOOD	NULL	NULL
1946133015	31	wall	BEDROOM	BRICK	NULL	NULL
1955303132	32	wall	DININGROOM	BRICK	NULL	NULL
1509764573	1	ceiling	make a selection	make a selection	NULL	NULL
2141952413	6	window	LOUNGE	make a selection	NULL	NULL
-1777750209	33	wall	make a selection	make a selection	NULL	NULL
-1630578394	33	wall	make a selection	make a selection	NULL	NULL
-1390550681	8	wall	DINING	PLYWOOD	NULL	NULL

Check All / Uncheck All *With selected:*

Show : 30 row(s) starting from record # 0
in horizontal mode and repeat headers after 100 cells

FIG. 5: An example from MySQL database for some architectural objects placed in the virtual environment using the tools in IMAVE interface.

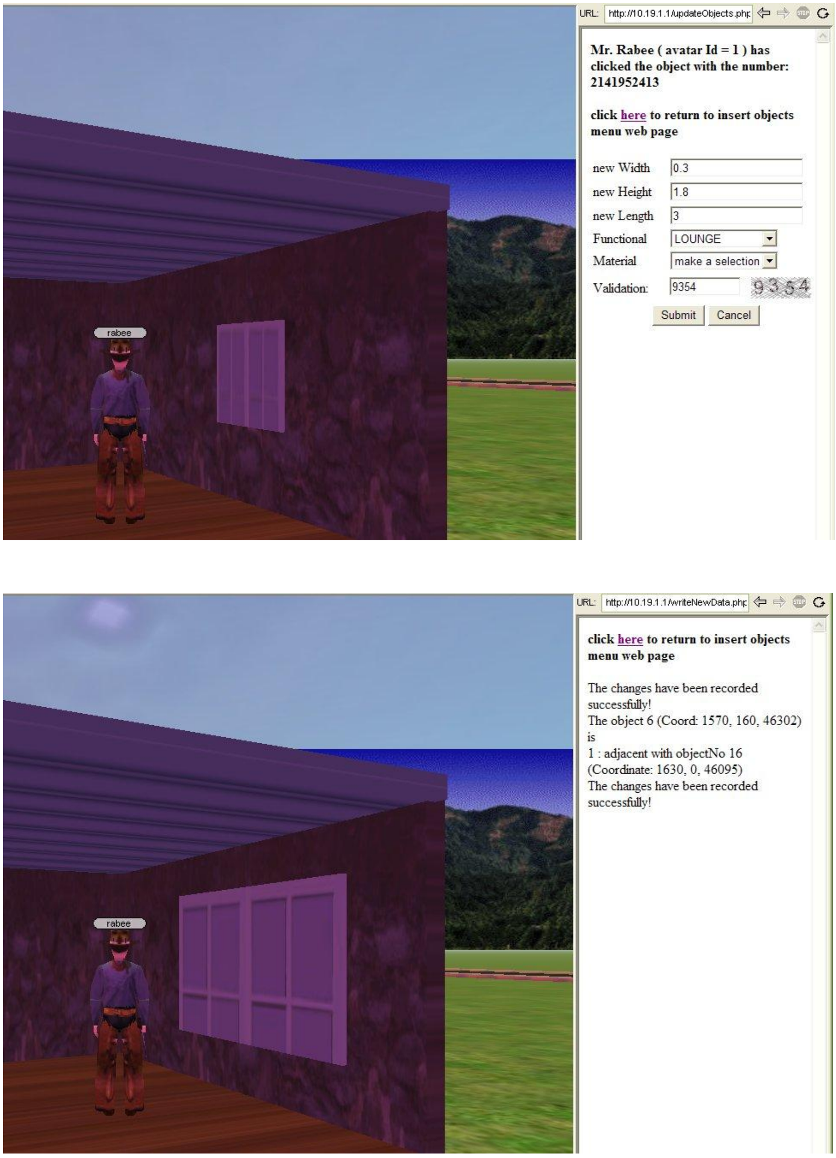
4.2 Modifying geometrical and non geometrical properties of architectural objects

Both geometrical and non-geometrical properties of all placed objects using the IAMVE objects can be easily modified by pressing a single click of the left mouse button on the selected object within the virtual environment. This action activates the IMAVE system to query the properties of the selected object from MySQL database and the Web-based interface presents object properties and their values to the user to modify some or all of these values as required by the user. Once new values for any objects are introduced, they are automatically stored in the MySQL database. The Constraint-based Rules Program regularly reads the MySQL database every five seconds and updates the graphical representations of all objects in the virtual environment according to the latest update in the database. For instance, FIG. 6 (a) shows the menu of changing the geometrical and non-geometrical properties of the selected window object with the new inserted values for width, length, functional and material properties of the selected window object. FIG. 6 (b) illustrates the automated update by the Constraint-based Rules Program for the graphical representation of the window object in the virtual environment reflecting its new values.

4.3 Maintaining the assembly hierarchy and interrelationships between architectural objects

The Constraint-based Rules Program takes care of maintaining the assembly hierarchy and predefined interrelationships between architectural objects. This helps to facilitate the propagation of changes from architectural objects to their dependent and related objects. These rules include the embodiment of doors and windows within the wall object and the automated connection between the wall objects that are not accurately placed to form appropriate corners of the building. There are three types of predefined rules in the Constraint-based Rules Program: (a) connecting the edges of non-parallel walls that are within the distance of 50 cm, (b) maintaining an assembly hierarchy between the wall and its doors and windows, and (c) maintaining an assembly hierarchy between the window and its shading devices. These rules are invoked by the Constraint-based Rules Program whenever it reads or updates the MySQL and then updates the graphical representations of all architectural objects in the virtual environment according to the application of these rules.

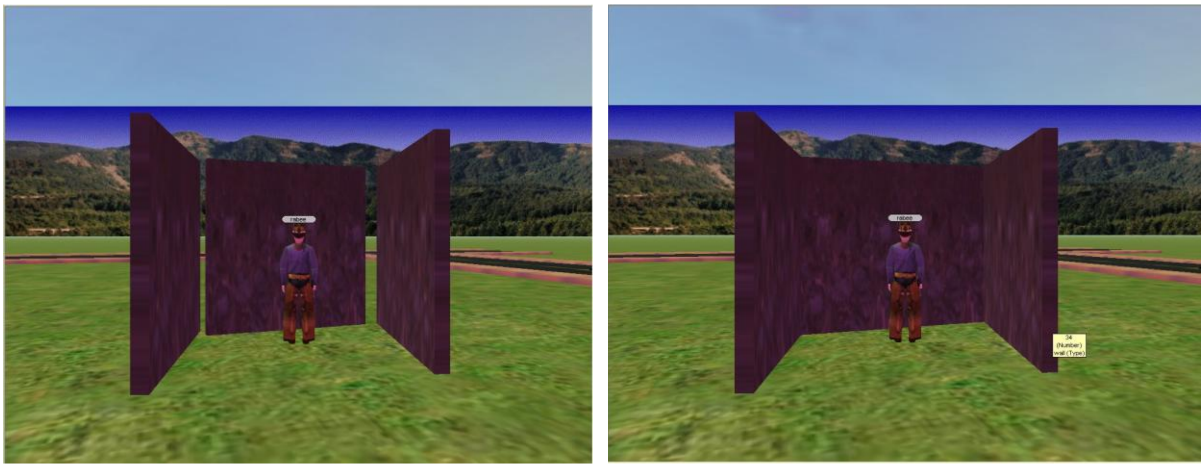
For instance, FIG. 7 (a) shows an example of a group of three walls with gaps in between. FIG. 7 (b) shows the results of using Constraint-based Rules Program for connecting the edges of non-parallel walls that have gaps within the distance of 50 cm. This automated process by the Constraint-based Rules Program helps to appropriately connect these walls to form accurate building corners. The Constraint-based Rules Program also maintains the relationships between the wall object and its interrelated components that include doors and windows. Whenever the user moves, or rotates, or deletes a certain wall, then the Constraint-based Rules Program accordingly moves, or rotates, or deletes all interrelated components of that wall. Similarly, the Constraint-based Rules Program maintains the relationships between the window object and its shading devices. Hence, whenever the user modifies the geometrical properties (height and/or width) of a window then the Constraint-based Rules Program accordingly modifies the geometrical properties (height and/or width) of the shading devises of that window. The IMAVE system provides three types of shading devices: horizontal, vertical and box. These modifications are based on a rule that assumes equal width and height for the windows and its shading devices. An example of an architectural design of a one floor house developed by the user and automatically modified by the IMAVE system is shown in FIG. 8.



(a)

(b)

FIG. 6: (a) A menu for changing the geometrical and non-geometrical properties of the selected window object; and (b) the automated update by the Constraint-based Rules Program for the graphical representation of the window object in the virtual environment reflecting its new values..



(a) (b)

FIG. 7: (a) A menu for changing the geometrical and non-geometrical properties of the selected window object; and (b) the automated update of changes for new geometrical properties of the window object within the virtual environment.

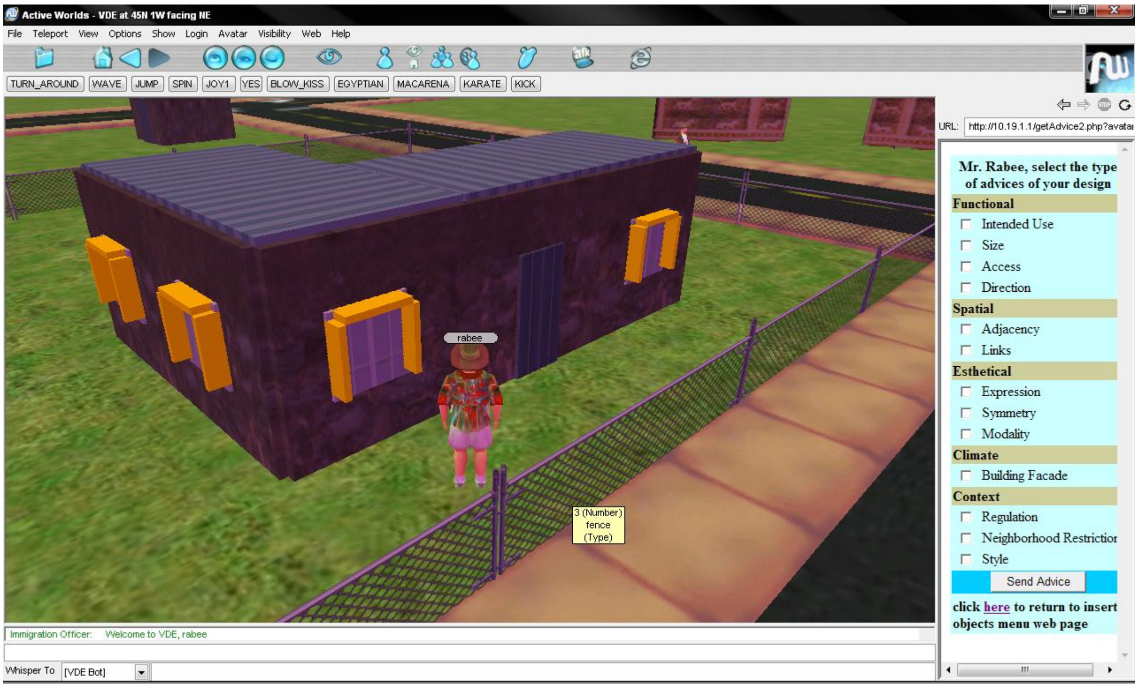


FIG. 8: An example of an architectural design of a one floor house developed by the user and automatically modified by the IMAVE system.

5. DISCUSSION

Most of the current computational tools used in architectural design at the conceptual stage are single user and hardly support collaborative designs. Developing tools that allow creating architectural designs within multi-users real time 3D virtual environments provides the bases for extending the utilization of virtual environments to be interactive and collaborative medium in architectural design education and practice. Hence, this work will potentially improve the status of current virtual environments that were initially developed for chatting and presentation to be employed for design support in architecture at the conceptual design stage. This paper presented the development of a computational system for interactive architectural modeling within virtual environments (IAMVE), e.g. Activeworlds. The features of implemented IAMVE system include providing a set of 3D objects (walls, doors, windows, floors, ceiling, and shading devices) that are made available within the virtual environment and can be utilized by the user to create architectural compositions more interactively by placing them at the avatar (user) location. The system allows users to modify these objects including both geometrical and non-geometrical properties such as function and material. All objects are identified based on the ID of the user placing them in the virtual environment using the IAMVE system. The object properties and their interrelationships are recorded in the IAMVE system's database for further analysis with instant update and direct link between the virtual environment and database of the developed system. Furthermore, constraint-based rules are used within the system to maintain certain qualities of architectural compositions such as the interrelationships between walls, doors, windows, and shading devices. Such relationships are automated by the developed system in real-time. This provides more flexibility for the users “designers” and allows them to focus on designing while the IMAVE system takes care of the modeling issues through the constraint-based rules. The above qualities provide an edge to multi-user real time 3D virtual environments to be better utilized in the context of architectural design.

The work reported in this paper presents the results of the first phase and the base of a design support system for the architectural design of houses in real-time 3D collaborative virtual environments. The features of the IMAVE system introduced above form a necessary tool for creating architectural designs within virtual environments. The forthcoming design support system in real-time 3D virtual environments aims to provide architectural design support in the form of design advices based on designers' request via observing, recognizing and conceptualizing designers’ actions in the virtual environment and advise them according the knowledge available in the system’s knowledge base. The logs and data recorded in MySQL database in the developed IMAVE system provide the essential design information for reasoning in the forthcoming design support system. This design support system is a multi-agents system that has a set of agents covering the functional, spatial, esthetical, climate, and contextual issues of designing detached houses.

6. ACKNOWLEDGEMENT

The authors would like to acknowledge and thank the support of King Fahd University of Petroleum and Minerals (KFUPM) via the Deanship of Scientific Research for funding this research through the Internal KFUPM Research Project No. INT-311.

7. REFERENCES

Anderson L., Esser J. and Interrante V. (2003). A Virtual Environment for Conceptual Design in Architecture, *Ninth Euro-graphics Workshop on Virtual Environments*, Zurich, Switzerland.

COVEN (Collaborative Virtual Environment), http://coven.lancs.ac.uk/home.htm.

Engeli M. E. (2001). *Bits and Spaces: CAAD for Physical, Virtual and Hybrid Architecture at ETH Zurich*, Birkhäuser, Basel, Switzerland.

Frécon E., Ståhl O. and Wallberg A. (1991). DIVE (Distributed Interactive Virtual Environment), http://www.sics.se/dive.

Greenhalgh C. (1995). MASSIVE: Model, Architecture and System for Spatial Interaction in Virtual Environment, a Collaborative Virtual Environment for Designing, http://www.crg.cs.nott.ac.uk/research/systems/MASSIVE/.

Kurmann D., Elte N. and Engeli M. (1997). Real-time modeling with architectural space, in R. Junge (ed.), *CAAD Futures 1997*, Kluwer Academic, Dordrecht, The Netherlands, pp. 809-819.

Maher M. L., Simoff S. and Cicognani A. (1999). *Understanding Virtual Design Studios*, Springer, London.

Maher M.L., Liew P-S, Gu N. and Ding L. (2003). An Agent Approach to Supporting Collaborative Design in 3D Virtual Worlds, in Dokonal, W. and Hirschberg, U. eds, eCAADe21 Digital Design, *Proceeding of eCAADe 2003*, Graz University of Technology, Austria, pp. 47-52.

Rosenman M., Merrick K., Maher M. L. and Marchant D. (2006). DesignWorld: A Multidisciplinary, Collaborative Design Environment Using Agents in a Virtual Worlds, *Proceedings of the Second International Conference on Design Computing and Cognition*, Eindhoven, Netherlands.

Rosenman M., Smith G., Maher M. L., Ding L. and Marchant D. (2007). Multidisciplinary collaborative design in virtual environments, *Automation in Construction*, 16, pp. 37 – 44.

Schmitt G. (1997). Design Medium-Design Object, Keynote, in R. Junge (ed.), *CAAD Futures 1997*, Kluwer Academic, Dordrecht, The Netherlands, pp. 3-13.

Schnabel M. and Kvan T. (2001). Design Communication in Immersive Virtual Environments: An Initial Exploration, Architectural Information Management, *Proceedings of 19th eCAADe Conference*, pp 394-400.

Scola N. (2006). Avatar Politics: The Social Applications of Second Life, Report for the Institute for Politics, Democracy, and the Internet, Available from: www.ipdi.org.

Sherman W. and Craig A. (2003). *Understanding Virtual Reality: Interface, Application and Design*, Morgan Kaufmann Publishers, San Francisco, CA.

Strehlke K. and Engeli M. (2001). [roomz]&[connectionz] - scenarios in space and time, in B. de Vries, J. Leeuwen and H. Achten (eds), *CAAD Futures 2001*, Kluwer Academic, Dordrecht, The Netherlands, pp. 173-186.

Turk Z. (1991). Integration of Existing Programs Using Frames, *CIB Seminar Computer Integrated Future*, 16-17 September, Eindhoven, Netherlands.