A Usability-based Building Model for Environmental Behavior Simulation

Wei Yan[1], Yehuda Kalay[2]
[1]Texas A&M University, USA, [2]University of California, Berkeley, USA

This paper describes a usability-based building model we developed for behavior simulation in design phase and a systematic approach to implement it. Our model possesses not only geometric information of design elements but also non-geometric information of these elements, specifically, the relationships between the elements and the intended users. This relationship, which was implicitly understood by the designer, now becomes explicit to the simulated users. It utilizes CAD models and automates the conversion from CAD models to usability models and therefore requires minimal additional effort in preparing building data for behavior simulation, which has proven to be efficient and effective.

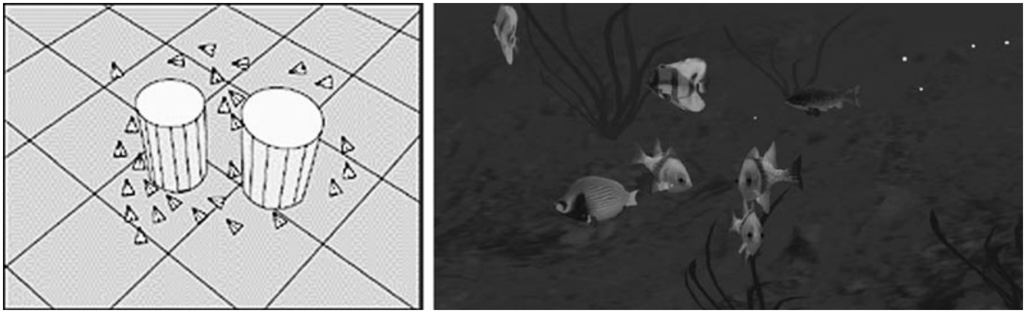
Keywords: *Usability; Building Model; Behavior; Simulation; Virtual User.*

Introduction

Environmental behavior simulation can be used to predict and evaluate the impact of environments on human behavior and it is of great interest to designers and clients. Research has been conducted in simulation of specific human behaviors, such as pedestrian traffic, fire egress, activities in open public spaces, etc. (e.g. Haklay, et al., 2001; Helbing, et al., 1997 and 2001; Ozel, 1993; O'Neill, 1992.) Modeling the characteristics of the environment that directly affect human behavior (as well as user modeling) is a key to such simulations. Attempts to include such information in otherwise geometric models have become necessary along with the recent trend in behavior simulation based on agent-based approaches, such as Artificial Life algorithms (Batty, 2001; Kerridge, et al., 2001). However, built environments are relatively more complicated, compared to environments of the kind used in Artificial Life simulations—where autonomous life-like agents fly in the sky or swim in the tank with certain motivations and according to certain behavior rules. For example, Boids environments (Reynolds, 1987) focus on simulating interaction among bird-like agents during flight, while avoiding obstacles (Figure 1 left). Similarly, the Artificial Fish (Tu, 1996) environment consists of water current, seaweed and plankton. The fish are able to perceive their environments and act by avoiding collisions and pursuing a moving target (Figure 1 right). In contrast, built environments consist of a large number of design elements, such as walls, doors, windows, stairs, columns, benches, fountains, lawns, paving, etc. (Figure 2). A simulated user (a virtual user, see Yan and Kalay, 2006) needs to perceive and understand them – including both geometric and non-geometric information – in order to behave properly, e.g. walk through a door, stand by a fountain and sit on a bench.

For our simulation, we seek an approach to model the built environment to represent not only the geometry, but also the meaning of the design

Figure 1
Left: Boids: bird-like objects schooling and avoiding obstacles. Source: Reynolds (1987); Right: Artificial fish simulation. Source: Tu (1996).



objects, i.e. the types and usability of the design elements, so virtual users can perceive and understand them and know, for example, whether they encounter a stair, a wall, a bench, or a tree. An object like a bench possesses graphical properties of geometry, location, and orientation. In addition, it also has usability properties such as how many people can sit on it, whether it is movable, whether it is occupied by a user and so forth. The virtual users can understand this information and can respond to it, e.g., to find an empty bench when they are so inclined and use it in the proper way (i.e., sit on it).

Earlier work in building models includes geometric models and object-oriented models, which are built on graphics-based CAD foundations and as a result are not suitable for creating and managing information about a building. Recent developments, the Building Product Models and Building Information Models (BIM), are capable of representing non-geometrical information of building environments to facilitate interoperability and productivity (see Eastman, 1999; Khemlani, 2003), but require considerable effort to construct and update.

Computer-based 2D drawings and 3D models are used in design phases for communication, presentation, and evaluation in terms of energy, lighting, acoustics, esthetics, etc. The models contain very rich graphical information of the geometry and some textual information about design elements' types, e.g. the layer names in layer-based models. We consider the rich graphical information and limited textual information in CAD models potentially very useful for our modeling. For example, the element names (e.g. layer names) that are stored in CAD files or databases would be accessible to virtual users as a resource that can be further processed to let the virtual users understand the type of each element.

However, CAD models are not immediately applicable in the simulation because they lack the usability information. For example, a flat surface without a name can be either the top of a bench that a user can sit on or the ground that the user can only walk on, which the virtual users cannot tell. Even with a name, e.g. Fountain-Side (a low wall around a fountain), the virtual users still can't tell whether it can be used for sitting or not. Moreover, the data structure of CAD models, e.g. the layer-based structure, does not allow users to gain spatial information, e.g., where they are and where to go, in the simulation. What we need is a new model containing both geometrical information derived from CAD models and properties of usability,

Figure 2
A university campus plaza.

which are non-geometrical information.

Normally, to evaluate the building's performance from CAD models required expensive manual translations into specialized evaluation software, e.g. for energy consumption, where a human operator identified and distinguished the various building components (Kalay, 2004). We have developed a usability-based building model for the purpose of behavior simulation in design phase, and a systematic approach to implement it. Unlike traditional CAD models, our model possesses both geometric information of design elements and non-geometric information about the usability properties of these elements. The environmental information is structured in a way that makes it perceivable and interpretable by the virtual users. To be more practical, the building model is not created from scratch. Instead, it is built on top of existing CAD models that come from everyday architectural design practices. Such CAD models already contain complete geometrical information but lack the non-geometrical information concerning usability of design elements, and therefore cannot be directly understood by virtual users. Our method can automatically convert standard CAD models (e.g. in DXF format) to usability-based models that are ready for the human behavior simulation.

Utilizing CAD models makes our simulation practical and eventually applicable to everyday design practice, which is the ultimate objective of this research. We have conducted a case study with a campus plaza – Sproul Plaza at the University of California at Berkeley, which contains distinctive paving, a fountain with low seating edge, large area of steps, and a few benches (Figure 2). By inserting the virtual users in our usability-based building model and letting them "explore" it on their own volition, our system reveals the interrelationship between the environment and its users.

Existing building models and our approach

Our building modeling fits in the paradigm of Building Information Modeling, with a specific scope focusing on the usability properties of design elements.

Building Information Modeling recently drew great attention in A/E/C community. It is a new generation of CAD models although its original idea can be traced back to models such as Building Product Model (Eastman, 1999). Previous generations of CAD models include graphic models and object-oriented models. In A/E/C, information is often presented graphically – in sketch, schematic design drawings, and construction drawings. Graphic models visualize the spatial configuration of the designed environments. However, graphical information has its limitation in the following two aspects: (a) It is not sufficient for automatic design evaluation, which employs computer programs to evaluate the performance of a design in terms of spatial use, e.g. people and traffic flows, cost estimation, energy consumption, lighting effect, government code requirements, etc. (b) It is difficult to be shared among professionals in design, and interoperability is not well achieved among architecture, interior design, structural engineering, mechanical engineering, and construction. Both evaluation and interoperability need semantics in additional to graphical information: properties of design elements, such as geometry, materials, and finishes, etc.

In the early 1990s, the introduction of object-oriented CAD brought data "objects" into the models. The objects in these systems—doors, walls, windows, roofs, etc. —stored non-geometrical data about a building in a logical structure together with the building graphics. However, object-oriented CAD systems remain rooted to building graphics, built on graphics-based CAD foundations, and as a result are not fully suitable for creating and managing information about a building (Autodesk 2002).

The Building Information Model, differing from its predecessors, contains not only a list of building components and locations but also the relationships among the components. Our building model, based on similar concepts to Building Information Models, extends the relationships to that between the building components and the intended users. This relationship, which was implicitly understood by the designer, now becomes explicit to the virtual users, e.g. that the fountain-side in a plaza can support the activity of sitting becomes explicit in the model.

We have created our building model by converting the traditional graphic models into usability-based models automatically. Our approach is not tied to any particular CAD software. Rather, it is designed as a reusable tool for working with traditional graphic models and Building Information Models.

Geometry modeling

To make the case study simple without losing generality, we used the popular layer-based DXF format to build the CAD model of Sproul Plaza based on field survey and measurements (Figure 3). The rectangular area in the center of the plaza is the target region of our simulation.

In the CAD model, a rectangle on one layer can represent a bench, but on another layer can represent a building roof. To assign meanings to the geometries for preparing for the translation from the DXF model to the usability-based model, we used a regularized naming convention for the layers, and made sure each design component was placed in its appropriate layer. For example, LowWall-FountainSide is the name of a design component of the type LowWall, and its ID is FountainSide, meaning it is a low wall around the fountain; and Step-HallLevel1.1 is the name of a design component of type Step, and its ID is HallLevel1.1, meaning it is a step in front of the Hall and is the first step of the first stair. The layer name list of the model is shown in Table 1. We also marked the components' z-buffer (depths) according to their actual spatial relationships, e.g. fountain-side is on top of the ground.

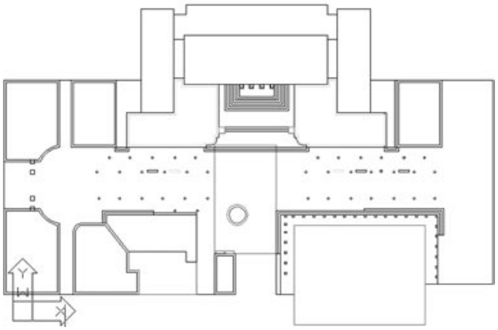


Figure 3
The plaza's CAD model in DXF format.

Table 1
Layers in the CAD model

Step-HallLevel1.1
Ground-plazaGround
Lawn-HallLeft
Lawn-HallRight
LowWall-FountainSide
....

By naming the design components in an easy-to-follow convention and marking the components' z-buffer correctly, the CAD model is ready to be converted to a usability-based model.

Usability modeling

We first convert the CAD model into an intermediate model in Scalable Vector Graphics format, and then built a discrete spatial model, which became our usability model.

Converting layer-based model to an intermediate model

To model usability information, we applied Scalable Vector Graphics (SVG), which is a language for describing two-dimensional graphics in XML (W3C Recommendation). Using a graphics formatter, an SVG document can be presented in a graphical way, e.g. a design drawing (Figure 4). The following advantages of SVG make it very useful for creating the usability-based model for our simulation: SVG is semantically rich, which means that the graphical information is always associated with meaningful textual information; It allows representation of both graphical and non-graphical data in an open, published XML schema. Thus a rectangle represents not only a shape, but also the meaning, which can

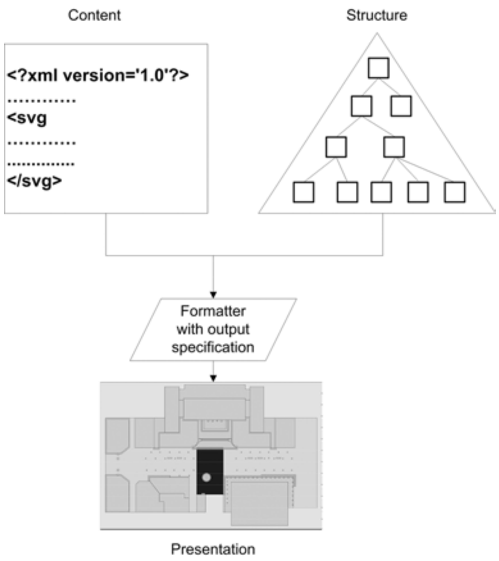


Figure 4
SVG components of the plaza model.

be a door, a window, or a wall. By using SVG, blueprints, drawings, and models are presented in a more meaningful context. Also, as a subset of XML, SVG's element tree data structure can be traversed to search for a specific node, which can be checked for attributes such as usability.

The DXF model of the plaza is a text file that has all the information about the geometries and their associated layer names. A partial DXF textual content is shown in Table 2. Our modeling tool can automatically parse the text file and convert the layer names and geometry information such as locations and coordinates into SVG elements. As the geometries in the DXF format are converted into SVG elements, their architectural semantics are inserted into the SVG element-tree according to their layer names. To insert the architectural semantics, we extended the SVG by adding architectural namespace – "arch". Custom tags of the architectural components with their attributes appear in the data structure. For example, the fountain-side element, where people can sit, is defined by its element type (LowWall), followed by its geometry element (circle) (Table 3). As another example, the fountain (water) is defined by its element type (Fountain), followed by its geometry element (circle) (Table 3).

The SVG's graphical view is shown in Figure 5.

This conversion is an intermediate step of the usability modeling. Based on this SVG model, we can create a discrete spatial model, where the space is discretized into cells and each cell is assigned a design element type and usability properties.

Discrete spatial model

Most of the activities in our simulation take place in the plaza's center area—the target region. Virtual users must know where they are and what they can do at any given location. For example, a user by the fountain can sit on a small area of the low wall. We subdivide this large space object into many small objects, in the form of a grid of cells. Each cell in the grid is an object that possesses several layers of properties. They can include information such as

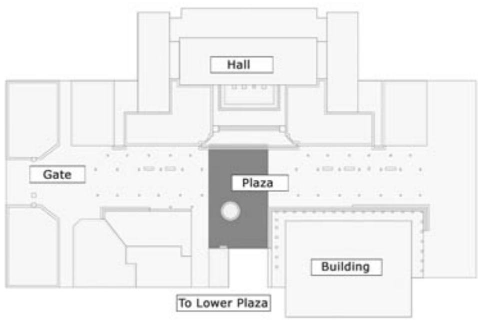


Figure 5
Graphical view of the SVG model

```
…………
CIRCLE
…………
LowWall-FountainSide
100
AcDbCircle
```

Table 2
DXF code for LowWall-FountainSide

```
<arch:LowWall id='FountainSide'/>
<circle cx='85.5' cy='90.75' r='3.75' style='&DesignElementStyle;'/>
…………..
<arch:Fountain id='Fountain1'/>
<circle cx='85.5' cy='90.75' r='3.0' style='&DesignElementStyle;'/>
```

Table 3
SVG code for the fountain-side and the fountain (water).

whether the cell is sittable, whether it is in the sun or in the shade, whether it is occupied by a user, etc. Every user occupies one grid cell at any given time. Given the potentially large number of cells in a plaza, we balanced computational efficiency against the physical reality by using 750mm sized cells, and 1 second time steps. The application of discrete space and time in our virtual user simulation has proven that we can accurately describe the system using that cell size and that time step.

We split the target region into cells and classified the cells according to their underlying design elements. Initially, each cell can have a number of design element types. For example, it can be a fountain-side and a ground, at the same time. The ambiguity is caused by the original CAD model, in which geometries overlap. We applied Painter's algorithm (Hearn and Baker, 1997) to solve the ambiguity. The algorithm scans the cells from left to right and top to bottom and assigns each cell's type using its top-most underlying level's design element according to geometries' z-buffers. For example, a cell's underlying levels can include overlapping elements: fountain side (a circle), fountain water (a circle), and ground (a rectangle). The cell is assigned to fountain water, which is the top-most level's type. In another example, if a cell's underlying levels include fountain side and ground, it will be assigned to fountain side, also the top-most level's type. That way we correctly marked all the cells, e.g. fountain side, fountain water, benches, steps, and ground (see Figure 6).

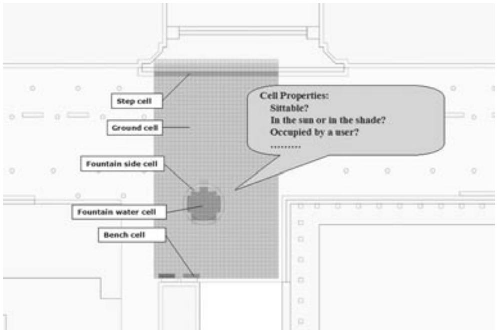


Figure 6
Discrete space model consisting of cells. A cell possesses several layers of usability properties.

Each cell's usability property is then added to the element's attribute list through the following way. From the textual view we see that the cells use predefined reusable modules, such as cell_0, cell_6, and cell_7. All the fountain-side cells use cell_6 as their module and have the same geometry and appearance, but different translation (location). All fountain-side cells hold the archID of "FountainSide". They are assigned properties of "usability=sit, stand" dynamically through usability assignment code when we run the simulation. In addition, the seating direction of each cell of the fountain-side is calculated using the location of the cell and the center of the fountain – each cell's seating direction is facing outwards from the center of the fountain. That way, a virtual user can sit on the fountain-side facing a reasonable direction.

This discrete space model becomes our usability-based model, which allows virtual users to access and understand the space's properties, beyond mere geometry. The whole process is shown in Figure 7.

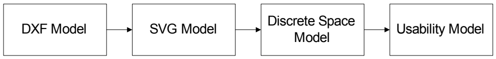


Figure 7.
Process of creating a usability model

Applying the usability-based building modeling to behavior simulation

We applied the building model to our behavior simulation in the run time through a simulation engine. The simulation engine first loads the building model and parses the model's graphical and usability properties, then creates a virtual user group—a list that allows an unlimited number of users to be added into, and upon completion of a journey removed from the list. The engine runs the simulation step by step, and at each time step (one second) it adds users from the entrances and moves all the users by one step. The engine passes the environment properties to the virtual users so that the users know, for example, where they can walk and where they can sit. Then the engine lets the virtual users move following behavior rules, e.g. shortest path, group movement rules, and social spaces (see Yan and Kalay, 2006 for details of virtual user modeling). The simulation engine uses Batik SVG toolkit with Java2D rendering engine, and Document Object Model (DOM) to traverse the element tree (see Watt, A. et al. 2003 for details of Batik SVG toolkit).

The simulation results include

- a 2D animation of virtual users movements, including walking and standing in the plaza, sitting at different places, and meeting other users, etc. (Figure 8 left); and
- a behavior data set that records all users' behavior information associated with their paths, including the coordinates along paths, arrival time, motions (walking, sitting, or standing), sitting directions, and duration of stay (Figure 8 right).

Finally, by inserting 3D models of virtual users to a 3D model of the plaza and letting the users move following the recorded behavior paths, we realized behavior visualization—animations in which the virtual users exhibit similar traits to those observed in reality (walking, sitting, meeting other virtual users, etc., see Figure 9).

Conclusion

Our building modeling supports design-oriented behavior simulation. It can easily include a wide range of design elements through a systematic approach. It is not tied to any particular CAD software. On the contrary, it is designed as a reusable tool for working with traditional geometric models and other building models. It utilizes CAD models and automates the conversion from CAD models to usability mod-

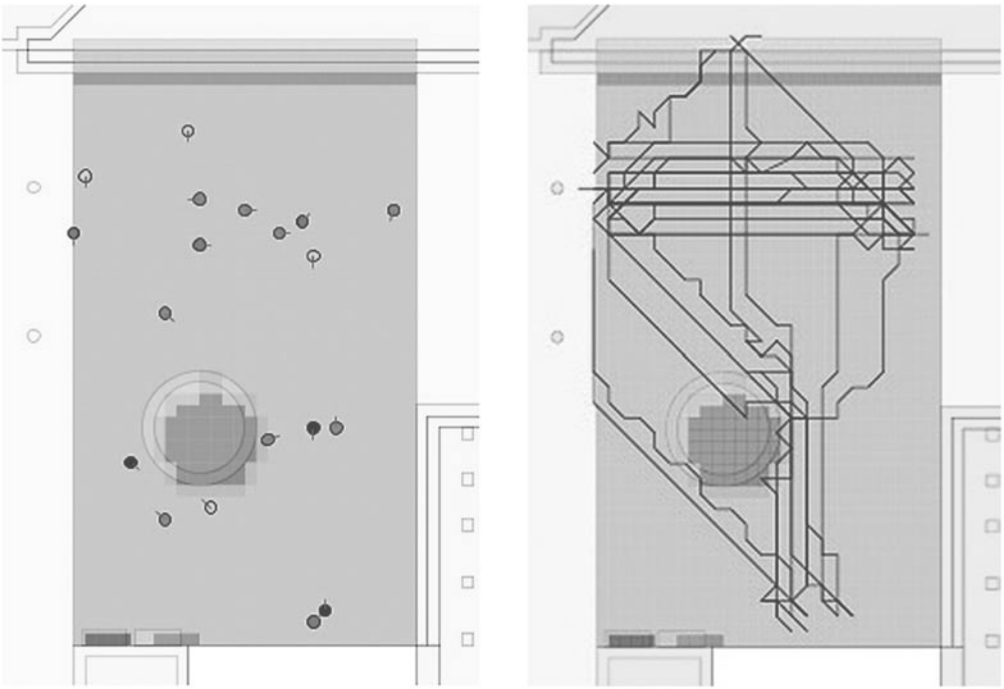
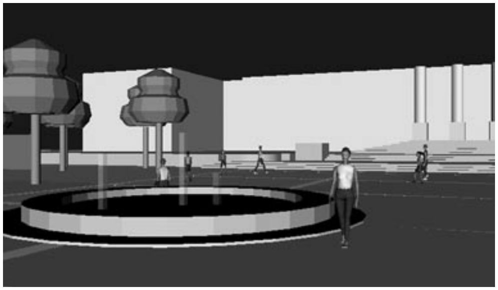


Figure 8
Left: 2D animation of virtual users' movements. Right: Virtual users' paths.

Figure 9
3D visualization of the behaviour simulation.



els and therefore requires minimal additional effort in preparing building data for behavior simulation, which has proven to be very efficient and effective.

References

Autodesk: 2002, Building Information Modeling. URL: www3.autodesk.com/adsk/files/2255344_BIM_WP_Rev5.pdf.

Batty, M.: 2001, Editorial: Agent-based pedestrian modeling, Environment and Planning B: Planning and Design 28(3), pp. 321-326. Pion. Ltd.

Eastman, C.M.: 1999, Building Product Models: Computer Environments Supporting Design and Construction. CRC Press, Boca Raton, FL.

Haklay, M., O'Sullivan, D., et al: 2001, So go downtown: Simulating Pedestrian Movement in Town Centres. Environment and Planning B: Planning and Design 28(3): pp. 343-359. Pion. Ltd.

Hearn, D. and Baker, M.P.: 1997, Computer Graphics, Prentice Hall, Inc. New Jersey.

Helbing, D., Schweitzer, F., et al.: 1997, Active Walker Model for the Formation of Human and Animal Trail System. Physical Review E 56, pp. 2527-2539.

Helbing, D., Molnar, P., et al: 2001, Self-organizing Pedestrian Movement. Environment and Planning B: Planning and Design 28(3), pp. 361-383. Pion. Ltd.

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

Kerridge, J., Hine, J., et al.: 2001, Agent-based modeling of pedestrian Movements: The Questions That Need to Be Asked. Environment and Planning B: Planning and Design 28(3), pp. 327-341. Pion. Ltd.

Khemlani, L.: 2003, Potentials and Challenges: Building Information Modeling Under the Microscope, WWW: http://www.aecnews.com/LKhemlaniBIM/06-LachmiBIM.html

O'Neill, M.J.: 1992, A Neural Network Simulation as a Computer-Aided Design Tool for Evaluating Building Legibility. Evaluating and Predicting Design Performance. Y.E. Kalay, ed., John Wiley & Sons, New York. pp. 347-366.

Ozel, F.: 1993, Computer Simulation of Behavior in Spaces. Environmental Simulation - Research and Policy Issues. D. Stokols and R. W. Marans. New York, Plenum Press. pp. 191-212.

Reynolds, C.W: 1987, Flocks, Herds, and Schools: A Distributed Behavioral Model, in Computer Graphics, 21(4) (SIGGRAPH '87 Conference Proceedings) pp. 25-34.

Tu, X.: 1996, Artificial Animals for Computer Animation: Biomechanics, Locomotion, Perception, and Behavior. Dissertation. University of Toronto.

Watt, A. et al: 2003, SVG Unleashed. Sams Publishing.

Yan, W. and Kalay, Y.E.: 2006, Geometric, Cognitive, and Behavioral Modeling of Environmental Users, in the Proceeding of the Second International Conference on Design Computing and Cognition (DCC'06), Netherlands.