Agent Models of 3D Virtual Worlds

Mary Lou Maher
John S Gero

Key Centre of Design Computing and Cognition, University of Sydney

Abstract

Architectural design has relevance to the design of virtual worlds that create a sense of place through the metaphor of buildings, rooms, and inhabitable spaces. The design and implementation of virtual worlds has focused on the design of 3D form for fast rendering to allow real time exploration of the world. Using platforms that were originally designed for computer games, some virtual worlds now contain preprogrammed interactive behaviors. We present an agent model of virtual worlds in which the objects in the world have agency, that is, the objects can sense their environment, reason about their goals, and make changes to the environment. This agent model is presented and illustrated using a wall agent. Following from the wall agent, we generalize how agency can be attached to any 3D model in a virtual world.

1 Introduction

Designing virtual worlds as architecture considers the form and function of spatial virtual environments as an alternative kind of architectural design. A virtual world is a composition of architectural metaphors and computing entities. The architectural metaphors are useful for providing a sense of place and, if multi-user, a sense of awareness of others. As an assembly of computing entities, virtual worlds can have programmable functions to support various online activities.

Where the use of digital media and 3D models has provided a way of visualizing, simulating and documenting architectural designs for the physical world, these media types are now used to design and create virtual worlds whose functions are available without a translation to physical structures. Designs and design representation issues of virtual worlds are concerned more with the digital representation in its own right rather than the use of digital media as design documentation. The designs need to take account of the virtual presence of the people in the world and need to go beyond the representation of form and geometry to include object behaviors. Examples of designs and design issues are discussed in (Li and Maher 2000; Maher, Gu, and Li 2001; Gu and Maher 2001; Maher and Gu 2001; Maher, Simoff, Gu, and Lau 2001a and 2001b.

Current 3D virtual worlds are largely static. The world's creator can make changes to the world but only in special cases can the users change the world. Such a restriction makes these 3D virtual worlds useful for modeling existing designs but less useful as tools for designing since designing involves change. This need to be able to have users (here the designer) modify their worlds provides the motivation for this work. There are two levels at which modifications can occur: the user can directly change the world through their direct actions, and the world can modify itself as a consequence of the user's actions.

We are developing an agent model of 3D virtual worlds that assumes a persistent object-oriented representation of the world. We go beyond the 3D model representation to give each object agency. An agent is a system that operates independently and rationally, seeking to achieve its goals by interacting with its environment. It has goals and beliefs, and executes actions based on those goals and beliefs

(Russell and Norvig 1995). We are developing a cognitively-based agent model (Gero and Fujii 2000) for virtual worlds using Sensors and Effectors as the interface between the agent and the environment, and identifying the main computational processes as sensation, perception, conception, hypothesizer, and action activation.

In this paper we present the rationale for such virtual worlds and highlight some existing approaches to design and implementation platforms. We present an agent model for 3D virtual worlds and how that model can be realized in a networked, multi-user, 3D virtual world environment.

2 3D Virtual Worlds

Developing interactive 3D models as virtual worlds is a major focus for most of the virtual world design platforms. This focus leads to a strong emphasis on the visual aspect of the virtual world. In these visualization-focused virtual worlds, interactions are attached to 3D models to support predefined actions such as animations and teleporting. Typically a world will also support avatar movements and interpersonal communication by talking. More and more platforms are becoming available for the development of 3D virtual worlds. In this section we briefly review a few that have the essential features of: 3D models, some interactivity, and avatars to represent the location of a person in the world.

The Blaxxun virtual worlds platform is developed by Blaxxun Interactive[1], which has been applied to develop many professional virtual worlds like Cybertown, IBM Canada and Munich airport center, Figure 1. The 3D model provides a visualization of the world with real time rendering. Functions are restricted to allowing avatars to move around and providing various types of digital communication such as chat, messages, and bulletin boards. This platform does not use the 3D model as the focus of the interactivity, but rather as the visualization of the world.

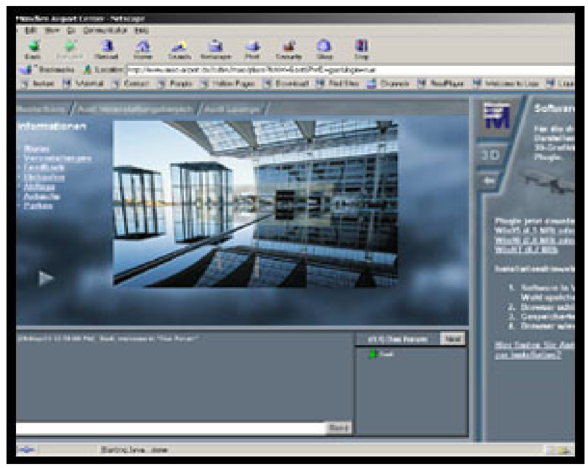

Figure 1. Munich Airport Center Using Blaxxun

LambdaMOO[2] is a multi-user text-based approach for designing virtual worlds. The world is made up of programmable objects that represent the world as rooms, exits, things, and players. The visualization of the objects in the world can be provided through image files or VRML models on a web page. Objects have been programmed in LambdaMOO to support educational and research activities, for example, projectors, whiteboards, recorders, conversational robots and so on. The 3D models provide a visual reference only. Commands need to be typed in manually to activate the rest of the functions. A VRML visualization of a LambdaMOO room is shown in Figure 2.

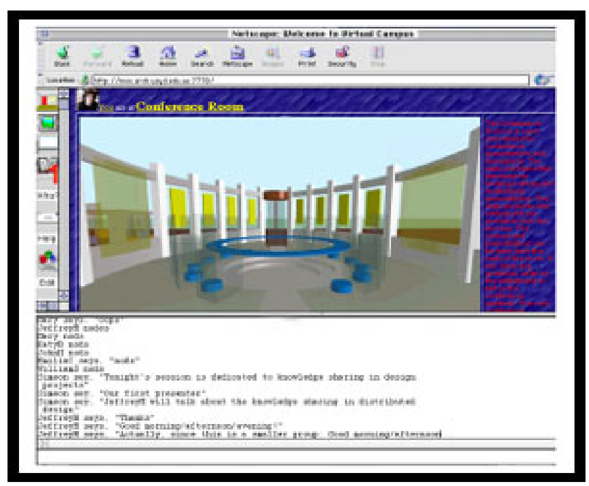

Figure 2. A virtual seminar room in LambdaMOO using VRML as the 3D interface

Active Worlds[3] is a 3D collaborative environment that allows multiple people to interact through their avatars. Active Worlds allows its citizens (users) not only to navigate the virtual world but also to design, implement and extend the environment. Objects in the world can have preprogrammed event driven behaviors such as opening a web page when clicked or teleporting the avatar when it bumps into an object. The interactivity can be extended using the software development kit. A seminar room built using Active Worlds is shown in Figure 3.

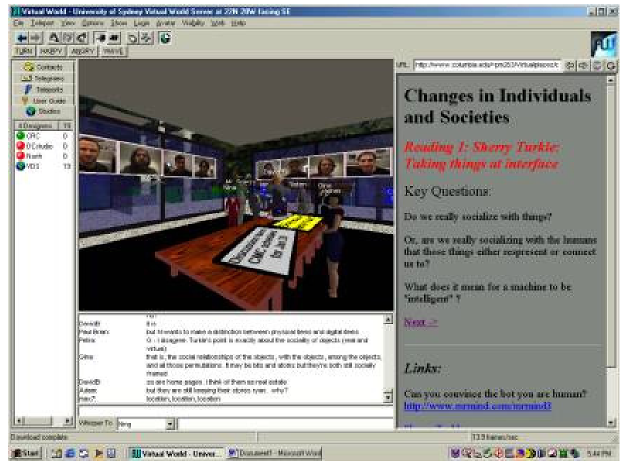

Figure 3. A seminar room in Active Worlds

The Virtual Worlds design platform, developed by Microsoft Research[4], provides 3D environments from a persistent and distributed objected-oriented database similar in structure to LambdaMOO. People interact with each other through their avatars in a 3D world. In this platform, each object can be programmed and there are preprogrammed basic behaviors such as opening web pages. A place for meetings is shown in Figure 4.

Figure 4. A place designed in Virtual Worlds

Virtools Behavior Company[5] develops Virtools for designing interactive internet gaming environments. Each object has a 3D model with associated behaviors. Behaviors specify what the object can do in terms of movement and interaction with other objects. The world developer is given a visual programming interface to implement simple and complex behaviors using a basic behavior library. A room implemented in VirTools is shown in Figure 5. Most applications using Virtools are single user and are not networked.

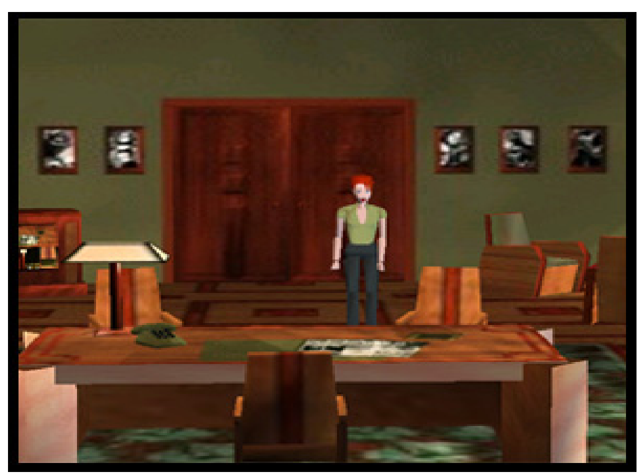

Figure 5. A virtual room built using the VirTools platform

From the review above we can see that there are several platforms for building virtual worlds that provide support for 3D modeling and rendering, with the capability of preprogramming the behaviors of the 3D objects. In the platforms described above, the behaviors are programmed either through a scripting language or using the programming language supported by the platform and its SDK. These behaviors tend to be predefined actions that are initiated by the user through the input devices of the keyboard and mouse.

In our approach, we use agent models so that the objects in the world can respond more generally to their use. In current virtual worlds a behavior is initiated by an event that a user performs on a specific object and the action is predetermined by the type of event. In an agent-based virtual world, a behavior may be triggered by any change in the data about the world and the action is determined through the agent's ability to reason about itself in the world. For example, the first kind of interaction is produced when the user clicks on an object while the second kind may be produced when an additional "person" enters a room and the room senses that there are more people than previously and reconfigures itself appropriately.

3 Agent-Based Approach for Designing Virtual Worlds

3.1 Agents and Multiple Agents

We propose a way to extend the concept of virtual worlds from preprogrammed interactive 3D models to places with objects that respond to their use by reasoning about the environment and then modifying the environment. Each object in the world is an agent element so that the world is a society of agents. Each agent element can sense and respond to the current state of the world. This is illustrated in Figure 6.

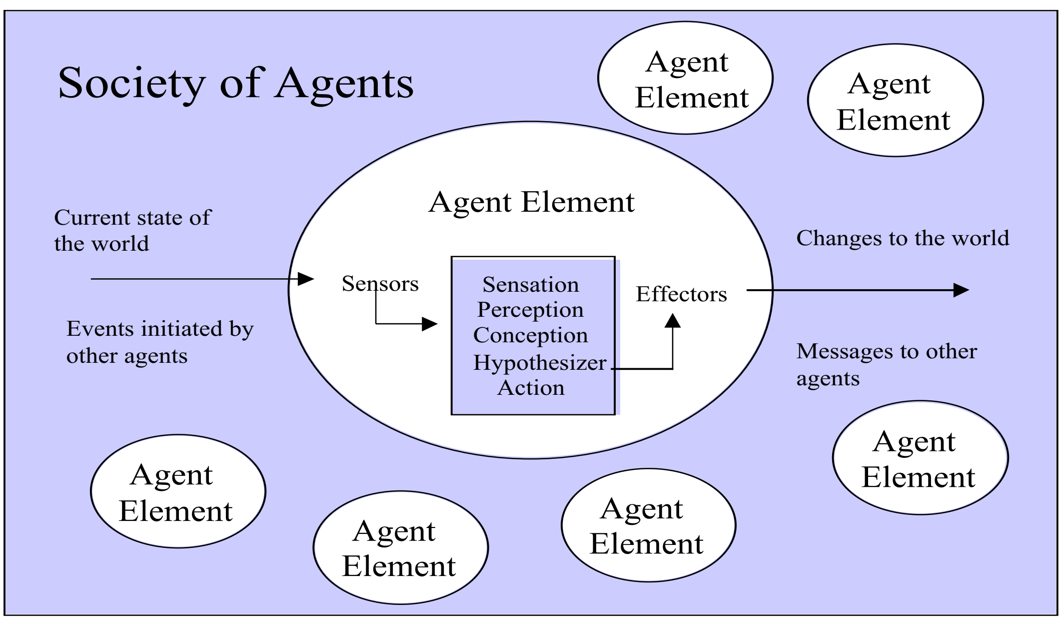

Figure 6. A virtual world as a society of agents.

Agent-based computing started in the 1970s, and recently the concept of agents has become important for internet applications, drawing ideas from Artificial Intelligence and Artificial Life. There is no universal definition for the term agent. However in the context of computer science, agents as intentional systems operate independently and rationally, seeking to achieve goals by interacting with their environment (Wooldridge and Jennings 1995). An agent has the ability to operate usefully by itself, however the increasing interconnection and networking of computers is making this situation rare. Typically, the agent interacts with other agents (Huhns and Stephens 1999). Hence the concept of multi-agent system is introduced with the applications of distributed artificial intelligence.

Object-oriented programming is one of the major types of programming methods. In object-oriented systems, objects are defined as computational entities that encapsulate some states, are able to perform actions, or methods on this state, and communicate by message passing. There are similarities between agents and objects, but there are also significant differences (Wooldridge 1999):

- Agents embody a stronger notion of autonomy than objects, and in particular, they decide for themselves whether or not to perform an action on request from another agent.
- Agents are capable of flexible (reflexive, reactive, reflective/proactive and social) behaviors, and the standard object model has nothing to say about such types of behaviors.
- A multi-agent system is inherently multi-thread, in that each agent is assumed to have at least one thread of control.

The intelligence of agents also reflects on its direct interaction with multi-agent environments. (Huhns and Stephens 1999) summarize the characteristics of multi-agent environments:

- Multi-agent environments provide an infrastructure specifying communication and interaction protocols.
- Multi-agent environments are typically open and have no centralized designers.
- Multi-agent environments contain agents that are autonomous and distributed, and may be self-interested or cooperative.

Multi-agent systems have been widely applied in many areas such as problem solving, collective robotics, kinetic program design and others (Ferber 1999). The ones are related to virtual worlds are:

- Multi-agent simulation: multi-agent systems bring a radically new solution to the concept of modeling and simulation in environmental sciences, by offering the possibilities of directly representing individuals, their behaviors and interactions. One of the examples is SIMDELTA. According to (Ferber 1999), the SIMDELTA simulator adopts a context of the fisheries of the central Niger delta in Mali. The idea is to model both quantitative data such as the evolution of the Niger's floods and qualitative data like fishing techniques.
- The construction of Synthetic Worlds: this type of application plays a large part in research into multi-agent systems. They do not enable anyone to solve specific problems, do not use physical agents and do not simulate any real worlds. A few of the examples are Hunt, introduced by (Ferber 1999), and examples of agents in virtual worlds using Java3D by (Maher and Smith 2001).

We propose a multi-agent system as the core of a 3D multi-user virtual world. Each object in the world is an agent in a multi-agent system. The agent model provides a common vocabulary for describing, representing, and implementing agent knowledge and communication. The agent can sense its own environment and can generate or modify the spatial infrastructure needed for a specific collaborative or communication need of the users of the world. Our common agent model is illustrated in Figure 7, where each agent has five kinds of reasoning: sensation, perception, conception, hypothesizer, and action.

The components of the agent element are described below.

- Sensors recognize two kinds of events: sense_data in which the agent identifies relevant data by monitoring the world and receive_data in which the agent receives a message from another agent. An example of sense-data for a virtual world agent is the ascii character stream that is the conversation in the environment of the agent

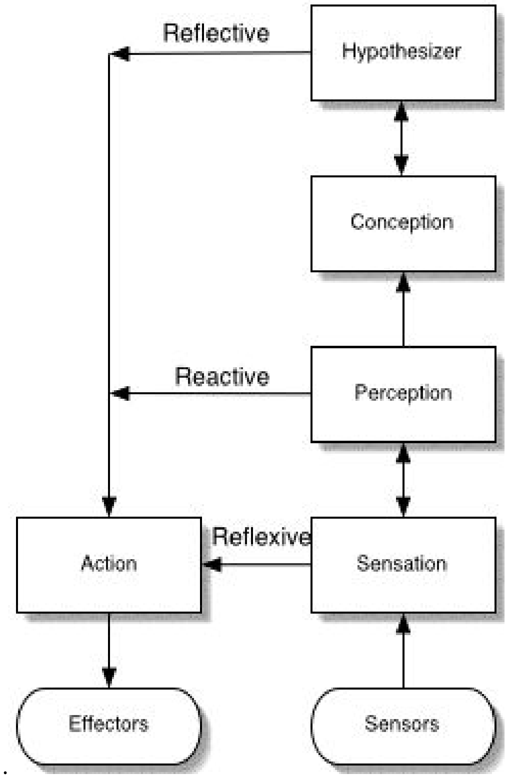

Figure 7. Agent model showing modes of behavior

- Sensation transforms raw input from the Sensors into structures more appropriate for reasoning and learning.
- Perception transforms sense-data into the percepts, or perceptual objects, that are used both to interpret interactions and as the units with which concepts are constructed. Percepts are grounded patterns of invariance over interactive experiences, and are constructed by clustering like patterns into equivalence classes so as to partition the sensory representation space. Perception is driven both by concepts and by the sense-data.
- Conception learns and uses concepts to reinforce or modify it beliefs and goals. Concepts are abstractions of experience that confer a predictive ability for new situations. The concept of a meeting, for example, is a representation of the activities of the agent with which meetings are involved, and its meaning is its predictions of possible interaction.
- Hypothesizer identifies mismatches between the current and desired situation, which goals are relevant to the current state of the world and reasons about which goal should be achieved in order to reduce or eliminate that mismatch. It identifies possible actions which when executed will change the world to meet those goals.
- Action reasons about which sequence of operations on the world, when executed, can achieve a specific goal.
- Effectors are the means by which actions are achieved. Two types of effectors are: Change_data in which the agent causes a direct change in the world, and SendMsg_data in which the agent sends a message to another agent to respond by changing the world.

This agent model is derived from recent developments in cognitively-based design agents, where design is considered as a situated act (Gero 1998). The agents are developed to interact with the design and the design knowledge (Smith and Gero 2001; Saunders and Gero 2001). The agent approach to virtual worlds provides for new kinds of interaction among the elements of the virtual world representation and between individuals and project teams with the components of the virtual world that makes both the virtual environment and interactions with it dynamic.

3.2 Agent Functionality

Agents can function in three modes based on their internal processes: reflexive, reactive, and reflective. Each mode requires increasingly sophisticated reasoning, where reflexive is the simplest. These modes are indicated in Figure 7 by labels on the paths through specific agent processes. A simpler reasoning involves fewer agent processes.

- *Reflexive mode:* here the agent responds to sense data from the environment with a preprogrammed response – a reflex without any reasoning. In this mode the agent behaves as if it embodies no intelligence. Only preprogrammed inputs can be responded to directly. Actions are a direct consequence of sense data. This mode is equivalent to the kinds of behaviors that are available in current virtual worlds.
- *Reactive mode:* here the agent exhibits the capacity to carry out reasoning that involves both the sense data, the perception processes that manipulate and operate on that sense data and knowledge about processes. In this mode the agent behaves as if it embodies a limited form of intelligence. Such agent behavior manifests itself as reasoning carried out within a fixed set of goals. It allows an agent to change the world to work towards achieving those goals once a change in the world is sensed. Actions are a consequence not only of sense data but also how that data is perceived by the agent. The agent's perception will vary as a consequence of its experience.
- *Reflective mode:* here the agent partially controls its sensors to determine its sense data depending on its current goals and beliefs; it also partially controls its perception processes again depending on its current goals and beliefs; its concepts may change as a consequence of its experiences. The concepts it has form the basis of its capacity to "reflect", ie not simply to react but to hypothesize possible desired external states and propose alternate actions that will achieve those desired states through its effectors. The reflective mode allows an agent to re-orient the direction of interest by using different goals at different times in different situations (Gero and Kannegiesser 2002).

3.3 Example of a Wall Agent

We illustrate the use of the agent model in the design of a wall agent for a virtual world. Although we isolate the wall for illustration purposes, the wall agent is one agent in a virtual world where all objects are agents. The wall is a combination of a 3D model that provides a visual boundary to a room and agent software that allows the wall to react to the use of the place. We can think of the world as being comprised of objects that have these two components, as shown in Figure 8. There are generic attributes of an object such as its owner and its location in the world. There are also general operations on an object such as move and clone. The attributes and operations specific to the 3D model and agent model are shown in the different components of the agent model. The agent component encapsulates the five processes/operations described above. The 3D model encapsulates the attributes and operations specific to manipulating the visual representation of the wall.

The room shown in Figure 9 is a seminar room designed for a group of 10-15 students. The room has four walls, on which is hung photographs of the students in the class. The colored panels at the top of the wall provides links to information associated with the course, such as reading material, recorded classroom discussions, and photos of classes in session. The place was implemented in Active Worlds and the interactivity provided by the predefined actions are primarily clicking to open a web page with information about the course.

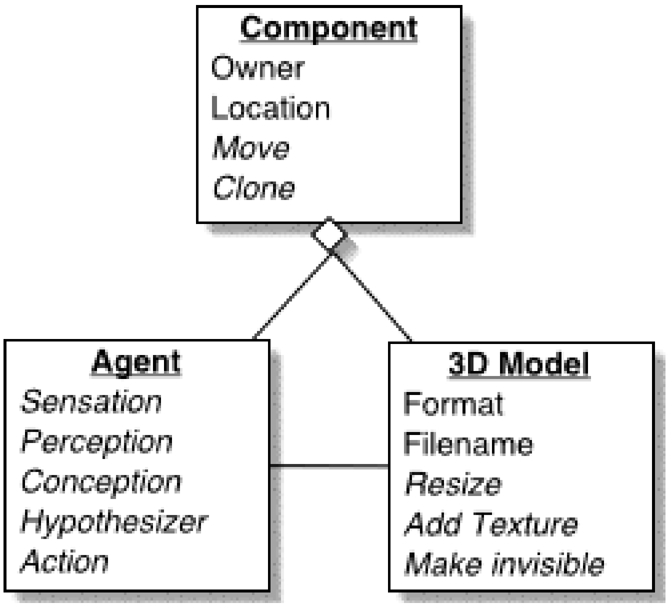

Figure 8. Each component of the virtual world has an agent model and a 3D model

Figure 9. Seminar room

When designing the wall as an agent using our agent model, we would like the wall to be able to respond to its environment by:

- Moving when the number of people in the room exceeds the capacity of the room
- Becoming opaque when visual privacy is needed and transparent when privacy is not needed
- Locking the room when interruptions are not allowed

The data in the virtual world that is to this agent includes:

- The locations of the other walls that form the room
- The number of avatars within the area enclosed by the walls
- The content of the conversations by the people/avatars in the room

Sensation is be the process of sampling the raw data in the virtual world at a regular rate and transforming the data relevant to the wall agent to a representation that allows further processing. The transformed data can be stored internally to the agent program or in an external database.

Perception organizes the sensed data into percepts such as number of avatars in the room, the area of the room, the type of meeting and references to security and privacy needs. These percepts can then provide the basis for expectations about the meeting.

The conception considers the percepts to determine if the room is too crowded and to determine if the use of the room requires security or privacy.

The hypothesis process would reason about goals such as the need to accommodate more people, a requirement for limited access to one or more spaces and the need for additional security. The goals would be achieved when certain concepts were true and the hypothesis process would reason about which goals it can achieve.

The action process would reason about how to achieve a goal, such as whether the wall should move and to what new location, whether the wall should change its transparency, and whether the wall should allow an avatar to pass through. The agent would then implement these changes and use its sensors determine whether they were successful.

The effectors would change the value of the representation of the 3D wall object by modifying its location, opacity, and solidity (a wall can have the solid attribute off which allows an avatar to pass through it, or on which prevents an avatar from passing through).

3.4 Generalizing the agent model

The example above shows how our agent model can be used for an intelligent wall object. By considering the elements of the model for a variety of specific virtual world agents, we can develop a generalization of the data and the processes that are relevant to any agent in a virtual world. Such a generalization makes it possible for generic agents to be attached to specific 3D models where the agent learns how to behave in the world. We attempt such a generalization here.

The sense data and the effectors provide the essential interface to the 3D world and the remaining parts of the model allow various behaviors to occur. We can generalize the agent model for an object in a virtual world as:

Sensation: The raw data for an agent is the properties of all objects in the virtual world. Sensation can be generalized to be the process of sampling the objects within a certain radius of a specific agent and detecting changes within the radius. In this process the proximal raw data is stored and a history is maintained for the agent.

Perception: This process looks for patterns in the data, such as objects in the world that are adjacent and space enclosing, objects such as avatars that form clusters because they are near each other, and key words or repeating topics in conversations.

Conception: This process can be generalized as a recognition of concepts that are important in a virtual world, such as size of enclosed space, availability of certain functionality, need for security and privacy.

Hypothesizer: Once the concepts are generalized, the hypothesizer can be implemented as a process of finding mismatches between the current state as defined by the perception process and the desired state and understood by the conception process. When a mismatch is recognized, the hypothesizer can establish a goal to reduce the mismatch.

Actions: The actions are the means for an agent to achieve a goal identified by the hypothesizer. The generalized actions for a virtual world agent include:

- change location of itself or another object
- change the size of itself or another object
- change the functional parameters of itself or another object
- change the visual parameters of itself or another object

4 Conclusions

The agent model developed in this paper shows how virtual worlds can provide environments that respond automatically to their use. This response can result in a dynamic world that configures and reconfigures itself as needed. This is not the same as an event driven interface that current virtual worlds have because the components of the agent based world reason about the state of the world rather than wait for an event to trigger an action. Our agent model assumes different levels of reasoning that provide flexibility in the behavior of the agent. By separating sensation, perception, conception, hypothesizing and action, we can develop intelligent objects that reason and act on different levels of abstraction. This effectively defines an intelligent world as a society of intelligent agents.

Acknowledgments

This research is funded in part by the Australian Research Council Large Grants Program. Many of the ideas have been developed through discussion with our research students and staff. Specifically, we would like to acknowledge Ning Gu, Greg Smith, and Udo Kannengiesser.

Notes

1 http://www.blaxxun.com

2 http://www.ccs.neu.edu/home/ eostrom/muds/lambdamoo.html

3 http://www.activeworlds.com

4 http://www.vworlds.org

5 http://www.vtools.com

References

Ferber, J.: 1999, *Multi-Agent Systems: An Introduction to Distributed Artificial Intelligence*, Addison Wesley Longman, England.

Gero, J. S. (1998) Conceptual designing as a sequence of situated acts, *in* I. Smith (ed.), *Artificial Intelligence in Structural Engineering,* Springer, Berlin, pp. 165-177.

Gero, J. S. and Fujii, H. (2000) A computational framework for concept formation in a situated design agent, *Knowledge-Based Systems* **13**(6): 361-368.

Gero, JS and Kannengiesser, U (2002) The situated Function-Behavior-Structure framework, *in* JS Gero (ed.), *Artificial Intelligence in Design'02,* Kluwer, Dordrecht, pp. 89-104.

Gu, N. and Maher, M. L.(2001) Architectural design of a virtual campus *in* Y-T. Liu (ed.), *Defining Digital Architecture,* Dialogue, pp. 158-161.

Huhns, M.N. and Stephen, L.: 1999, Multiagent Systems and Society of Agents, *in* Weiss, G. (ed.), *Multiagent Systems, A Modern Approach to Distributed Artificial Intelligence*, MIT Press, MA, pp.79-120.

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

Maher, M. L., Gu, N. and Li, F. (2001)Visualisation and object design in virtual architecture *in* J. S. Gero, S. Chase and M. Rosenman (eds), *CAADRIA2001*, Key Centre of Design Computing and Cognition, University of Sydney, pp. 39-50.

Maher, M. L. and Gu, N. (2001) 3D virtual world, *in* M. Engeli and P. Carrard (eds), *ETH World: Virtual and Physical Presence,* Karl Schwegler AG, Zurich, pp. 146-147.

Maher, M. L., Simoff, S., Gu, N. and Lau, K. H. (2001a) Virtual conference centre *in* M. Burry (ed.), *Cyberspace: The World of Digital Architecture*, Images Publishing, Mulgrave, Vic, pp. 192-195.

Maher, M. L., Simoff, S., Gu, N. and Lau, K. H. (2001b) A virtual office *in* M. Burry (ed.), *Cyberspace: The World of Digital Architecture,* Images Publishing, Mulgrave, Vic, pp. 196-199.

Russell, S. and Norvig, P.: 1995, *Artificial Intelligence: A Modern Approach*, Prentice Hall, Englewood Cliffs, NJ.

Saunders, R and Gero, JS (2001) A curious design agent, *in* JS Gero, S Chase and M Rosenman (eds), *CAADRIA'01*, Key Centre of Design Computing and Cognition, University of Sydney, pp. 345-350.

Smith, G and Gero, JS (2001) Situated design interpretation using a configuration of actor capabilities, *in* JS Gero, S Chase and M Rosenman (eds), *CAADRIA'01,* Key Centre of Design Computing and Cognition, University of Sydney, 2001, pp. 15-24.

Smith, G.: 2001, Preliminary Report on Agents in Active Worlds, *Working Paper,* Key Centre of Design Computing and Cognition, University of Sydney, Australia.

Wooldridge, M.: 1999, Intelligent agents, *in* Weiss, G. (ed.), *Multiagent Systems, A Modern Approach to Distributed Artificial Intelligence*, MIT Press, MA, pp.27-78.

Wooldridge, M. and Jennings, N. R.: 1995, Intelligent agents: Theory and practice*, Knowledge Engineering Review* **10**(2): 115–152.