VISUALISATION AND OBJECT DESIGN IN VIRTUAL ARCHITECTURE

MARY LOU MAHER, NING GU, FEI LI
Key Centre of Design Computing and Cognition
Faculty of Architecture
University of Sydney

Abstract. The design of virtual architecture is currently ill-defined and lacks a framework for understanding existing designs. We present a basis for the representation of virtual architecture that follows from the idea of conceptual metaphor. This approach addresses the limitations of current environments for designing virtual architecture by providing a basis for combining visualisation and object design.

1. Introduction

Virtual architecture has different meanings in different contexts for example, as information architecture (Schmitt, 1999) or virtual place (Novak, 1990). For the purposes of this paper, we distinguish virtual architecture from digital architecture and (physical)[1] architecture.

- **Digital architecture** is the use digital representations in the development of architectural designs.
- **Physical architecture** is the result of architectural design as a physical building.
- **Virtual architecture** is the result of architectural design that serves its purpose as a digital representation.

Virtual architecture provides both a sense of place and a sense of presence, as is assumed in physical architecture. In physical architecture, the place is defined by its boundaries and contents, and sense of presence gives us an awareness of others. In virtual architecture a sense of place can be achieved by visualising the boundaries and contents of functional places that share some of the functions of physical architecture. A sense

[1] The word “physical” is in brackets because most people assume that architecture refers to physical architecture. The distinction is necessary only when virtual architecture is possible.

of presence in virtual architecture can be achieved through the explicit representation of and communication among ourselves and others in the virtual place.

Existing virtual worlds provide the implementation platforms for designing and building virtual architecture. Representative examples of these platforms are:

- LambdaMOO: LambdaMOO (Curtis, 1993; Rowley, 1997) is a multi-user text-based virtual.
- Active Worlds: Active Worlds[2] is a 3D world that supports collaboration and building.
- VWorlds: VWorlds[3] is a 3D object-oriented virtual world that is based on the object hierarchy of LambdaMOO.

Limitations in the current design and implementation of virtual architecture are:

- There is a lack of consistency in the underlying representation and implementation of interactive 3D models and the representational needs of virtual worlds that can support a broad range of human activities[4].
- Virtual architecture is currently understood in terms of its visualisation (for example, Anders, 1999) and does not adequately carry through to functional virtual places.
- Existing 3D virtual environments for collaboration support the design of the visualisation of a virtual world but are limited to a predefined set of behaviours (elaborated below).

This paper elaborates on the design and representation of existing object-oriented virtual architecture and presents a model for object design and visualisation with the three components: conceptual basis, semantic frame and visualisation shell.

2. Object Design in Virtual Worlds

The best current examples of virtual architecture are virtual worlds. Generally, virtual worlds adopt a spatial metaphor and can be classified into two categories: non object-oriented systems like The Palace[5] and object-oriented systems like MOO based systems (Curtis, 1993; Rowley, 1997). An object-oriented system has many advantages over a non

[2] www.activeworlds.com

[3] www.vworlds.org

[4] The computer games that use 3D models for human interaction provide examples of possible representations for virtual architecture, however, they only support a narrow range of activities that centre around killing the enemy.

[5] http://www.palacetools.com/home.php3

object-oriented system. In implementation terms, the object-oriented system has a more robust and dynamic software core, which can be consistently modified and expanded. The merits of the object-oriented system in terms of design and representation are:

- When representing virtual architecture, we can represent the design as classes of objects in accordance with the object-oriented characteristics of the system. Thus the representation of virtual architecture, which is based on the definition of objects and their relationship, can also be conceptually depicted as an object-based representation framework.
- Design in the sense of a sequential refinement process can be better handled with this object-based representation. The representation of a design model as class objects not only passes the common identifying design properties of the class to the descendants, but also provides a framework from which the refinement process starts. As such, the design itself can be more efficient and manipulable.
- Conceptual objects described by design can correspond to the objects in the object-oriented database. This makes possible the use, as an analogy, of the research in design of physical architecture that use the prototype formalism (Gero, 1990). For example, we can consider the function, behavior, and structure of objects in each case.

Here we look at three types of object-oriented virtual architecture in terms of their object design and visualisation. These three provide a list of precedents when considering the object design of new virtual architecture.

2.1. THE VIRTUAL CAMPUS IN LAMBDAMOO

The Virtual Campus (VC) in the Faculty of Architecture at the University of Sydney (http://www.arch.usyd.edu.au:7778; Maher, 1999) is an example of virtual architecture that combines the object design in LambdaMOO with visualisation. In a virtual world constructed in LambdaMOO, each entity or programmed behaviour is represented as an object. LambdaMOO (Curtis, 1993; Rowley, 1997) is a multi-user text-based virtual world that is implemented in a persistent object-oriented database where the client is a telnet window that connects to a database server. The virtual world comprises rooms, people, things, and exits. Recent developments in the use of the LambdaMOO database has lead to WWW interfaces that provide a visualisation of the rooms and things in the virtual world. The visualisation of the world can be achieved by attaching an image or a VRML file as an attribute of the object and embedding this file in a web window. Due to the limitations in the interaction with images and VRML on the WWW, interaction with the visualisation of the place in LambdaMOO is limited. Most of the interaction with the objects in a LambdaMOO virtual world occurs through a text-based command language similar to the command shell in Unix.

The fundamental class structure of LamdaMOO has a root class with the basic properties of all objects. The next level of classes, generic room, generic thing, generic player, and generic exit, define the major types of objects in the world. The generic room is the basic representation of place. The generic thing is a representation of the objects that can be placed in a room. The generic player is an object representation of the people in the world. The generic exit is the basis for navigating and taking things and people from one room to another. Through the design of virtual architecture using these basic objects as building blocks, "geographical locations" and ‘buildings" can be composed.

The Virtual Campus uses the MOO object structure to create an inheritance hierarchy of rooms and things in a room. The organisation of the Virtual Campus into buildings follows a functional decomposition of places.

The visualisation of rooms in the VC is presented in a web browser. Figure 1 shows the conference room that is the basis for seminar presentations. This room is typical of the classrooms and is an example of the integration of the visualisation of the room and its functions. The tool bar marked as "1" in Figure 1 allows the student to switch his attention from the room view, the slide projector screen, the whiteboard, or the course materials.

There are several ways of defining and visualising the function of the room. The various functions are programmed into the room object as methods and there are things in the room that have a specific purpose. Clicking on icons or typing commands give access to the things in the room, for example the projector and recorder. The communication functions of the room, for example asking who is in the room and talking to someone in the room, are provided in a chat-like "talk by typing".

Additional functionality is usually added incrementally by inserting another object in the class hiearchy. For example, the class structure of the meeting room prototype is:

Generic Room(#3), Generic Improved Room(#184), Generic Improved Room with Cleaning and Scripts(#206), Meeting Room Prototype(#211)

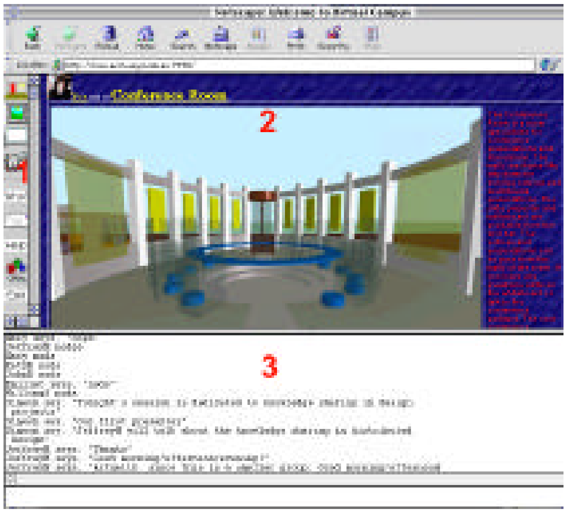


Figure 1. Seminar room in the Virtual Campus

A major limitation in a MOO virtual world is the incremental development of the objects without a consideration of the entire world as a collection of conceptual entities. Essentially, MOO worlds are built and revised without consideration for the design of the objects as part of a coherent virtual world.

Another major limitation is the link between the object definition and the visualisation, where each is a separate computational entity. A person can interact with an object through a command language and see the visualisation on a web browser window. There is limited interactivity in the web browser window that supports moving from one room to another and clicking on tools that have one function. A closer relationship between the object design and the visualisation would allow for multiple functionality to be available from the visualised world.

- ACTIVE WORLDS

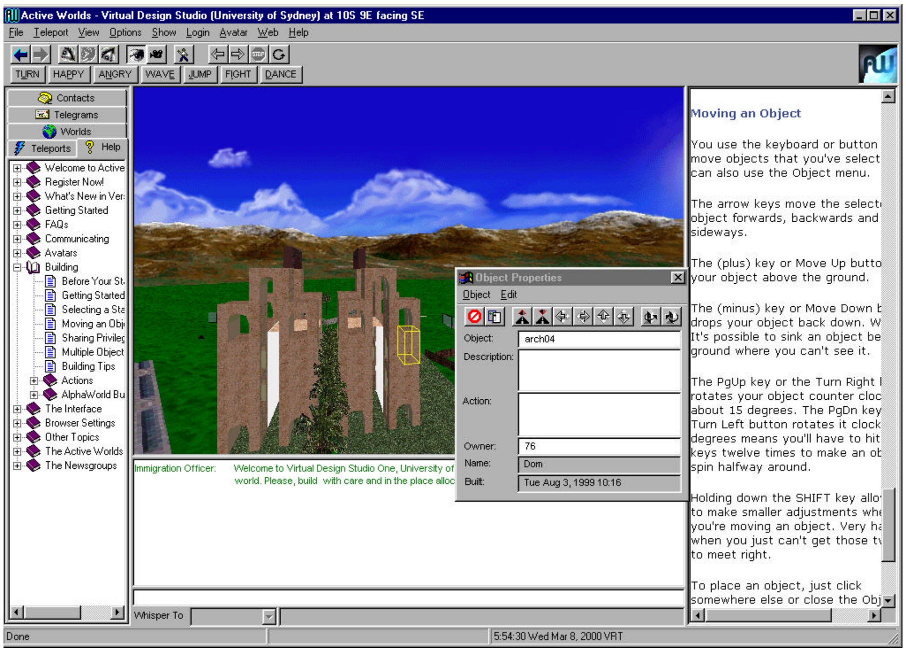
Active Worlds[6] is a 3D object-oriented virtual world accessed through a specialised client connected to an Active Worlds database server. The virtual world comprises a site, the building objects that create a sense of place, and the avatars that represent the people that are citizens or tourists in the world.

A world is a virtual geographical territory of a specified size measured by kilometers. This "land" is surrounded by a panoramic skyline picture, which gives the world a scene. The system provides users with modeling tools and object building blocks. Users can clone these building blocks to

[6] www.activeworlds.com

construct their own buildings. In addition to building models and land scenes, there are also avatars and events. Events are programs attached to a model in the world that provides interactions. For example, the most used events are those that change the position or orientation of an avatar or a model. In Active Worlds, despite the richness of a 3D enhanced place, the function of the place is restricted to talking and building.

Building objects are the basis for creating virtual architecture. Each object has an associated 3D model stored in a separate file on the server. The building objects have a predefined set of attributes that allow for a fixed set of behaviours such as "open a web page". One attribute takes on the value of a filename of the associated 3D model of the object. Figure 2 shows the Active Worlds client with the object editor window open over the 3D world window. A major limitation in Active Worlds is that the objects cannot be extended to include other attributes or behaviours.



Immigration Officer: Welcome to Virtual Design Studio One, University of world. Please, build with care and in the place alloc

Object: arch04
Description:
Action:
Owner: 76
Name: Dom
Built: Tue Aug 3, 1999 10:16

5:54:30 Wed Mar 8, 2000 VRT

Moving an Object

You use the keyboard or button move objects that you've select can also use the Object menu.

The arrow keys move the select object forwards, backwards and sideways.

The (plus) key or Move Up butto your object above the ground.

The (minus) key or Move Down b drops your object back down. W It's possible to sink an object be ground where you can't see it.

The PgUp key or the Turn Right rotates your object counter cloc about 15 degrees. The PgDn key Turn Left button rotates it clock degrees means you'll have to hit keys twelve times to make an ob spin halfway around.

Holding down the SHIFT key allo to make smaller adjustments whe you're moving an object. Very ha when you just can't get those tv to meet right.

To place an object, just click somewhere else or close the Obj

Figure 2. Active Worlds object editor

From the representation point of view, we see that there is no place object existing in Active Worlds. A building is just the stacking of building blocks. Building blocks are objects with properties that can be modified. However, the buildings themselves are not objects. A building does not have a separate identification in the world. It doesn't have any properties and functional attachment, hence it does not have a dedicated representation.

VIRTUAL WORLDS

VWorlds[7] is a 3D object-oriented virtual world that combines the advantages of Active Worlds, a 3D interactive environment, and LambdaMOO, object classes include a room object and a portal object in addition to the building objects and avatars. The explicit representation of the room and portal provides the basis for creating a more comprehensive virtual world in which people can navigate and create specialised environments for different activities. VWorlds is not treated in detail here because it has many of the same advantages and disadvantages of Active Worlds in designing virtual architecture.

3 Combining Object Design and Visualisation

The examples of virtual architecture above have the benefits of an object-oriented representation as their implementation platform, but lack a consistent and extendable use of objects that combine the object representation with the visualisation of the design. VWorlds provides a good basis for the implementation of virtual architecture, but still maintains a separation of visualisation and object design. Where LambdaMOO provides a language that can support object design, Active Worlds and Virtual Worlds have focussed on the support for designing the visualisation of the virtual world.

We propose a model for the design of virtual architecture that builds on object design as the basis for designing. The model is based on developments in representing the design of physical objects, such as the FBS model (Gero, 1990), and conceptual metaphor as a cognitive structure (Lakoff and Johnson, 1999). In this model we consider the difference between designing physical worlds (PW) and virtual worlds (VW) as metaphorical worlds.

The FBS model characterises design objects and is the basis for a representation of designs. F (function) is related to the purpose of the design. It is not directly related to any substantial design structural component. The function of place design in PW and VW are the same in the sense that in both cases the place is intended for similar purposes. B (behavior) reflects the performance of the design artifact or the design components. It is closely related to design structure. Behavior includes

[7] www.vworlds.org

expected behavior (Be) and actual behavior (Ba). When Ba equals Be, we infer that the design satisfies the intended design function. In the designs in the physical world, it is generally known how B is derived from F and how B is linked to S. These relationships are defined by the long formed design convention and protocol. This is not the case in design in VW. A meeting room in the VW is not a room in a building even if we call it a “room". In metaphorical design, we name the design behaviors after those in physical place design. However, they aren’t derived from either the function or the metaphorical structure of the object. Behaviours in a VW are defined by the code that implements the VW. S (structure) is the basic condition of existence, and it is the carrier of the design behaviors. In a VW, verbs and properties in the object permit the existence of virtual entities. What makes the design artifact a room and a room for meeting is totally different for the PW and VW. One of the differences of the (F, B, S) model for VW representation is in the identification of the design structure. In the physical world design representation, for example in the case of a wall design, the design structure is the wall itself, and the design structural elements are the components that make up the wall. The structure of the wall is the mechanism that produces the behaviours and together they are responsible for the fulfillment of the design function. However, in VW design, the metaphorical structure is not the mechanism that produces the behaviour.

"Direct" design (in contrast to metaphorical design) is designing the design artifact as what it is; metaphorical design is to design something as if it is something else. The design artifacts of metaphorical design have two parts:

- The design artifact in the form of what it is in the design environment. In metaphorical design, if the representation addresses the design artifact as only what it is, the design F, B, S can be too unfamiliar and too abstract to grasp. For example, to treat the design object as what it is in the computer, our understanding of the object and its performance may not be much higher than the "bit" level. If we use the metaphoric structure to help us understand the design, we can design behaviours and functions consistent with the metaphor.
- A shell outside the design artifact that makes it seem to be something else. The shell is something added and unique to metaphorical design, which is not part of the (F, B, S) model.

Figure 3 illustrates the object representation using the FBS model in the physical world from an object representation in the virtual world with a shell. We use (F, B, S) to refer to a physical design object and (f, b, s) to refer to its metaphorical equivalent in a virtual world. In the metaphorical design of the virtual object, we take F as the design function of the virtual object. However, F only bestows f with meanings that are relevant to the physical world. F cannot replace f. B is introduced partly into the virtual design to name b, yet B and b are actually different because b is programmed and B is a physical phenomenon. s is only a metaphorical reference to S.

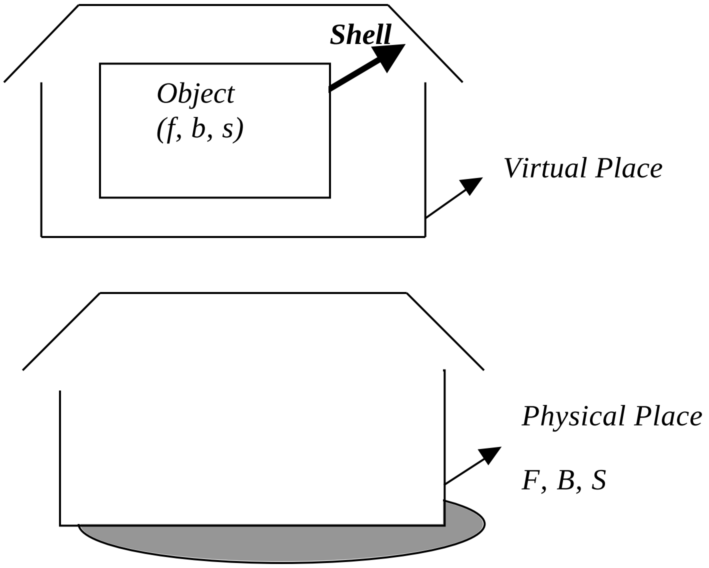


Figure 3. The use of a shell to communication the virtual place design

Lakoff and Johnson (Lakoff and Johnson, 1980, 1999) present four structures of the cognitive unconscious that provide the basis for understanding how metaphor influences our ability to make sense of subjective experience:

- Basic level concepts
- Semantic frames
- Spatial-relation concepts
- Conceptual metaphor

According to Lakoff and Johnson, the cognitive unconscious is all unconscious mental operations concerned with conceptual systems, meaning, inference, and language. Appealing to the cognitive unconscious in the design of virtual worlds allows us to conceive of and develop a virtual world that can be used by people with a more "natural" response. Since we are born and learn to act in a physical world, much of our unconscious thinking is based on our learned responses to the physical world. Lakoff and Johnson have argued that much of our thinking is also based on conjunctions of physical experiences with subjective experiences. In designing and understanding virtual worlds, creating a place that is consistent with our understanding of the physical world will allow us to consistently apply the primary metaphors we used.

Basic level concepts are a result of our innate ability to categorize the world. We categorise all the time to distinguish the things in the world in order to survive, but also in order to comprehend the world around us. Establishing the basic level of categories as the model for designing a virtual world allows users to use their intuition in interacting with the components of the world. A semantic frame defines relationships among whole fields of related concepts and words that express them. Using a consistent metaphor allowing a person to draw on their semantic frames can be the basis for designing in a virtual world. For example, when designing a virtual classroom, appealing to the semantic frame, a person would be able to develop a relationship between the classroom, a lecture, a blackboard, a desk, etc. The use of these words as the design extends the metaphor to draw on the physical classroom to provide more functionality in the virtual world. Spatial-relation concepts allow the designer to define consistent actions on the virtual object as the person would expect to do with the physical object. A person would put things "on" the desk, go "out" of a room, and write "on" the blackboard. The world is a metaphor, the programmed virtual world does not exist spatially. The use of these words provides a consistent experience in the virtual world when compared to the physical world. Conceptual metaphor allows us to conceptualize the virtual world in terms of time and motion in the physical world of architecture.

Based on the analysis of virtual design in VW using the (F, B, S) model and Lakoff and Johnson’s theory of metaphor concerning the four cognitive structures affecting the human’s understanding of experience, we present a representation model for VP design in the VW. It consists of the following three elements:

- conceptual basis,
- semantic frame, and
- visualisation shell.

Conceptual Basis (CB) is developed from the "basic level concept" of Lakoff and Johnson’s theory about metaphor. It is the part of VW design corresponding to people’s ability to categorize. In metaphorical design, CB defines the basic concepts and conditions of the existence of a design object. For example, in designing a VW, the CB assigns "Lend Lease room" to the room representation category; "Wilkinson Building" to the building category; "Slide projector" to the tool category. In metaphorical design, the CB not only clarifies the basic concepts for design, but also provides knowledge about the common characteristics and properties of this concept and the knowledge about common actions and the possible methods of interaction of the users with this concept. In the VW, CB is the object in the system’s object-oriented software environment. The object is the basic condition of existence. In the VW, the object is the basis for the use of the world. In a programming sense the object is made of groups of properties and verbs. A designer's task is not to study the verbs and properties or how they make the existence possible. These are the technological issues related to the software core of the VW system. A designer whould characterize and classify the design concepts using a building or room metaphor. For example, a room is a "container". A user can "go" in and out of it. It belongs to a building. It contains the characters of "exclusiveness" and "security", etc. In the design of VP, CB is in the set of class objects in a VW. A class object carries mechanisms that are responsible for the object’s existence and use.

Semantic Frame (SF) corresponds to the "semantic frames" in Lakoff and Johnson’s theory. A person understands the virtual environment he/she is experiencing through what can be done in the environment. And this is manifested as mechanisms that represent the technological possibilities of the virtual environment. In virtual world design, these mechanisms can either be in the design object that stands for the place, or they can be in the objects in the place, for example "recorder", "projector". In a VW, there are many mechanisms responsible for the "actions" in the design object. These mechanisms serve roughly the following general purposes: communication (saying, whispering, paging, mailing other users); activities (such as recording a conversation, showing slides on the projector); information Access. (links); and navigation (moving from one place to another).

Visualisation Shell (VS) is a shell that provides the visualisation of the concepts and semantic frame. For the place design in VW, VS creates spatiality for the design object. In metaphorical design, the critical aspects of the VS are: names and a naming system for objects, properties and verbs, and 2D and 3D visual representation.

Summary

Where most designs of virtual architecture focus on the visualisation of the design, we present a model for the object design of virtual architecture. The model includes conceptual basis, semantic frame, and visualisation as the three key components of objects in virtual architecture. These components draw upon conceptual metaphor and the FBS model of physical design. Our model does not imply that the visualisation is not central to virtual architecture, only that it does not directly take into consideration the functionality and use of the design.

Acknowledgements

This research has been supported by an Australian Postgraduate Research Award, the Australian Research Council, and the University of Sydney Small Grants Scheme.

References

Anders, P.: 1999, *Envisioning Cyberspace*, McGraw-Hill, New York.

Curtis, P. et al.: 1993, MUDs *Grow Up: Social Virtual Reality in the Real World*, Technical Report, Xerox PARC, Palo Alto, Calif., 1993; ftp://parcftp.xerox.com/pub/MOO/papers/MUDsGrowUp.ps.

Honda, Y., Matsuda, K., Rekimoto, J. and Lea, R.: 1995, Virtual society: extending the WWW to support a multi-user interactive shared 3D environment, Proceedings of VRML'95, San Diego, CA.

Lakoff, G. and Johnson, M.: 1980, *Metaphors We Live By,* Chicago: University of Chicago Press

Lakoff, G. and Johnson, M.: 1999, *Philosophy in the Flesh: The Embodied Mind and its Challenge to Western Thought*, Chicago: University of Chicago Press

Li, F. and Maher, M.L.: 2000, Representing virtual places - A design model for metaphorical design, in M. Clayton (ed) *ACADIA2000*.

Maher, M. L.: 1999) *Design the Virtual Campus as a Virtual World*, Computer-Supported Collaborative Learning, unext.com, Stanford University, pp. 376-382.

Maher, M.L., Skow, B., and Cicognani, A.: 1999, Designing the virtual campus, *Design Studies* **20**: 319-342.

Maher, M.L.and Simoff, S.: 2000, Collaboratively designing within the design, in S. Scrivener, L. Ball, and A. Woodcock (eds) *Collaborative Design*, Springer, London, pp. 391-399.

Novak, M.: 1990, Liquid architectures in Cyberspace. *in* Michael Benedikt (ed.), *Cyberspace: First Steps*, MIT Press, Cambridge, MA, pp 225-254.

Rowley, M.: 1997, Distributed MOO-Based Shared Worlds, in *Proceedings of the Sixth Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises*, IEEE Computer Society, pp 155-166.

Schlager, M. and Schank, P.:1997, TAPPED IN: A new on-line teacher community concept for the next generation of internet technology, in *CSCL'97*, The Second International Conference on Computer Support for Collaborative Learning.

Schmitt, G.: 1999, *Information Architecture*, Birkhauser, Basel.