Form, Style and Function

A Constraint-Based Generative System for Apartment Façade Design

Ming-xian Lee[1] *and Ji-Hyun Lee*[2]
[1], [2]*Graduate School of Computational Design, NYUST*
[2] *http://jihyun.gcd.yuntech.edu.tw/*

This paper describes the development of a constraint-based generative system (FSF system) to support the design of middle and high-rise apartment façades from architectural plans. Floor plan and façade designs are heavily interrelated, and, sometimes, the plan constrains the façade design during the design process. This relationship lends itself to apply constraint-based systems and we have designed the system to connect intelligently between apartment plan and façade. In our system, we define constraints into three categories: structural form, architectural style and function. We use genetic algorithm to generate plausible alternatives quickly and augmented by a constraint-based system, façades can be generated and modified much more easily in terms of real-time visual feedback for checking violence of the constraints and of dealing with updates smoothly through intelligent connecting plans to façades.

Keywords: *Generative system; Plan-to-façade; Constraint-based system; Intelligent CAD; Style description.*

From Plan to façade

Architectural façades provide the first impression and image of a building. A floor plan and façade design are typically essential processes in designing an apartment building. The design process is iterative one checking the plan and façade back-and-forth until the architect has been satisfied with both the results. The floor plan and façade design, therefore, are heavily interrelated, and, sometimes, the plan constrains the façade design during the design process.

However, there are several difficulties associated with designing façades. For instance, there is the issue of the amount of time needed to generate the façade of an apartment building making out a draft of the floor plan. In generating a façade design, architects tend to use a large number of auxiliary lines to understand the relationship between plan and façade. Once a single floor of façade design has been finished, especially in an apartment building design, the completion of the single floor may involve repetitive redrawing of similar shapes in façade. Secondly, designers often wish to generate façade designs according to a certain style or as references to other styles. The conventional approach, however, is to spend time and effort in searching for cases or precedents. Thirdly, when parts of the plan are changed, the conventional approach is neither intuitive nor convenient. In order to deal with these problems, an intelligent CAD system to connect between apartment floor plan and façade is necessary. This interrelationship between floor plan and façade lends itself to examining constraint-based systems (CBS), which, recently, have become popular — for example, for plant room layout (Medjdoub et al., 2001), ceiling void layout (Bi and Medjdoub, 2004), building representation (Suter et al., 1999), etc. However, there has been little research into the plan-to-façade relationship and constraints.

Given this, we have designed a constraint-based generative system to connect intelligently between apartment plan and façade. We define a façade as comprising three constraints: structural form, architectural style, and function. Genetic algorithm is used to generate plausible alternatives quickly and augmented by a CBS, façade can be generated and modified much more easily in terms of real-time visual feedback for checking violence of the constraints and of dealing with updates smoothly through intelligent connection between apartment plans to façades.

The Design Process of Apartment façade in Taiwan

In the research of the apartment façade design process, we reviewed the apartment literature and interviewed practicing architects to gather information. When architects design an apartment building, they often start by deciding upon a main concept on which the design is based. Typically, taking the concept as a guide, the architect starts to decide upon a floor plan first and develops the façade accordingly. After transforming the plan information to the façade, the architect divides the façade into three parts: top, body and base. The architect starts to do the façade design part by part, modifies each part and iteratively checks the plan and the façade design to finalize them (Chien, 1998).

Two interviews were conducted to develop design process for apartment building. They both were practicing architects and had experiences to design apartment buildings in Taiwan. On the basis of the literature survey and interviews with practicing architects, we develop seven apartment building design procedures, each of which describes a specific development phase.

1. An architect decides upon a main concept on which the design is based or sometimes the main concept is given to the architect.
2. Taking the concept as a guide, the architect has finished the schematic layout of the floor plan.
3. On the basis of the plan, the architect draws many auxiliary lines to match the plan with the façade.
4. The architect divides the façade into top, body and base parts and has designed the façade part by part.
5. The architect checks and modifies the relationship between the plan and façade back-and-forth until the architect has been satisfied with both the results.
6. The architect repeats from the step 3 to step 5 to finish other sides of the façades.
7. The architect starts to make the construction drawing of the apartment building.

Figure 1 shows the relationship between the plan and façade in an apartment building.

Specifications of Constraint Sets in Apartment façades

Constraint-Based System (CBS) is a system which represents constraints explicitly and operates on constraints to solve the problem or to maintain consistency. CBS is used to deal with the problem with the constraints explicitly. Given constraints and preferences, the system will try to solve the problem using Constraint Satisfaction Problem (CSP) solving techniques or over-constraint satisfaction problem solving techniques. The extension to deal with more complex geometry will be based on Constraint Satisfaction Problem (CSP) approach (Bi, 2004).

We define a façade as comprising three constraints: structural form, architectural style and architectural function.

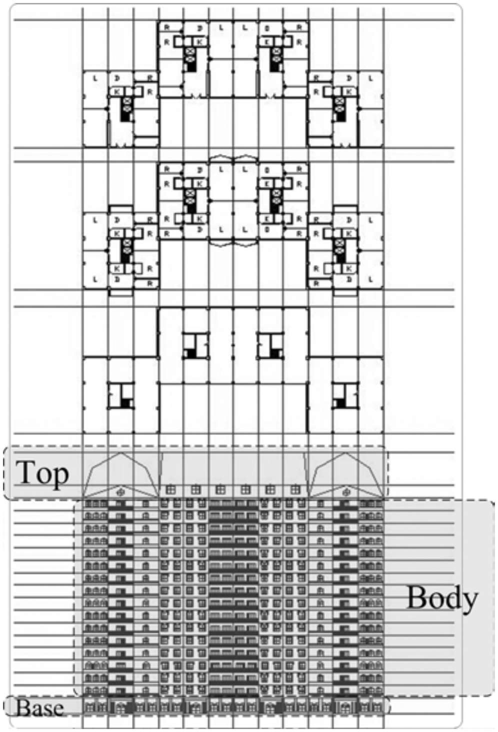


Figure 1
The conventional relationship between the plan and façade in an apartment building.

Structural form constraints

The size of an opening in a wall or roof plane can be determined by the material and construction of the wall or roof plane (Ching, 1979). The façade elements such as windows and doors, therefore, can be limited by the structural form.

The elements of the structural form constraints in an apartment building are the width, height and the position of the each object. Those elements can be calculated by Cartesian coordinate system. Figure 2 shows an example of the violation of the horizontal coordinates of window overlapping the horizontal coordinates of structure.

Architectural style constraints

Style is the constant forms and qualities that can be identified implicitly and explicitly in a group of objects. If an object has enough form elements characterizing a certain style, this object may be considered to belong to this style. Style is represented by a set of elements composed of sub-elements and their relationships which are embedded into a knowledge structure. It represents as follows (Gero, 1999):

$$\text{Style}(N) = \{(\cup M), (\cup F)\} \qquad (1)$$

Here F denotes form elements, N is name of

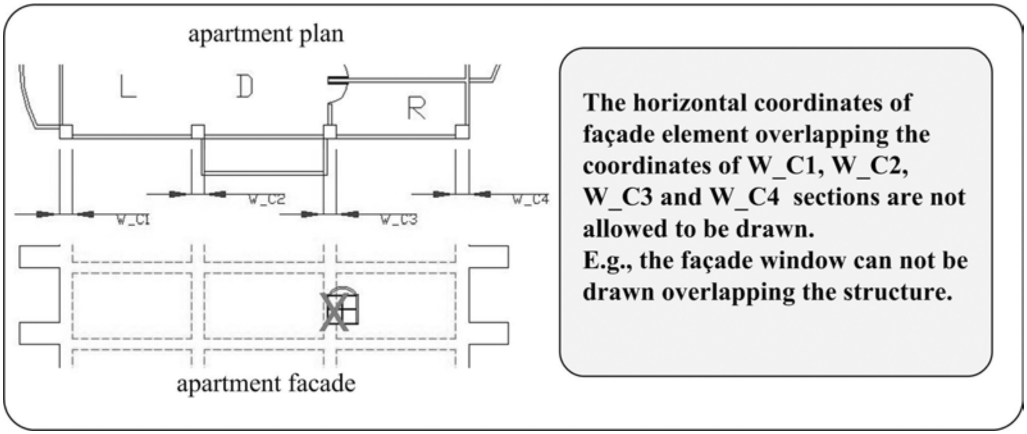


Figure 2
The structural form constraints.

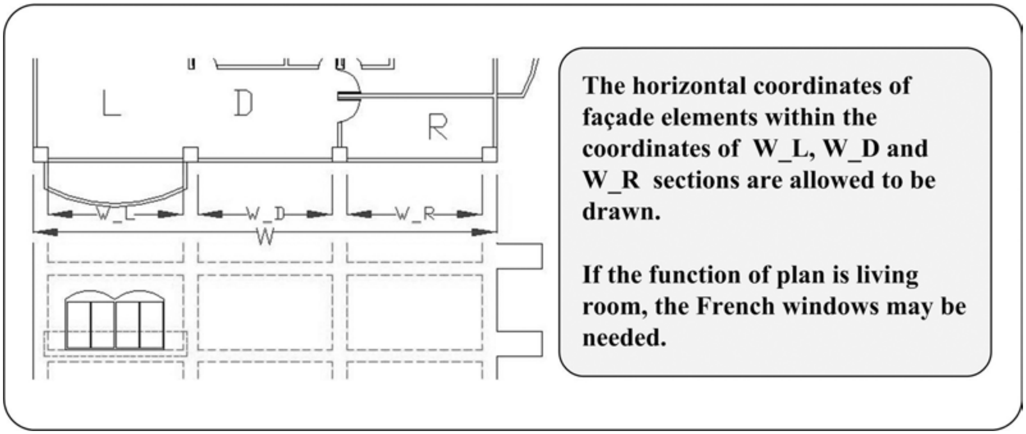


Figure 3
The architectural plan function constraints.

style, and M is members of style. For example: Style (Gothic) = {(Paris Cathedral, Laon Cathedral, Nayon Cathedral, Reims Cathedral), (Pointed Arches, Flying buttresses, Pyramidal roofs, Cruciforms, Pinnacles, Ribbed vaults, Stained glass windows, etc)}.

Architectural style touches upon many areas which include structure, behavior, function, society, culture, history and so on. Every style contains a common particular meaning; its semantics has a unique label. This label refers to its common complex semantics, e.g. Classical, Gothic, High Tech, Post Modern, and Chinese traditional and so on (Gero, 1997).

"European styles", "Japanese styles", "Chinese styles" and "Modern style" are the most popular apartment facade style in Taiwan. In addition, architects sometimes adopted the "Multi-Style", which is combined existed style to a new one. If two or three styles are combined, the feeling of visualization is quite different. An example of the Multi-Style is combining the "Modern style" with other styles[1]. According to equation (1), we can represent the architectural styles as follows:

Style(Modern) = {(Le Corbusier apartment in Marseille, Walter Gropius Interbau apartment house, Ludwig Mies van der Rohe's the lake shore drive apartments, etc), (concrete wall, big area glass window, horizontal roof, Building module, etc)}

Style (Europe) = {(Paris apartment, Netherlands apartment, Switzerland apartments, etc), (stone wall, arch window, dormer, Mansard roof, Bargue roof, iron railing, etc)}

Architectural function constraints

There are two kinds of constraint of architectural design function: constraints of architectural plan and constraints of architectural façade. Ching (1979) mentioned about *constraints of architectural plan function* that openings between the enclosing planes of a space isolate the planes visually and articulate their individuality. As these openings increase in number and size, the space loses its sense of enclosure, because more diffuse, and begins to merge with adjacent spaces. It illustrates that there are heavy relationship between the plan functions and the façade elements. According to the building plan function, the façade elements can be different. For instance, if the function of plan is living room, the French windows can be considered as the façade elements (Figure 3).

Constraints of architectural façade function decide two conditions: (1) the orders and layouts of façade elements (e.g., the layer of balcony is top of the layer of window); (2) the requirements of the façade element at different direction (e.g., the west side windows of

1 www.libertytimes.com.tw/2005/new/jul/17/today-e4.htm

the apartment may be needed overhang).

Genetic Algorithms as a Generative Tool

Genetic algorithms (GA) were invented by John Holland in the 1960s and were developed by Holland and his students and colleagues at the University of Michigan in the 1960s and the 1970s. Since then, GAs has given rise to many new applications in a variety of disciplines. Most methods called GAs have at least the following elements in common: populations of chromosomes, selection according to fitness, crossover to produce new offspring, and random mutation of new offspring (Mitchell, 1996).
When GAs used in the architecture design domain, it not only can find optimization solutions of building facades, but also generate evolutionary architecture forms. Very large numbers of evolutionary steps can be generated in a short space of time and the emergent forms are often unexpected (Frazer, 1995). GAs are particularly suited to be used in an architectural design context because their output is not a single solution but a number of high-performance solutions, one or more of which the architect can further develop by considering other criteria not included in the search process (Caldas, 2001). Therefore, we select the GA as the mechanism of generating the apartment building façades.

Genetic representation and Initialization

We encode all syntax rules (i.e., form, style and function constraints) as initial genes. The genotype of the initial individuals contains top, body and base parts. Figure 4 shows the genotype of an individual.

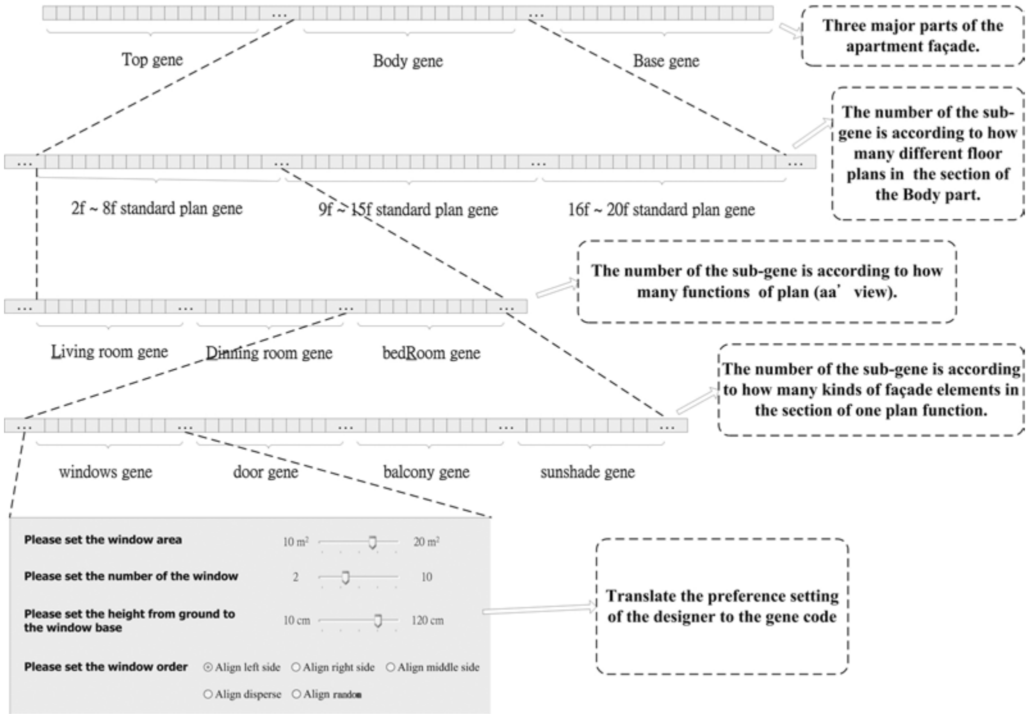


Figure 4
The genotype, sub-genotypes and setting of the initial individuals

Fitness function

The fitness function is a measuring mechanism that is used to evaluate the status of a solution (Man et al, 1999). We make an interface to let designers to check the fitness function using descriptions (Figure 5).

Production

Through crossover and mutation, it can generate second generation population of solutions form those selected. In this paper, we only consider the crossover genetic operator. There are several common crossover techniques, such as one-point crossover, two-point crossover, uniform *crossover*, etc (Man et al, 1999). Figure 6 shows the example of two-point crossover.

We make our GA system not only to generate single style façade, but also to generate the multi-style façade. After the crossover of initial individuals, we can generate various appropriate solutions of apartment façade (Figure 7).

CBS as an Adaptive Mechanism

Architects sometimes want to modify some parts of the façade that was generated by GAs. The CBS can check the constraints and connect the plan and façade intelligently. For example, when the designer modify the position of the window, the CBS will check the structural form constraints first, check the plan function constraints as well as the façade function constraints and decide whether the modification is allowed or not. If the modification is allowed, the CBS will update the plan to adapt the new façade.

Implementation

We build a constraint-based system, Form, Style and Function (FSF) system, to illustrate the usability of the suggested cases. Figure 8 shows the system architecture of the first implementation. The FSF system combines generative system (GS) using GA with constraint-based system (CBS), which contains structural form, architectural style, and function con-

Figure 5
Choosing fitness

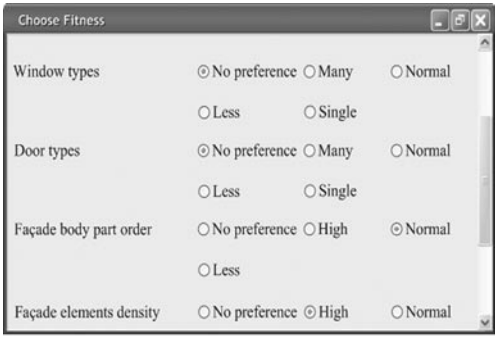


Figure 6
Example of two-point crossover.

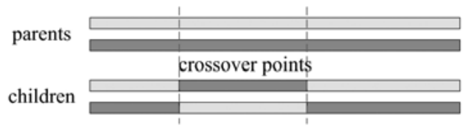
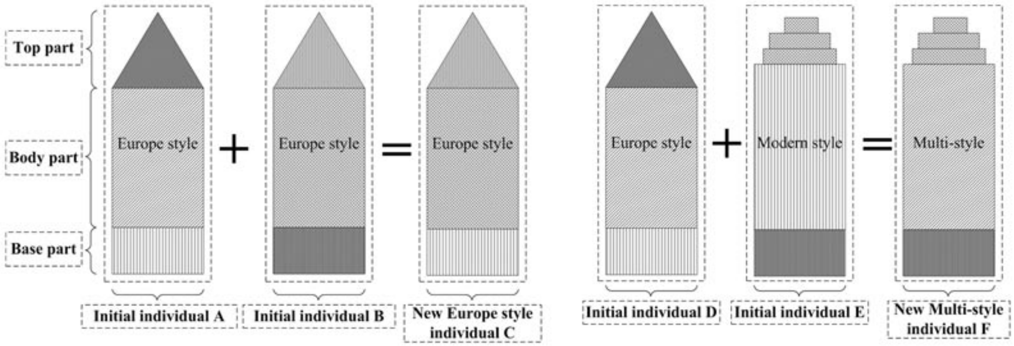


Figure 7
The crossover conceptual diagram of the FSF system



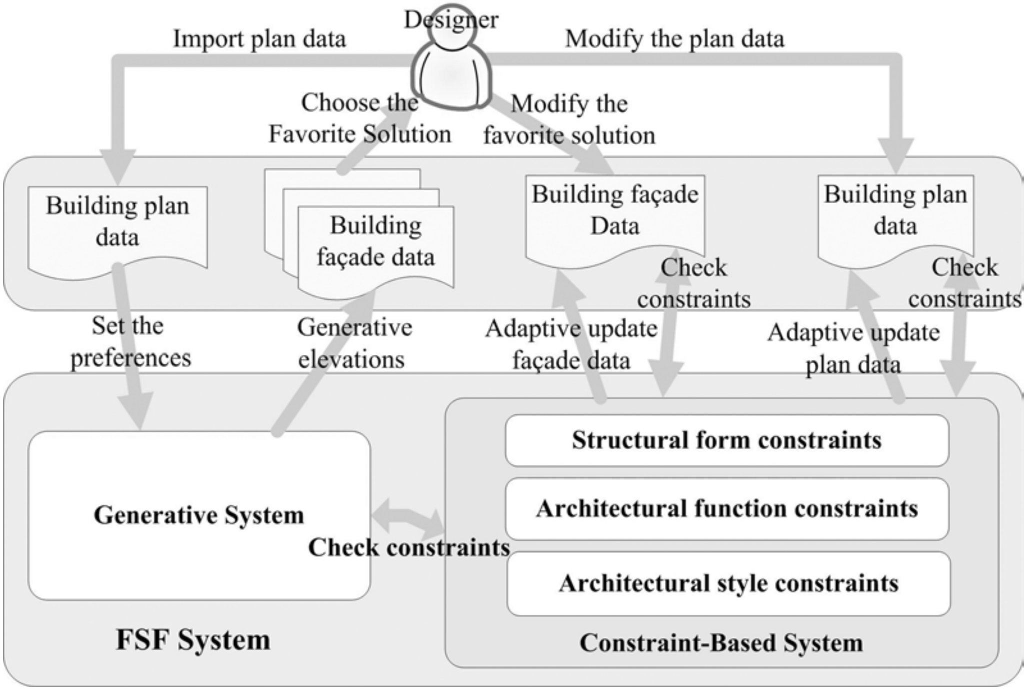


Figure 8
System architecture of the FSF system

straints.

Visual LISP applications are implemented on the top of AutoCAD as a GUI, to customize it, to seamlessly connect it to the GS and CBS, and to provide an efficient mechanism to communicate between the user and system.

When the user starts to operate the FSF system, he/she imports an apartment plan file and the FSF system provides the constraint sets of the corresponding façade. The user can select the form, style, and functional constraints as shown in Figure 9.

Then, the user can set the GA settings. For ex-

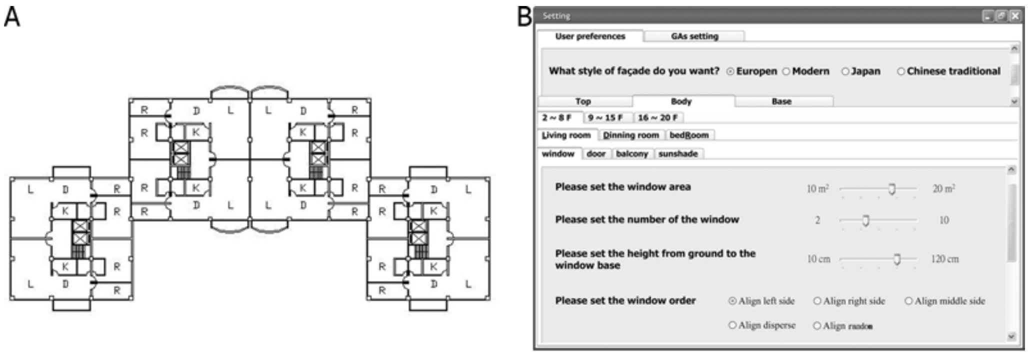


Figure 9
(A) Imported one apartment plans; (B). Interface snapshot of designer preference setting.

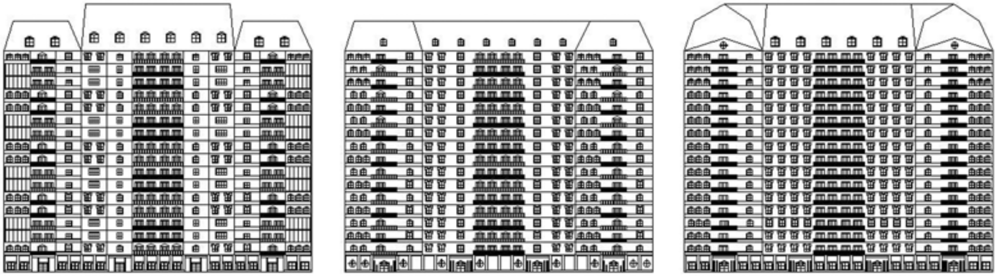


Figure 10
The preliminary solutions of the FSF system.

ample, the user defines 20 individuals of the initial population, the maximum number of generations as 15 and the probability of crossover as 0.67. Then, the FSF system generates the apartment façades as shown in Figure 10.

After the generation, the FSF system ranks the solutions according to the fitness function. The user can select one or more alternatives by his/her preference and modify them if needed. When the user modifies the façade, the FSF system checks the constraints promptly. If the modification is violated by the constraints, the system will show the warning message to the user (Figure 11). If the modification is allowed to be done, the CBS will update the plan to adapt the new façade. Figure 12 shows an example of the final results.

Conclusion

This paper describes the development of a constraint-based generative system (FSF system) to support the design of apartment façade from architectural plan. From the plan design, the system can generate alternatives of the façade very quickly. Using the benefits of the GA mechanism, the alternatives can be generated by user's preferred style. The selected façade can also be modified safely by checking constraints and be dealt with update smoothly through intelligently connecting plans to façades.

In the future, we want to use different types of buildings to exemplify our concept and examine the results. It would be also powerful if we can deal with the 3D modeling and rendering. Since we use AutoCAD software, our system has a potential to extend to the 3D environment.

References

Bi G. and Medjdoub B.: 2004, Hybrid Approach to Solve Space Planning Problems in Building Services, in Recent Advances in Design & Decision Support Systems in Architecture and Urban Planning (J.P. Van Leeuwen and H.J.P. Timmermans eds.), Dordrecht: Kluwer Academic Publishers, pp. 247-261.

Chien S.M.: 1998, An Operational Scheme for The Design of Building façade, Master Thesis, Graduate School of Engineering, National Taiwan University of Science and Technology, Taiwan.

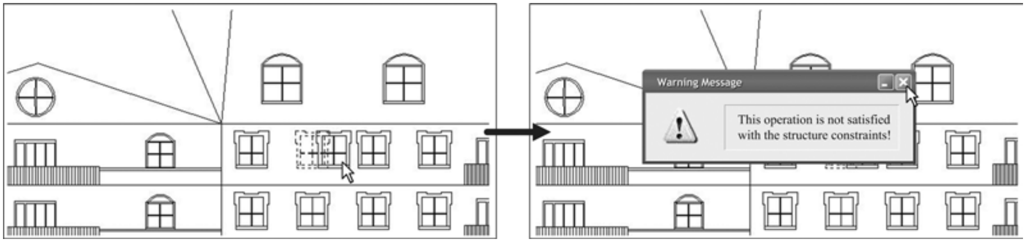


Figure 11
The modification and constraint checking of the FSF system.

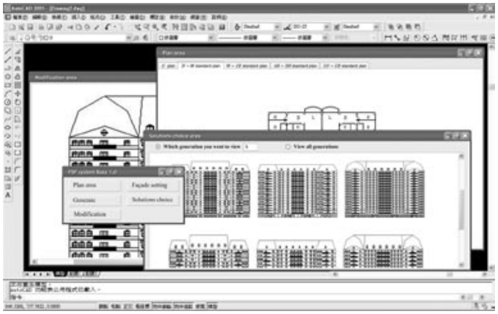


Figure 12
The system interface of the FSF system with the final results

Chen W.H.: 1999, A Computer-Aided Design System for Building façade, Master Thesis, Dept. of Architecture, Tamkang University, Taiwan.

Caldas L. and Norford L.K.: 2001, Architectural Constraints in a Generative Design System: Interpreting Energy Consumption Levels, Proceedings of the 7th International Building Performance Simulation Association (Rio de Janeiro. eds.), pp. 1397-1404.

Cha M.Y. and Gero J.: 1999, Style Learning: Inductive Generalisation of Architectural Shape Patterns, in Proceedings of 17th European Conference on Education in Computer Aided Architectural Design in Europe (A. Brown, M. Knight and P. Berridge eds.), September 15-17, Liverpool, UK, pp. 629-644.

Ching F.D.: 1979, Architecture: Form, Space & Order, Van Nostrand Reinhold, New York.

Frazer J.: 1995, Evolutionary Architecture, London: Architectural Association.

Gero J.S. and Ding L.: 1997, Exploring Style Emergence in Architectural Designs, in Proceedings of the 12th International Conference on Computer Aided Architectural Design Research in Asia (Liu, Y., Tsou, J. and Hou, J. eds.), April17-19, Taiwan, pp. 287-296.

Holland J.H.: 1975. Adaptation in Natural and Artificial Systems. Cambridge, MA, MIT Press.

Kilian A.: 2005, Design Innovation through Constraint Modeling, in Proceedings of 23nd European Conference on Education in Computer Aided Architectural Design in Europe (P. Duarte, G. Ducla-Soares and Z. Sampaio eds.), September 21-24, Lisbon, Portugal, pp. 671-678.

Li S.P., Frazer J.H. and Tang M.-X.: 2000, A Constraint Based Generative System for Floor Layouts, in Proceedings of the 15th International Conference on Computer Aided Architectural Design Research in Asia (Tan, B.K., Tan, M. and Wong, Y.-C. eds.), May 18-19, Singapore, pp. 441-450.

Medjdoub B., Richens P. and Barnard N.: 2001, Building Services Standard Solutions: Variational Generation of Plant Room Layouts, in Proceedings of the Ninth International Conference on Computer Aided Architectural Design Futures (B. Vries, J. Leeuwen and H. Achten eds.), July 8-11, Eindhoven, the Netherlands, pp. 479-493.

Mitchell M.: 1996, An introduction to genetic algorithms, MIT Press, Cambridge, MA.

Man K.F., Tang K.S., and Kwong S.: 1996, Genetic algorithms: concepts and applications, IEEE Trans, Industrial Electronics, vol.43, no. 5, pp. 519-534.

Suter G., Mahdavi A. and Kirshnamurti R.: 1999, A Performance-inspired Building Representation for Computational Design, in Proceedings of the 8th International Conference on Computer Aided Architectural Design Futures (G. Augenbroe and C. Eastman eds.), June 7-8, Atlanta, USA, pp. 117-132.