INTELLIGENT BEHAVIOR CONTROL OF 3D OBJECTS IN VIRTUAL ENVIRONMENTS

JAEWOOK LEE, YONGWOOK JEONG, SEUNG WOOK KIM, YEHUDA E. KALAY
Digital Design Research Group, Department of Architecture, University of California, Berkeley, USA
{jaylee69, ywjeong, swkim}@berkeley.edu,
kalay@socrates.berkeley.edu

Abstract. Cyberspace is more malleable than a physical environment, so it can afford much wider range of responsiveness. By applying the concept of place-making, we are experimenting virtual environments which are responsive to their users' context-specific needs. Since objects are essential components that anchor the users' various activities, having interactive objects in a 3D virtual environment is a major design concern for developing a dynamic and experience-rich virtual environment. We propose a layered agent model for intelligent behavior control of 3D objects, based on constraint solving process.

1. Introduction

Nicholas Negroponte, in his book *Soft Architecture Machines* [1975], stated that an intelligent environment would be responsive to occupants' requirements by changing itself dynamically as its context changes. Recently, many researchers have equipped building components with some degree of intelligence so they can self adjust their behaviors according to environmental changes. Information technology has given a great impetus to developing such intelligent building systems, ranging from 'smart' materials to 'intelligent' whole buildings. Some of these developments have already been applied in practice (e.g. NASDAQ MarketSite Tower [Kalay 2004], Digital Desk project [Wellner 1993], ACHE project [Mozer 1988], iRoom project [Peter and Kaj 2000], iDorm Project [Holmes et al 2003], etc.). Such intelligence turns a building environment from a passive entity into an active participant that can respond to its users' needs.

In general, an intelligent building system selects a sequence of behaviors based on sensing and responding mechanisms. The type of responsiveness and its degree of adaptability depend on its level of autonomy and can be categorized as *feedback regulated*, *model-based*, or *total environmental adaptability* (Kalay 2004). However, their functionality is limited due to the static nature of physical architecture. In contrast, cyberspace—as an information space—is much more malleable and does not have the limitations that a physical environment has. As such, it can afford a much wider range of responsive environments.

Currently, we are developing 3D virtual learning environments (Figure 1) as alternatives to physical environments, where users enjoy a variety of place-based learning experiences. To build such virtual learning environments, we have applied the concept of place-making that would let the environment be responsive to its users' context-specific needs (Kalay 2003). Context-sensitivity, according to Canter [1977], Champion and Dave [2002], and others is critical to interpretation of the meaning of an activity. In this paper we present a method of making such a context-dependent virtual environment responsive to users' needs. We propose a layered agent model for intelligent behavior control of 3D objects, based on constraint solving process for a dynamic virtual environment.

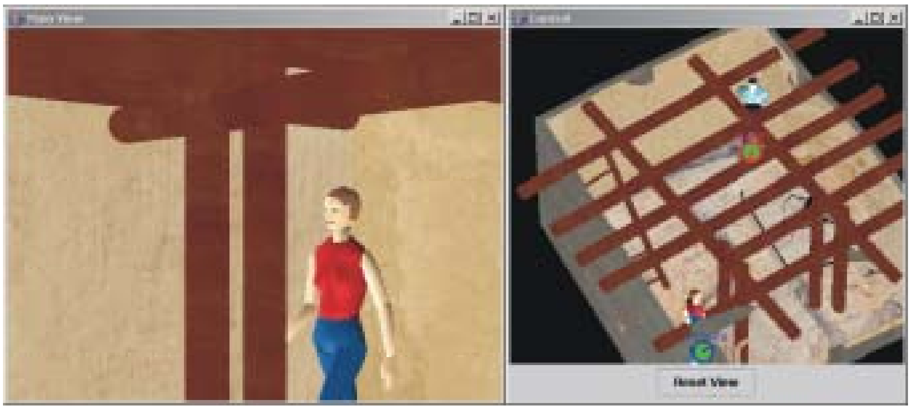

Figure 1. VIP (Virtual Inhabitation and Presence) Project

2. 3D Virtual Environments

Whether in a physical or virtual environments, objects are essential components that anchor the users' various activities. Therefore, having interactive objects in a 3D virtual environment is a major design concern for developing a dynamic and experience-rich virtual world.

As hardware and software technologies advance, inhabitable 3D virtual environments are becoming more popular and readily available over the World Wide Web (e.g. Active Worlds, Second Life, etc). However, although some researchers have taken steps towards their incorporation (e.g. Maher and Gero [2002]), in the majority of these environments objects still lack intelligence and are not fully responsive to users' needs. Instead, their behaviors are static and arbitrarily determined.

In general, the components that comprise a virtual environment and their characteristic can be briefly described as follow:

- ***User***, in the form of an avatar or a simple cursor/keyboard input that generates specific activities.
- ***Activity*** determined by a user and represents the user's desire at any given time in the environment. It is usually associated with one or more objects.
- ***Object***, which works as a medium to support users' activities.
- ***Context***—a state of the environment that affects overall user and object behaviors. For example, soccer players' behavior patterns strongly depend on the context of a match (winning vs. losing).

To properly design a dynamic virtual environment, interactions among these components should be carefully considered. In our proposed system, when a user execute an activity, objects (represented as virtual agents) will perceive it and select an appropriate sequence of behaviors to support user's desire, while sensitive to the context of that action.

3. Layered Multi-Agent System

We are developing a layered multi-agent system that can model and execute such action sequences. In our approach, each object is represented as an intelligent agent that knows how to behave given any activity of a user, and has the ability to perceive contextual changes of the environment and adjust its behaviors in accordance with its surrounding context. Through this dynamic behavior adjustment, object agents can perform different behaviors in response to the same activity in different contexts. This entire process is done in real-time through sensing, reasoning, and acting mechanism of each object agent.

By definition, agents are autonomous or semi-autonomous hardware or software systems that perform tasks in complex, dynamically changing environments [Muller 1996]. Agent technology has been widely used in many industries, including robotics, process control system, email clients, and search engines, for the purpose of developing intelligent applications. It is not a single technology but rather an integrated application with multiple technologies [Caglayan et al 1997]. Since we are dealing with a network-based multi-user environment where dynamic interactions among different entities happen, we found that an agent-based approach has promise in developing a dynamic 3D environment.

Agent-based systems comprise of different types of agents, determined according to the nature of task environment. A certain type agent may work well in one environment, but may not in another. Similarly, a dynamic virtual environment can be designed with either a single, omniscient agent, or a multiagent system. A multiagent system seems more appropriate for our needs than a single agent system, because it is almost impossible to create a single omniscient agent that knows everything about the environment, and is not encumbered by speed, reliability, maintainability, and so on.

Several researchers have already applied multiagent technology for constructing a dynamic virtual environment. A virtual human simulation system by Farenc et al [2000] applied a multiagent technology for managing (human) agent - (3D virtual) object interactions. They used predefined behavior rules to control both human and object behaviors during the simulation. Since the main purpose of the system was to simulate human behaviors in a virtual environment, little attention was paid to dynamic object interactions and contextual changes. Maher and Gero [2002] focused on dynamic interactions among objects in a virtual world where each object is an agent that can sense, reason, and respond to the environment according to its goal and belief. Their agent model may work in an environment with limited number of objects and less significant contextual changes. But since only a limited portion of the environment data is available to each individual agent, this system may find it difficult to maintain reasoning consistency about the world: one agent may perceive its current state of the world differently from how other agents do. Such inconsistency may lead to behavior conflicts among agents.

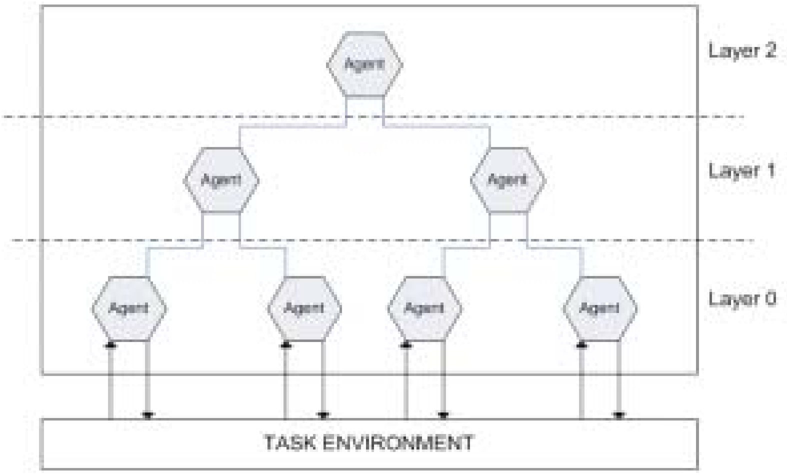

Figure 2. Layered Agent Model: Three Layers

Because we deal with a complex task environment that is dynamic and partly-observable, to ensure reasoning consistency and avoid behavior conflicts among agents, we have applied a layered agent model (Figure 2). The hierarchical structure divides the work into tasks and simplifies the management of interactions among different agents. The behaviors of agents in lowest level of the hierarchy are considered simple atomic actions, while upper layer agents have more abstract behaviors (Scerri 1998). Agents need only to look up for behavior instruction from their supervisors in the hierarchy, then look down to get help in carrying out their tasks from agents in lower layers (Minsky 1986). For instance, when an upper layer agent views privacy as the most dominant context of a room, it will instruct a door, which an object agent in a lower layer, not to allow itself to open. The opening of the door is considered a lower-level behavior, which the upper level agent need not be concerned with, whereas the privacy issue is considered a higher-level behavior that the door agent does not know about.

4. Constraint-Based Behavior Management

In a complex, dynamic environment, every user-initiated activity is associated with one or more objects, which may interact with other object(s). Object behaviors need to be selected in accordance with interrelationships among objects. Using the above listed example, a door object may respond differently to a user-initiated opening action, depending on privacy conditions established by other users. We propose to use constraint-based behavior management to dynamically modify the behaviors of agents in a 3D virtual environment. High level constraints establish a ‘current context’ for the environment, which establishes behavior constraints for objects in an active role. Every object performs a set of behaviors to satisfy given constraints. The context constraints, in turn, are informed by the active objects, and so on. Depending on the state of an environment, numbers and types of active constraints can be different. User requirements can be described as constraints that objects in the environment must satisfy.

Every time an object is added to the environment, corresponding behavior constraints are assigned to it. Thereafter, the object will behave in a particular way to satisfy these constraints (Kalay 2003). If an object with constraints is modified, the system will propagate the changes through other constrained objects within the environment until all the constraints are satisfied. By doing so, the system will dynamically modify the environment as a whole.

Constraint satisfaction technique has been widely used in space planning and solid modelling problems. SKETCHPAD (Sutherland 1963) was the first graphic editor to employ constraint-based modelling concept. Other researchers have also shown the potentials of constraint-based modeling approaches in 3D architectural modeling in virtual environments (e.g. Fernando et al 1999, Le Roux et al 2000, Eggink et al 2001). However, most of these projects only applied geometric constraints for manipulating virtual environments.

Since our proposed system is built on a network-based multi-user environment, a substantial number of objects can be placed in the environment simultaneously. In such a case, although only active constraints that were filtered by the context are given to corresponding objects, behavior conflicts among the objects may arise. For example, when a room (R1) needs to be enlarged, another room (R2) which is connected to the R1 may not allow R1’s enlargement due to R2’s internal context. To avoid this kind of conflict, the behavior management process prioritizes behaviors and selects the most desirable one by evaluating the impacts of the conflicting behaviors in advance.

5. System Overview

The system consists of three major hierarchical agents; Object Agent (OA), Behavior Manager Agent (BMA), and Environment Monitor Agent (EMA). At a given point of time, behaviors of each active agent are decided through communication among agents in the same or different layer(s). Agents and their tasks can be summarized as follow:

- ***Object Agent (OA)***: The lowest-layered agent interacts directly with the 3D virtual environment
- ***Behavior Manager Agent (BMA)***: The middle-layered agent processes behavior roles and constraints
- ***Environment Monitor Agent (EMA)***: The highest-layered agent monitors the context of the environment

A set of generic behaviors within each agent, which includes triggering methods and times, is defined according to its object type. For example, a door object can have generic open and close behaviors (e.g. swing open/close, sliding open/close, etc.). The overall context of an environment is determined by the EMA based on three factors; social, cultural, and physical factors. This context information is passed to the BMA, which further establishes specific roles and behavior constraints of individual objects and passes them to the OAs. Referring to the roles and associated constraints from the BMA, the OAs are ready to respond to user's input by choosing the best behavior(s) among the generic behaviors (Figure 3). Consequently, an object may perform a different behavior depending on the given context. For example, the behavior of a table object in the context of studying can be different from its behavior in the context of dining.

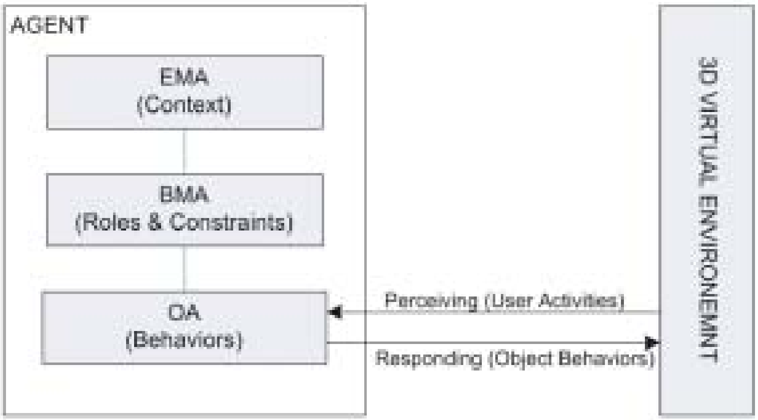

Figure 3. Agent Interaction Overview

When a user initiates an action on an object in the environment, it will trigger the object agent to be responsive to the given action. After communication and reasoning process between the active object and other associated agents, one or more behaviors will be executed by the object agent. If there exist one or more of additional object agents that are constrained by the formerly executed behavior, they will also perform the constrained behavior(s) in turn.

6. Case Study

In order to examine our approach, we have developed a simple virtual environment as a case study. The test environment is composed of five objects: two box objects, a door, a wall, and a floor (Figure 4). Here, the wall object divides the space into the two separate rooms as sub-divisions of the total environment. Although there might be additional wall objects in order to define the two rooms, we leave them out of consideration, since only the wall with the door is an active object in this case.

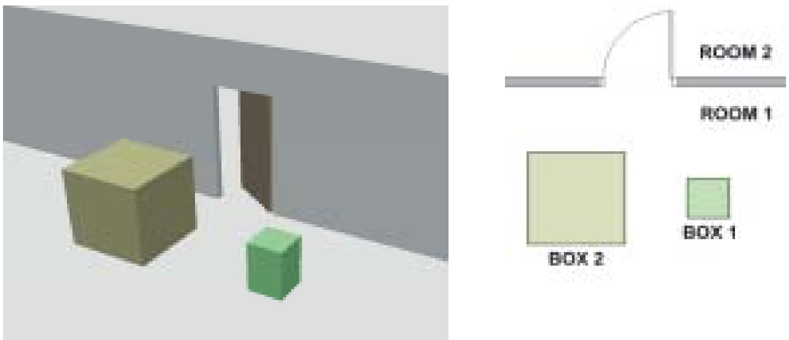

Figure 4. Case Study

Case 1
Suppose a user wants to move box1 (B1) from the room1 (R1) to the room (R2) through the door. When the user grasps B1 and approach to the door, the door object detects B1 and consults the BMA to check the active constraints in the current state to select its behavior(s). If the current context of R1 is a private condition, the door does not allow incoming traffic and thus it remains locked. On the other hand, if the BMA allows public traffic, the door detects B1 and checks its size. Since the size of the B1 is smaller than the width of the door, there is no additional active constraint. Then the door simply chooses the swing-open behavior and executes it, so that B1 can be moved to R2.

Case 2

In the case of moving B2 to R2, the internal process is much more complicated: because we are dealing with a virtual environment, we have much more freedom in designing behavior patterns of any object. Therefore, there are several possible ways that will yield a same or a different result.

- *Instant Transporting*: In this case, there is no active constraint other than getting permission from the BMA. After BMA's acknowledgement, B2 will simply be transported to R2 instantly without any geometric change.

- *Resizing*: If we stick to physical constraints in this virtual environment, object resizing would be more intuitive. Resizing method also can be applied in two ways. Either changing the size of the door or changing the size of B2 are possible (i.e. enlarging the door or shrinking B2). In either of these cases, the shape constraint is the active constraint that needs to be satisfied. When the door enlarges its size, the shape constraint will be activated in both the door and the wall objects (Figure 5). Since the door is strongly constrained with the wall, the enlarging process should be applied to both objects simultaneously (i.e. the wall object should enlarge its opening size in respect to the door size). Once B2 is transported to R2, both the door and B2 return to the default condition. In case of shrinking the size of B2, the shape constraint is only applied to the object itself (i.e. shrinking – transporting – enlarging). Thus changing the size of B2 may be a more practical solution in this case.

- *Blocking:* Further complications may arise if the size of the door is indicative of some internal condition of the room. For example, it may indicate that only objects of a certain size may be allowed to enter the room. This condition may be detected by the EMA, which will not allow either of the former two cases, until some other objects inside the room have been removed (or re-arranged).

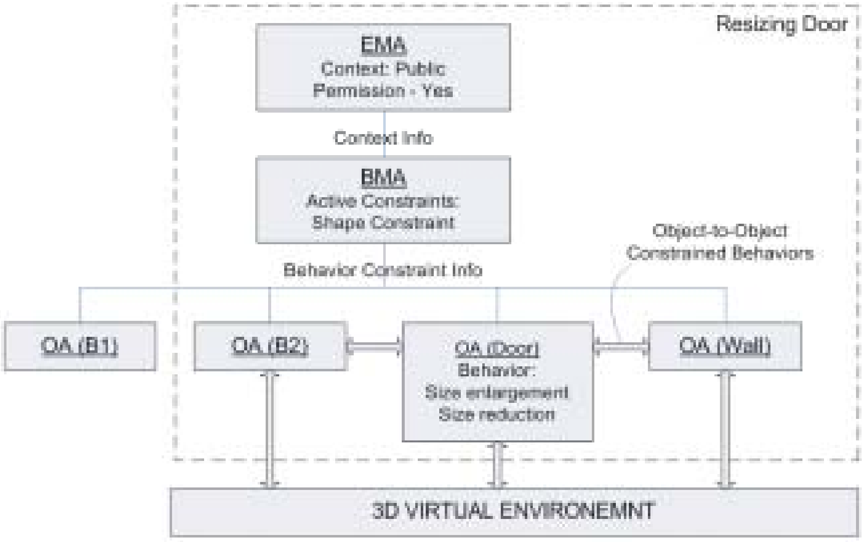

Figure 5. Behavior Managing Process: Case 2 – Resizing Door

A designer of this kind of environment may include all the possible behavior patterns listed above in the object description of a door object. However, at any point of time, the door should choose the best possible behavior that can satisfy both designer's intention and user's desire through agent-to-agent communication.

7. Conclusion and Future Works

In this paper, we discussed intelligent behavior control in a 3D virtual environment. The main idea of the work we presented here is to enhance dynamics and intelligence of a virtual environment as a whole to better support users' experience-rich activities. In this respect, a layered agent model can be an applicable solution to endow objects with contextualized behaviors in a given environment. The context of an environment works as a major driver for objects to decide their behaviors accordingly. Furthermore, its constraint-based behavior management can materialize object dynamics in addition to intelligent behavior control.

However, in order to make our proposed system practicable, we still have many tasks to do and some fundamental issues to solve. First, we must carefully design a performance measure for the success of each object behavior. In general, performance measures should be designed according to what users actually want to do in the environment rather than according to how a designer thinks the agent should behave (Russell and Norvig 2003). Although this performance criterion can be given by the designer, it is still not easy for her/him to determine the degree of desirability of any environmental changes made by any object behavior. This should be based on measuring how much a user is satisfied with the consequences of any object behavior. Second, a method of quantifying the context and its changes need to be carefully investigated for evaluating the environment as whole. This will ensure systematic role and behavior assignment for objects in a virtual environment. Last, we also need to take into account how to trade off between group preference and individual one. Since it is a multi-user environment, any agent behavior may work for one user but may not for another. Solutions for these issues can further enhance inhabitability as well as responsiveness of 3D virtual environments.

Acknowledgement

This research is part of the VIP project by DDR (Digital Design Research) Group in Berkeley. We would like to thank every project members for their inspiration on this research work.

References

Caglayan, A. K. and Harrison, C. G.: 1997, *Agent Sourcebook*, John Wiley & Sons, New York.

Canter, D.: 1977, *The psychology of place*, St. Martin Press, NY.

Eggink, D., and Gross M. D.: 2001, *Smart Objects: Constraints and Behaviors in a 3D Design Environment*, Education in Computer Aided Architectural Design

Farenc, N., S.R. Musse, E. Schweiss, M. Kallmann, O. Aune, R. Boulic and D. Thalmann: 2000, *A Paradigm for Controlling Virtual Humans in Urban Environment Simulations*, Applied Artificial Intelligence Journal, Special issue on intelligent virtual environments.

Fernando, T., N. Murray and K. Tan: 1999, *Software Architecture for a Constraint Based Virtual Environment*, VRST 99, New York: ACM Press, pp. 147-154.

Holmes, A., H. Duman, and A. Pounds-Cornish: 2002, *The iDorm: Gateway to Heterogeneous Networking Environments*, International ITEA Workshop on Virtual Home Environments.

Kalay, Y. E. and Marx, J.: 2003, *Changing the Metaphor: Cyberspace as a Place*, Proceedings of the 10th Computer Aided Architectural Design Futures.

Kalay, Y. E.: 2004, *Architecture's New Media: Principles, Theories, and Methods of Computer-Aided Design*, MIT Press, Cambridge, MA.

Khemlani K., A. Timerman, B. Benne and Y. E. Kalay: 1998, *Intelligent Representation for Computer-Aided Building Design*, Automation in Construction.

Krogh, P., Grønbæk, K.: 2001, *Roomware and Intelligent Buildings - objects and buildings become computer interfaces*, Proc. of Conference on Architectural Research and

Information Technology. April 27-29, Århus School of Architecture. Nordic Association for Architectural Research, Århus: pp. 63-68.

Le Roux, O., V. Gaildrat and R. Caubet: 2000, *Design of have New Constraint Solver for 3d Declarative Modeling: JACADI*, 3IA: Interactive computer graphics and Artificial Intelligence, Limoges.

Maher, M. L. and Gero, J. S.: 2002, *Agent Model of 3D Virtual Worlds*, Proceedings of the 22nd Association for Computer Aided Design in Architecture.

Martini K. : 1995, *Hierarchical geometric constraints for building design*, Computer-Aided Design 27(3).

Minsky, M.: 1986, *The Society of Mind*, Simon and Schuster, NY.

Mozer, M. C.: 1998, *The neural network house: An environment that adapts to its inhabitants*, Proc. AAAI Spring Symposium on Intelligent Environments, AAAI Press, pp. 110-114.

Muller, J. P.: 1996, *The Design of Intelligent Agents,* Springer, Berlin.

OMG Agent Working Group: 2000, *Agent Technology Green Paper*, Technical Report, Object Management Group.

Russell, S. and Norvig, P.: 2003, *Artificial Intelligence: a Modern Approach (2nd Edition)*, Prentice Hall Series in Artificial Intelligence, New Jersey.

Scerri, P.: 1998, *A Multi-Layered Behavior Based System for Controlling RoboCup Agents*, RoboCup-97: The First Robot World Cup Soccer Games and Conferences, Springer Verlag Lecture Notes in Artificial Intelligence.

Sutherland, I. E.: 1963, *Sketchpad: A Man-Machine Graphical Communication System*, Garland Publishing, NY.

Vries, B. de, A.J. Jessurun and R.H.M.C. Kelleners: 2000, *Using 3D Geometric Constraints in Architectural Design Support Systems*, Proceedings of the 8th International Conference in Central Europa on Computer Graphics, Visualization and Interactive Digital Media.

Wellner, P.: 1993, *Interacting with Paper on the DigitalDesk*, CACM.