

EVALUATING PHYSICAL AND SPATIAL REQUIREMENTS OF APARTMENT UNIT FLOOR PLANS

Jin Won Choi¹ and Jae Wan Park²

Department of Housing and Interior Design, Yonsei University
134 Sinchon-dong, Seodaemun-gu, Seoul, 120-749, Korea
¹jchoi@yonsei.ac.kr and ²PELLI@yonsei.ac.kr

ABSTRACT

This paper demonstrates an experience in the development of a design performance evaluation system that can frequently evaluate building design performance within the design process in a real-time manner. The evaluation system, that we call "Vitruvius Studio," is composed of several modules such as a front-end component-based CAD engine, a knowledge base, and a set of design agents. While a computer programmer compiles a set of programming codes to check compiling errors during the implementation of a software system, an architectural designer can 'compile' his/her intermediate design product to evaluate design errors during the design process. The compilation can be done immediately at any level or any time during the design process in a real-time manner. We expect that this compiling process will dramatically increase design feedbacks, and thus result in a better design product.

INTRODUCTION

This paper demonstrates an experience in the development of a design performance evaluation system that can frequently evaluate building design performance within the design process in a real-time manner. The evaluation system, that we call "Vitruvius Studio," is composed of three main modules such as a front-end component-based CAD engine, a knowledge base, and a set of design agents. The notion of the Vitruvius Studio is quite similar to a compiler for computer programming such as a C compiler. While a computer programmer compiles a set of programming codes to check compiling errors during the implementation of a software system, an architectural designer can 'compile' his/her intermediate design product to evaluate design errors during the design process. The compilation can be done immediately at any level or any time during the design process. We expect that this compiling process will dramatically increase design feedbacks, and thus result in a better design product.

To evaluate a design, we first define "design faults" that can be categorized in many criteria levels such as components, relationships between components

including spatial relationships, or more high-level criteria (space depth, circulation, visibility, etc.).

An architectural design can be efficiently defined by a well-defined floor plan in which building components belonging in it and even spatial networks are well represented. To do this, we developed a front-end standalone CAD engine, called StrPLAN (Choi, 2001). The floor plan editing module is based on a robust building data model that includes hierarchical building components such as 'building', 'plan', 'space', 'ring', 'wall skeleton', 'surface', 'column', etc. The creation algorithm developed assures a semantically rich and structurally correct floor plan at any point in the design process. Particularly, the floor plan constructed through the design process contains spatial information as well as other design information about the building components. Thus, the system effectively manages spatial design information in the real-time basis. Since the system implemented on the basis of the algorithm is the very first step toward a complete intelligent CAD system, research and development issues to be considered next are identified at the end of the paper.

A knowledge base is composed of a set of design knowledge that can be used in the evaluation process to check design violation. For example, "Each room must have a door" can be a simple design knowledge. As this knowledge is computationally stored in an object-oriented database (so we call it a knowledge base), a design agent can evaluate whether each room has at least a door or not.

A set of design agents is also under development. This paper demonstrates some of them that are successfully developed so far. These design agents can check the following design faults: proper location of components, proportion and size of components (e.g. the size and proportion of a room, the size of window, etc.), belongings of components (e.g. the number of doors in a room), spatial adjacency, spatial (door and window) connectivity, the degree of spatial integration by calculating space depth, wayfinding and evacuation, and circulation.

System overview

As mentioned above, this system consist of three parts: front-end component-based CAD engine, a knowledge base, and a set of design agents.

The agent is a system that operates autonomously and rationally for its goal (Russell and Norvig, 1995). However, due to the proliferation of the interaction and networking of computers, agents communicate and collaborate with other agents (Huhns and Stephen, 1998). Thus, analysis agents operate on being intertwined with design knowledge in the knowledge base. Figure 1 illustrate the architecture of Vitruvius Studio.

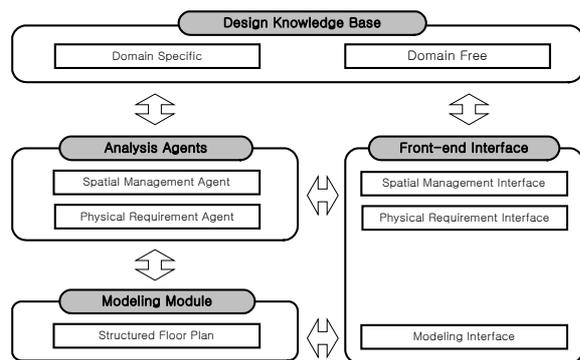


Figure 1 The architecture of Vitruvius Studio

CLASSIFYING THE ATTRIBUTES OF ANALYSIS AGENTS

According to the building type such as house, apartment, office building, and so on, analysis agents separately have specific database schemas as well as common database schemas that can be applied to all building types. Data sets called “domain specific” have different elements according to the building type. For example, an apartment unit floor plan consists of various spaces such as living room, bath room, kitchen, and so on and each space has a set of

design knowledge separately.

Table 2 shows some examples of database schemas applied to the design knowledge that can be used in the evaluation process to check design violation.

A DATA MODEL

Concept of structured floor plan

There has been some research on developing CAD systems that can build structured floor plans (Yessios, 1986b; Kalay et al., 1995; Choi, 1997). In such systems, a floor plan is structurally well defined by having some hierarchical components. Since space and form are the two main aspects in describing a building, they should be represented together in a system. To be more effective, this can take an object-oriented approach, where each building component is an object from the view of an object-oriented paradigm. That is, the object has its own data and methods of how to behave in certain situations.

A Building data model

Figure 2 illustrates a building data model developed through the study. Its structure is hierarchical and object-oriented, and it plays an important role in containing design information for each design project during the design process.

First of all, a project has a site and one or more buildings. Each building includes several plans that also have building components such as walls, columns, beams, slabs, and spaces. Here, important are walls and spaces. Each wall can have openings (doors and window) and two wall surfaces inside outside. A space generally has a single ring, but could have several rings as it contains holes inside. Several wall surfaces from a closed polygon representing a ring.

A data model for an apartment unit floor plan

Based on the building data model, a new data model can be developed differently in order to deal with

TABLE 1. Relationships of database schemas and Design Knowledge

ANALYSIS AGENTS	APPLIED ITEMS	EXAMPLES OF DESIGN KNOWLEDGE
Spatial Management	Circulation	Entrance should be by way of the living room.
	Connectivity	The Kitchen is placed between a dining room and a utility room.
Physical Requirement	Space - Utility – Door	The doorway of the utility room must be over 750mm wide.
	Space - Living Room - Visibility	View of kitchen equipment from the living room should be avoided.
	Space - Living Room - Area	The living room is the largest of all other Spaces.
	Space - Living Room - Area	An area of the living room occupies 33 percent of total area.
	Space - Bathroom - Window	Bathrooms without windows are acceptable.
	Space - Bedroom - Window	Best orientation of bedroom is the south.

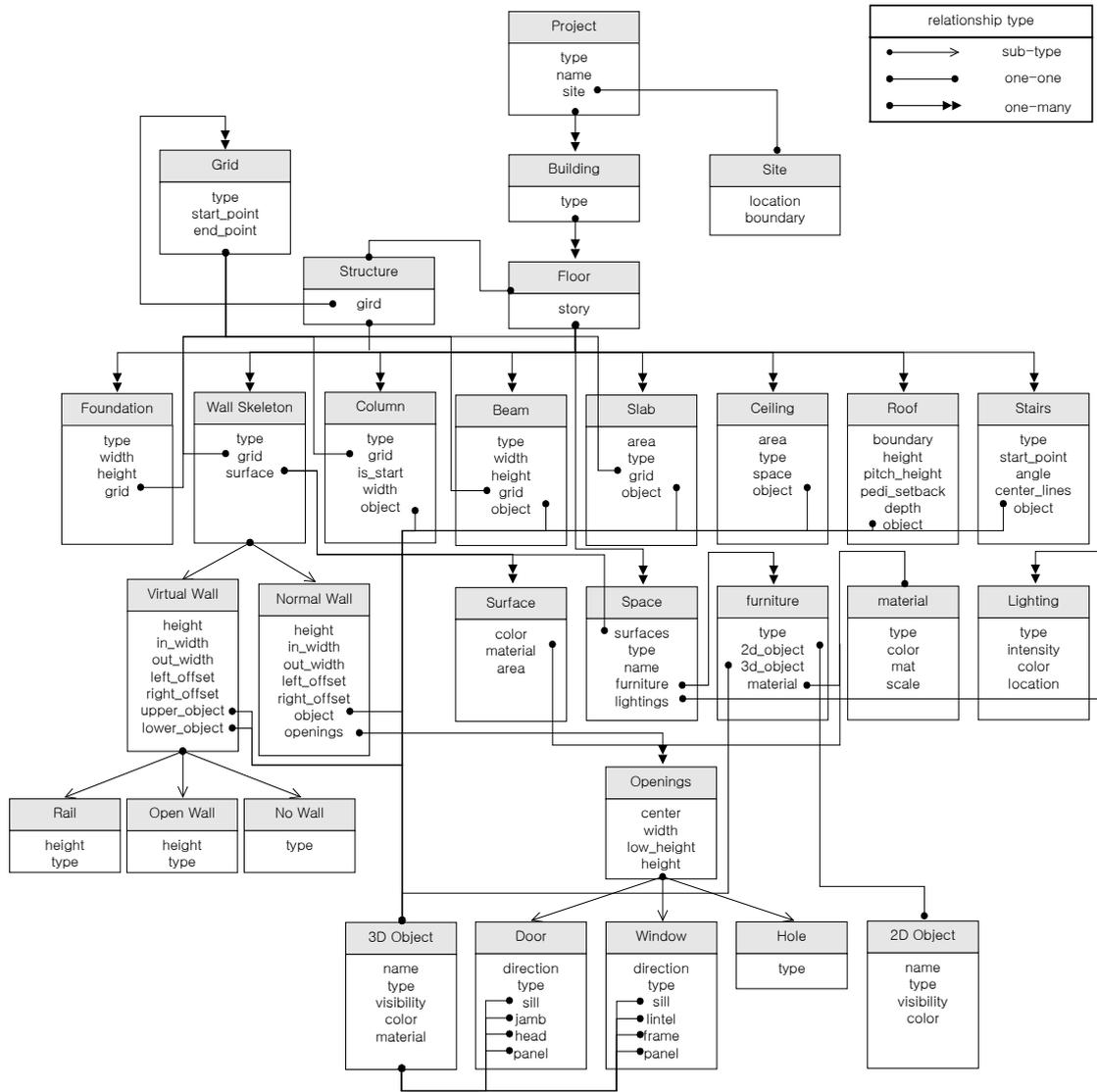


Figure 2 The building data model for the structured floor plan.

specific building types. In this system, it was developed for an apartment unit floor plan as shown in Figure 3. It consists of classes with physical hierarchy and attributes. A semantically-rich model, which can support multi-disciplinary design performance evaluation of buildings, must represent necessarily both space and structure (Khemlani et al., 1997). Thus, in classes with physical hierarchy, the building has been divided into two basic components: *plan* and *wall*. Also, a *space* class had been divided into *indoor* and *outdoor* having walls that enclose it (Carrara et al., 1994).

An apartment has public space such as elevator hall and stairs that dwellers employ together. Thus, An *indoor* class has been divided into *public* and *private*. Each component is then expanded further according to space types.

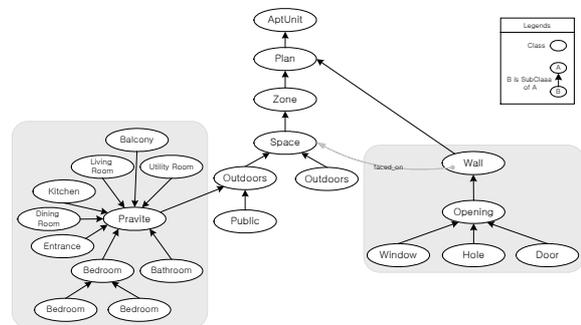


Figure 3 The data model for an apartment unit floor plan

THE ALGORITHMS FOR AGENTS

Spatial Management

A spatial management agent is used to evaluate spatial relationships such as connectivity, adjacency, circulation, space depth, and so on. It plays with several kinds of spatial networks consisting of the unit nodes of spaces and their linkages. The spatial network becomes a physical frame to describe the spatial information extracted from a structured floor plan immediately constructed by inputting geometrical data of a floor plan.

To extract the relational factors within a spatial network, the spatial management agent uses relationship matrices such as an adjacency matrix showing which spaces lie adjacent to each other and a connectivity matrix showing topological relationships between use-spaces directly connected to each other through the doorway. To evaluate spatial networks, the fault relational factors established by design knowledge are first stored. The relational factors extracted from the structured floor plan match with the relational fault factors to check design violation.

The circulation is represented as a graph with nodes for doorways and links between doorways. The node and links retain dimensional as well as locational information. After deriving the circulation, the shortest path from each room to all the other spaces is computed by the Dijkstra's Algorithm (E. W. Dijkstra, 1956). The shortest path can be employed to derive space depth that indicates how far the room is located apart from the other rooms within a floor plan. Figure 4 illustrates the algorithm for spatial management.

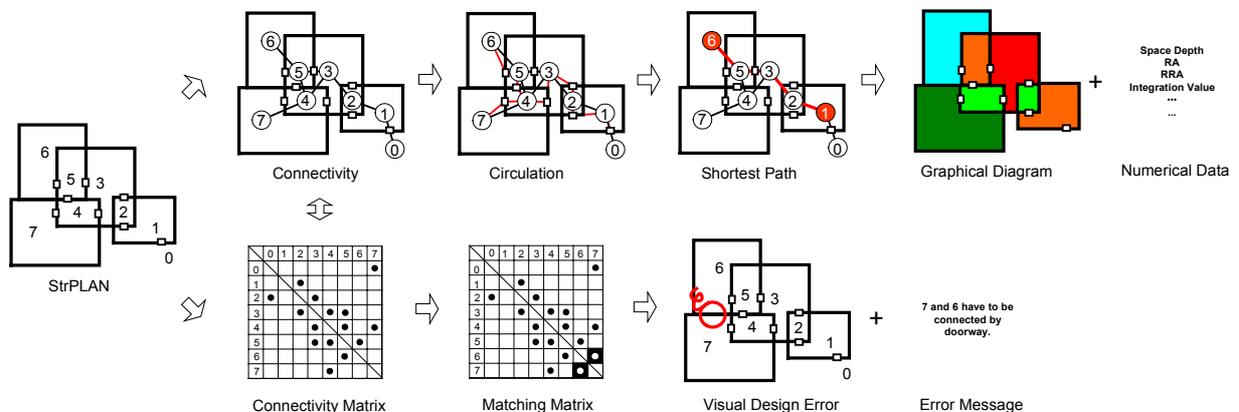


Figure 4 The algorithm for spatial management agent

Physical Requirements

To evaluate physical requirements of a floor plan, a set of design knowledge is stored in the knowledge base. The physical requirement agent checks design violation by comparing the attribute values of the physical hierarchy classes of a structured floor plan just by inputting data of a floor plan to attribute values stored in the knowledge base. The comparing between two attribute values can be achieved by means of If-Then rules.

IMPLEMENTATION

The Vitruvius Studio is an expert system along with a robust CAD engine that belongs to a broader category of programs known as knowledge-based systems. Expert systems must have flexibility required for easy editing of incorrect data and addition of new knowledge (Firebaugh, 1989). Thus, since each module is implemented separately, the whole system is flexibly operated.

TABLE 1. The Module of Vitruvius Studio

MODULE	DEVELOPING ENVIRONMENT	SYSTEM
Modeling: Structuring Data Object	Visual C++	Vitruvius Modeler
Indexing: Storing Design knowledge	MS SQL Sever 2000™	Vitruvius Knowledge Base
Evaluating: Evaluating Design	Visual C++	Vitruvius Agents

The modeling module of the Vitruvius Studio use Vitruvius Modeler developed with C++ by

Archmedia CAAD Research Group in the Department of Housing & Interior Design in Yonsei University. Vitruvius Modeler is a real-time floor-plan-structuring system. On only drawing a floor plan, it is automatically converted into well-structured floor plan. Thus, these spaces contain spatial information as well as other design information. Figure 3 illustrates an apartment floor plan designed.

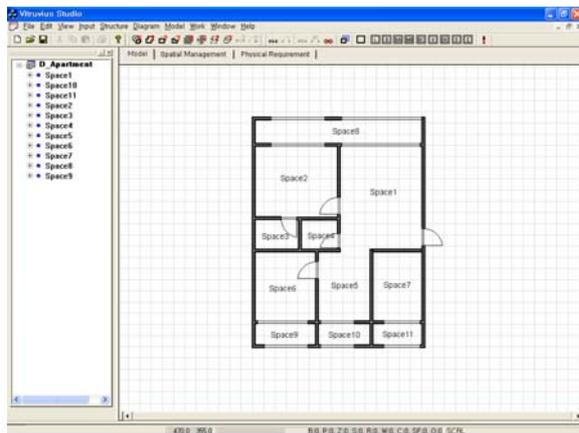


Figure 5 The modeling module designing an apartment floor plan

After drawing a structured floor plan, the user must define space types such as living room, master bedroom, bath room, and so on to each space for evaluating the domain specific design knowledge. If not defined, the system will just evaluate the floor plan with domain free design knowledge. Figure 6 illustrates the process of space definition.

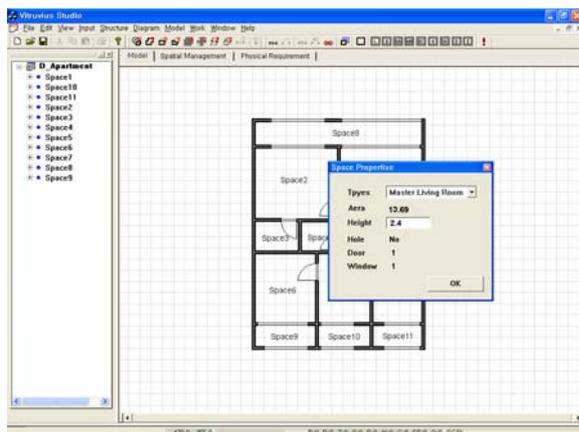


Figure 6 Defining space types in an apartment floor plan

If the user compile his/her design product to evaluate design error, Vitruvius Studio presents graphical check on the floor plan to visualize design errors, along with error messages stored in the database such as “The width of door is small” as shown in Figure 7.

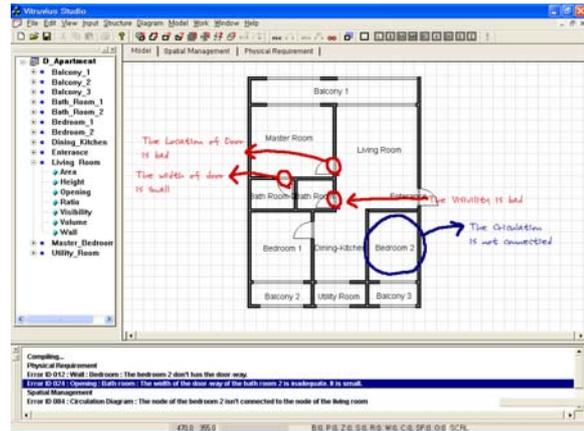


Figure 7 An apartment floor plan compiled

Analysis agents, called Vitruvius Agent, are composed of spatial management and physical requirement. The interfaces of these agents allow the user to view both the graphical diagrams and numeric data in an intuitive way.

Spatial management agents evaluate connectivity, adjacency, circulation, space depth and on so in a real time-manner. Space depth is terminology of “space syntax” that is defined by Hiller and Hanson (1984) as methodology for quantification and interpretation of spatial configuration in the dwelling. This agent employs the methodology of space syntax for evaluating design performance. Thus, space depth derived from connectivity is employed by calculating relative asymmetry(RA) and real relative asymmetry(RRA). Spatial integration value how easily the user can approach from each room to all rooms is measured by relative asymmetry and real relative asymmetry and is presented by the different colors for intuitive distinguishment. Figure 8 illustrates numeric data and graphical diagram.

Figure 9 illustrates physical requirement agent implemented to evaluate design errors. This agent consist of three modes; plan, wall and opening. The plan mode presents the area, the height and the volume of room and the visibility. The visibility is presented as the territories viewed through the opening in the room. It evaluates design knowledge such as “View of kitchen equipment from the living room should be avoided.”

The wall mode present the length, the height and the type of wall and the existence of the door and the opening mode present the size, the location, the direction and the type of the opening.

CONCLUSION

The Vitruvius Studio described in this paper is a system that integrates design and performance evaluation within the design process in a real-time manner. Generally building performance systems have been developed on focusing on building technology. This system, however, has been

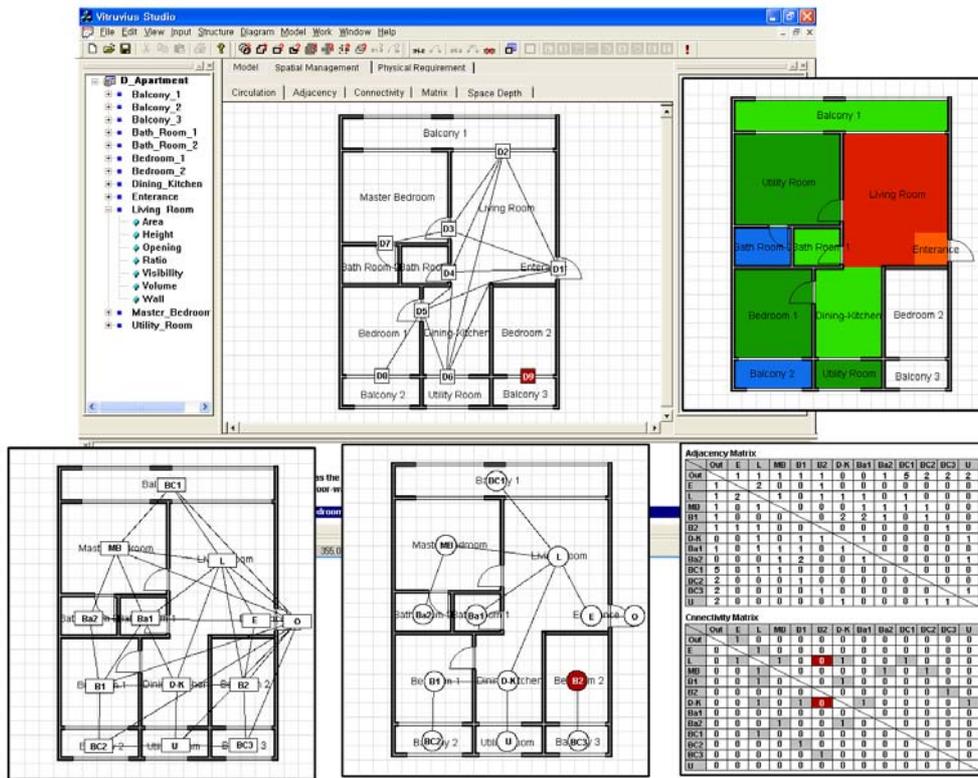


Figure 8 The feature of spatial management agent

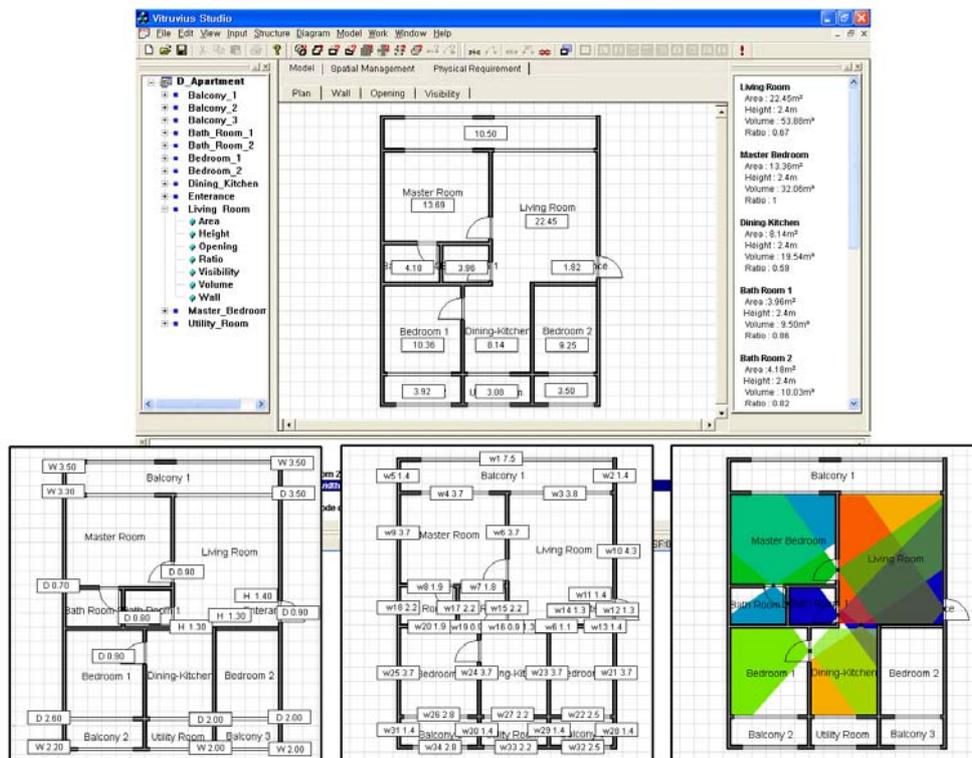


Figure 9 The feature of physical requirement agent

developed on focusing on the domain of architectural planning. We expect that this system especially will be employed usefully and practically in the early stage of design. Thus, the architectural designers can generate and evaluate immediately various design alternatives at the schematic phase during the design process by this system, and result in a better product.

The vitruvius modeler used in this system, has been developed by authors. A set of analysis agents, called vitruvius agent, is under development and at the same time, we are developing a script language to describe design knowledge exclusively and computationally.

The future development of the system will focus on tackling the following research issues:

1. Providing more powerful and convenient floor plan editing functionalities.
2. Extending functionalities to the building layout design.
3. Allowing to multi-story building representation by refining the building data model.
4. Developing various design agents as analysis modules for analyzing and simulating building performances.
5. Exploring better ways to represent and manage design knowledge.
6. Extending to a collaborative design environment that works on the Internet.

ACKNOWLEDGEMENT

This research was supported by the Yonsei University Research Fund at 2000.

REFERENCES

Carrara, G. et al, 1994, Knowledge-based computational support for architectural design, *Automation in Construction*, 3, pp. 157-175.

Chiara, D.J. and Callender J.H. (ed.), 1973, *Time-Saver Standards for Building Types*, McGRAW-HILL, New York.

Choi, J.W. et al, 2001, DesignBUF: Exploring and Extending 2D Boolean Set Operations with Multiple Modes in the Early Design Phase, *CAAD Futures 2001*, Eindhoven, pp. 589-602

Choi, J.W., 2001, Real-time Management of Spatial Information of Design: A Space-Based Floor Plan Representation of Buildings, Israel-Korea Binational Conference on Geometrical Modeling and Computer Graphics, Seoul, pp. 221-234.

Choi, J.W., 1997, A Development of An Intelligent CAD Engine to Support Architectural Design Collaboration, *Korea CAD/CAM Journal*, pp. 53-59

Dijkstra, E.W., 1959, A note on two problems in connexion with graphs, *Numerische mathematik*, 1, pp. 269-271.

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

Firebaugh M.W., 1989, *Artificial Intelligence: A Knowledge-Based Approach*, PWS-KENT, Boston.

Hillier, B., and Hanson, J., 1984, *The Social Logic of Space*, Cambridge University Press, London.

Hwang, J.E. and Choi, J.W., 2002, SpaceCore: Metadata for Retrieving Spatial Information in Architecture, *ACADIA 2002*, Pomona, pp. 199-217.

Huhns, M.N. and Stephen, L.S., 1998, Reading in Agents, in Huhns, in M.N. and Stephen, L.S.(ed.), *Agents and Multiagent Systems: Themes, Approaches, and Challenges*, Morgan Kaufman, San Francisco, pp.1-23.

Huhns, M.N. and Stephen, L.S., 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.

Snyder, J.C. and Catanese, A.J., 1979, *Introduction to Architecture*, McGRAW-HILL, New York.

Kalay Y.E., Khemlani L. and Choi, J.W., 1998, An Integrated Model to Support Distributed Collaborative Design of Buildings, *Automation in Construction*, 7, Elsevier, pp.177-188.

Khemlani L. and Kalay Y.E., 1997, An integrated computing environment for collaborative, multi-disciplinary building design, *CAAD Futures 1997*, Boston, pp. 389-416.

Kim, S., 1997, *Version Management in Computer-Aided Architectural Design*, Doctoral Dissertation, Harvard University, Cambridge, Massachusetts

Koile K., 1997, Design conversation with your computer: evaluating experiential qualities of physical form, *CAAD Futures 1997*, Boston, pp. 203-218.

Mahdavi, A., 1999, A comprehensive computational environment for performance based reasoning in building design and evaluation, *Automation in Construction* 8, pp.427-435.

Maher M.L. and Gero J. S., 2002, Agent Models of 3D Virtual Worlds, *ACADIA 2002*, Pomona, pp. 127-137.

- Russell, S. and Norvig, P., 1995, *Artificial Intelligence: A Modern Approach*, Prentice Hall, Englewood Cliffs, NJ.
- Yessios, C.I.: 1986a, *The Computability of Void Architectural Modeling*, Proceedings of Symposium on Computer-Aided Design at Suny Buffalo, New York.
- Yessios, C.I., 1986b, *Architectural Modeling*, Architecture 844 Lecture Notes I, Graduate Program in Computer-Aided Architectural Design Department of Architecture, The Ohio State University.