CASE-BASED SIMULATION AS A TECHNIQUE FOR ASSISTING ARCHITECTURAL DESIGN

HONG-SHENG CHEN
Ling Tung College, Taichung City, TAIWAN
chsh@ntu.edu.tw

Abstract. In this research, I propose a case-based simulation method for experimenting with a set of models aimed at analyzing the behavior of a dynamic system. Owing to the assistance of CBR-Works4 software, this idea is successfully tested and implemented on the Internet. A set of evacuation simulations, which model complex human behavior in a building in certain emergency situations, leads to some discoveries and conclusions.

1. Introduction

Since Kolodner (1993) explored a profound and basic theory about case-based reasoning (CBR) methodology, many researchers have furthered the development of CBR in architectural design (Maher et al., 1995; Maher eds. 1997; Kolodner et al., 2002). Furthermore, some CBR systems have taken advantage of the Internet and made CBR more powerful than ever (Hayes et al., 1999). Recently, research devoted to the connections between CBR and architectural design in the teaching process has come to our notice (Arlati et al., 1996; Aygen, 1998; Chiu, 2001). However, few researchers have ever applied this methodology to simulations. Therefore, the possibility of this application is proposed as this paper's main topic.

There are several methodologies of "simulation" currently in use among academics. In particular, gaming, modeling, and virtual reality have been well-known simulation topics in recent years. Among these methodologies, system simulation is an important category, and it has been applied in other academic fields. Derived from system simulation, agent-based modeling (ABM) has become a new and important branch of traditional simulation (Kerridge et al., 2001). It takes every physical item as an agent, and models their reaction to the environment and their interaction with one other.

Although there are other kinds of simulation, we only focus on dealing with ABM in this preliminary research.

Why does a designer need case-based simulation? We may argue that the causal relationships between architectural space and human behavior are not understood completely, so that there is little basis for having confidence in the application of deductive simulation methodologies. On the other hand, prior successful modeling methods need to be retained and even reused when similar conditions prevail in future simulations. Designers and analyzers are in need of empirical studies that describe system behavior. Therefore, case-based reasoning can be used to develop a behavior model based on experimental data. In contrast to standard simulation or purely inductive methods, case-based simulation can help designers directly apply prototypical cases to a new situation.

2. Designer and Simulation

Nowadays, in the design profession, designers are required to focus on how to generate creative outcomes, which include buildings, furniture and other artificial products. At the preliminary stage of design, designers need not concentrate on their products' functional performance. Especially at a large design firm, there may be a professional evaluation department measuring product performance. Therefore, in business, design and evaluation can be separated. However, this approach is quite different from traditional design training. Renaissance artists and architects, for example, were usually good at the whole design process, including creation and construction. They did not only design, but also brought their design to reality. We might argue that a designer does not need to know much beyond what is in his mind, but it is unwise for a designer just to give design proposals without knowing their functional performance and the subsequent steps of evaluation and construction.

Simulation is a profound methodology of evaluation. Designers have known that they can take advantage of the computer to simulate their designs and ascertain whether their ideas are workable. Unfortunately, and usually, the evaluative simulation is not an easy job for a designer in the absence of knowledge about computer simulation. Designers often do not know the latest techniques for constructing useful models. The complicated mechanics of model modification is also a difficult problem for them. Consequently, many of them are unwilling to deal with evaluative simulation and leave it to others. However, the results of simulation are quite important for evaluation. We may discover whether things will go wrong in a design before its actual construction; and we may fix them or use other design alternatives.

Therefore, providing a friendly and easy technique for designers is an essential research goal that needs to be accomplished.

3. Purpose of Case-Based Simulation (CBS)

As we know, a simulation system always consists of many models, which work identically and interactively. In a chosen scenario, not all models need to be turned on and executed. We may just want some useful models to work with one other, and keep some unused models turned off. On the other hand, the attributes of active agents also need to be controlled in a model. Based on the particulars of different cases, we may need to determine several details of agents' attributes. However, composing different models does not always work successfully in a computer system, and it may fail to function well when we combine different models without knowing how they function relative to one another. Therefore, users usually need to modify such parameters and to verify these models in order to make sure they are workable.

Confronted with the above-mentioned difficulties, we need a system that can manage relevant programs in the program library and suggest which models are able to work together. This is why we want to take the advantage of case-based reasoning for simulation studies. It may reduce the time wasted on trial-and-error among inexperienced users, and let experienced users pay more attention to their simulation goals.

4. How Case-Based Simulation Works

This section explains the mechanism of the proposed case-based simulation system. Readers may ask: what is a case in CBS? In the man-made world, a case could be a building, an article, or even a computer program. In this study, we regard a program template as a case. A program template is a framework containing several models, which are written for different uses. Depending on different scenarios, we need to turn on or off some models. A case-based simulation system has to provide a program library for users to retrieve similar cases. If a case can be reused directly, the user may apply it directly. If not, the system needs to revise the case for proper application. After that, new solutions can be retained into the system. These are the well-known four R's (retrieve, reuse, revise and retain) in case-based reasoning. This study adopts these mechanisms for case-based simulation.

4.1. SOFTWARE

CBR-Works4 is a convenient tool for constructing the basic part of this case-based simulation. Produced by the German company, tec:inno, it possesses a powerful capacity for connecting users on the internet. That is to say, we may use web pages as interfaces and share information with the public. It is good at retrieving cases in a library and retaining new cases, but it is not very good at revising cases. Therefore, we need another technique that is helpful for assisting in revision. ASP (active server page) is one of the useful methods for us to overcome these shortcomings.

4.2. FUNCTIONS

This case-based simulation system is expected to provide some useful functions. First, users can search cases (programs) in this system for their specific application even if they are not familiar with programming. Second, through the Internet, users can share many programs with one another. It will assist a community with the same interests. The database will increase in accordance with users' contributions. Third, a generative program system will be very useful for simulation. We may generate a new program, which is capable of providing suitable attributes and logical structures for different circumstances.

4.3. PROGRAM TEMPLATES

After explaining the concept of case-based simulation, now we may shift our efforts from making a program case by case to creating a family of possible and useful programs. Therefore, how to control and manage a family of programs becomes a problem. Oren (1991) suggested the idea of "program templates", the meaning of which is that we want to use typical templates for managing many programs. Sorting them with functions and keywords, the system stores a skeleton library of programs. Such program frameworks are not "finished" yet when they are stored in the library. Only when they have been retrieved does the system assign certain attributes, constants, and variables. A framework is completed with these attributes and becomes a whole program ready for application. These concepts and uses of dynamic program templates are applied in the flowing sections.

5. An Example of CBS

In this section, an application example of case-based simulation is conducted for testing CBS concepts. I use the AutoLISP program running on a CAD simulation system to illustrate the ideas of this study.

As many researchers have mentioned, pedestrian behavior is too complicated to predict (Batty, 2001). In agent-based simulation, we always need several models to describe different interactions between agents. For example, we need different models for simulating agents' speed, direction, and destination, and we need other models for describing the interactions between agents, such as the pedestrian interference behavior-model. Some models control the relationship between pedestrians and the physical environment (e.g. the building: doors, paths and staircases). Therefore, model control can be quite troublesome when a designer wants to use a simulation system, because he has to focus too much on physical design. A good evaluative simulation should reduce a designer's work, and CBS is helpful in this regard.

5.1. MODELS

The following is an example that shows the contribution of CBS and its generation of suitable programs. This dynamic simulation system analyzes pedestrian movement in a floor space under emergency circumstances, and is composed of several models. (Usually, a program made up of several models is needed for describing a simulation situation.) Its purpose is to determine whether the pedestrian can evacuate in time or not. Now, we will be able to derive a suitable program directly form a case-based simulation system. The computer interface helps the user to decide which models need to be turned on or off, and to select proper attributes for agents. Figure 1 is a search by querying cases similar to identified attributes. The result of this search shows case16 and case24 have similarity values of 0.75 and 0.625 with respect to the identified attributes.

The system provides some choices regarding the basic models and special models. The attributes A, B, C, and D in Figure 1 stand for four models. R1 stands for the pedestrian's walking speed, and R2 for the firemen's. V1 and V2 stand for, respectively, speed of hazard expansion (combustible or flameproof) and speed of information transfer (via oral communication or a fire notification system).

5.1.1. Walk Velocity Model

Walk velocity describes how fast the agents can walk. If we need to describe more than one kind of walking capacity (such as in a hospital, where there are different kinds of people), the CBS system will automatically embed two models in the whole program and give two attributes for different velocities.

5.1.2. Fireman Model

If we need to model the fireman’s behaviors, we may ask the CBS system to adopt a fireman model. This model describes the fireman’s behavior, which is quite different from the escaper’s. A fireman agent will run toward the fire; it is different from escapers’ destination (doorways and staircases).

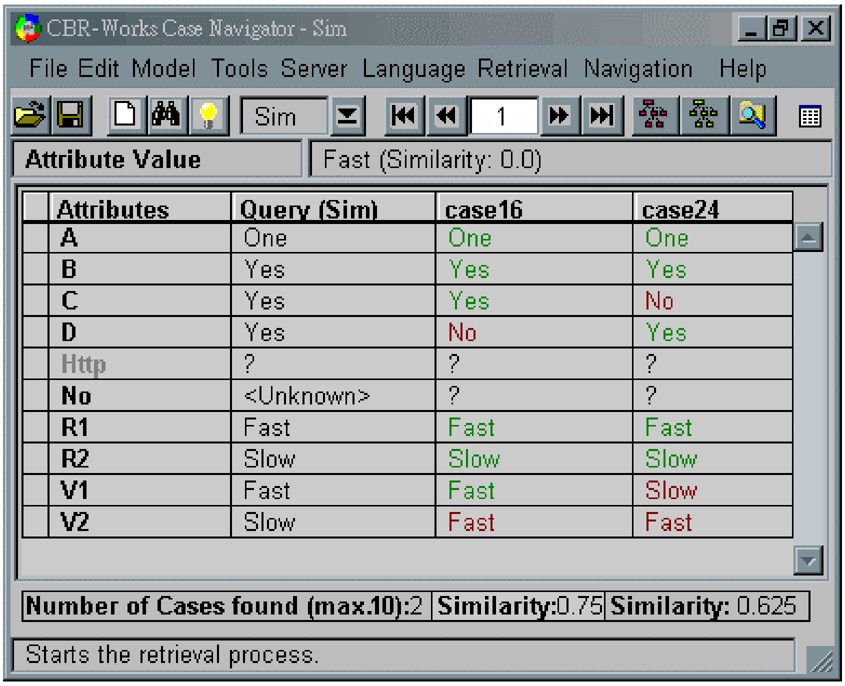

Figure 1. Software interface.

5.1.3. Expanding Fatal Disaster Model

A fire may cause death if the speed of the expanding of the disaster is quicker than pedestrians’ walk speed. The purpose of this model is to evaluate circumstances with serious fire and smoke. However, it is not always necessary to use this model, as we sometimes do not exactly know the expansion speed or we just want to see if there are over-crowded pathways. In short, this model may be turned off when we do not need it. On the other hand, the speed of fire expansion is related to building materials and the place where the fire takes place. We may use some attributes for setting these parameters and calculating the expanding speed of the disaster.

5.1.4. Special Behaviors Model

During emergency evacuations, people always have various and unexpected behaviors. For example, some of them may go back to the pathway that they came from, and some of them do not know where is the nearest staircase, so that they will not take the shortest path. Sometimes, we need to test such special behavior by dividing them in percentages. This CBS system provides some models for such scenarios involving special behavior. In addition, some models handle the interactions between two groups of agents, for example, the interactions between supervisors and customers in a store and between the nurses and patients in a hospital.

5.2. INTERFACES

CBR-Works4 uses a standardized exchange language: CQL. It provides communication between CBR-Works servers and clients as well as serving as an interface language between the CBR-Works components. CQL is an object-oriented language for storing and exchanging domain model descriptions and cases in the form of ASCII-Files. In addition, we may use CQL for transferring models and cases between CBR-Works servers and clients.

We may use web-design software to design a friendly interface. Figure 2 is an example that allows the user to set the model's parameters.

In accordance with the user's settings, CBS system will search for the most suitable model template in the case library. A template contains four or more models and ASP files, giving the user a further choice. The model template interface is shown in Figure 3. The user then can download programs (cbr1.lsp – cbr4.lsp) and run them directly, or he may use ASP files to revise some settings. For example, the layer names in AutoCAD can be revised, generating a program with proper attributes.

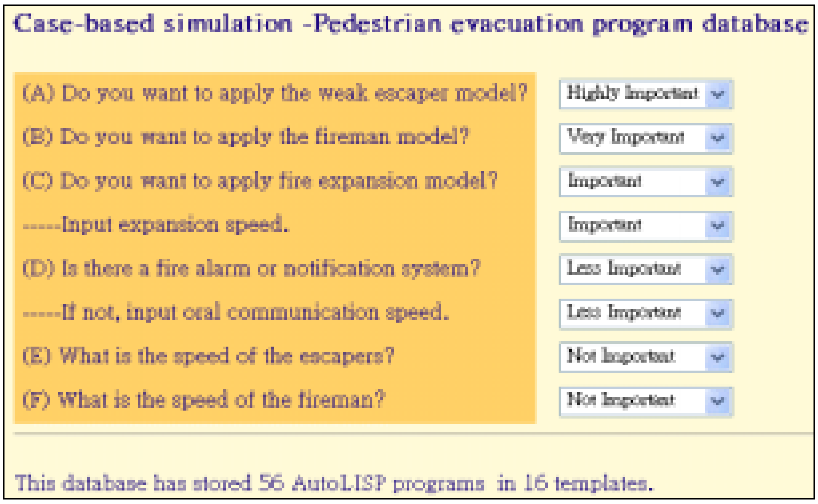

Figure 2. System interface.

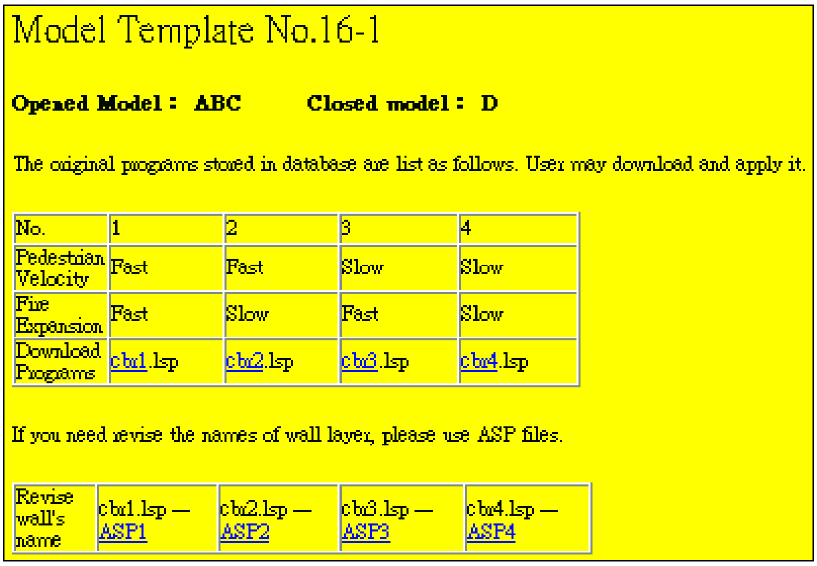

Figure 3. Model template.

6. Conclusion

After constructing this CBS system, I invited some designers to use it. They regarded this system as a potentially useful tool for assisting design evaluation, but there are still areas that need improvement. The case library is not sufficient yet, and we need more useful methods for modifying computer programs.

In conclusion, I outline the potential and benefit of case-based simulation as follows:

1. Designers are able to refer to previous cases and retain models to deal with new situations; in other words, they can avoid time spent on trial-and-error, and use similar models in simulations under similar circumstances.

2. Using CBR-Works4, the proposed case-based simulation method can be widely shared among researchers, who have a common interest in specific research topics.

3. Retaining old models expands the database, storing models for future use. In addition, users can retrieve models through Internet as needed.

4. Case-based simulation can provide assistance in the identification and modification of a relevant simulation case for a given design problem.

References

Arlati, E., Bottelli, V. and Fogh, C.: 1996, Applying CBR to the Teaching of Architectural Design, Education for Practice, 14th eCAADe Conference Proceedings, Lund (Sweden) 12-14 September 1996, pp. 41-50.

Aygen, Z. and Flemming, U.: 1998, Classification of Precedents - A Hybrid Approach to Indexing and Retrieving Design Cases in SEED (a Software Environment for the Early Phases of Building Design), CAADRIA '98, Proceedings of The Third Conference on Computer Aided Architectural Design Research in Asia, Osaka (Japan) 22-24 April 1998, pp. 435-444.

Batty, M.: 2001, Agent-based pedestrian modeling, Environment and Planning B: Planning and Design, vol.28, pp. 321-326.

Chiu, M.-L.: 2001, When and where is design situated in case-based design? CAADRIA 2001, Proceedings of the Sixth Conference on Computer Aided Architectural Design Research in Asia, Sydney 19-21 April 2001, pp. 3-13.

Hayes, C. and Cunningham, P.: 1999, Shaping a CBR View with XML, ICCBRí99, the 3 rd International Case-Based Reasoning Conference, Munich, Germany July, 1999, Proceedings by Springer Verlag.

Kerridge, J., Hine, J. & Wigan, M.:2001, Agent-based modelling of pedestrian movements: the question that need to be asked and answered, Environment and Planning B: Planning and Design. Vol. 28, No. 3, pp. 327-341.

Kolodner, J.L.: 1993, Case-Based Reasoning, Morgan Kaufmann, New York.

Kolodner, J. L., Gray, J. and Fasse, B.: 2002, Promoting Transfer through Case-Based Reasoning: Rituals and Practices in Learning by Design Classrooms. Cognitive Science Quarterly, Vol. 1.

Maher, M.L. and Pu, P. (eds.): 1997, Issues and Applications of Case-Based Reasoning in Design, Lawrence Erlbaum Associates, Hillsdale.

Maher, M.L., Balachandran, B. and Zhang, D.M.: 1995, Case-Based Reasoning in Design, Lawrence Erlbaum Associates.

Oren, T. I.: 1991, Dynamic templates and semantic rules for simulation advisors and certifies, fisgwick, P. and Modjeski ed. Knowledge-Based Simulation – Methodology and Application, New York, Spring-Verlag, pp. 53-76.