SPACE LAYOUT GAME: An Interactive Game of Space Layout for Teaching and Representing Design Knowledge

CHIEH-JEN LIN
Department of Interior Design, Tainan Woman's College of Arts & Technology, Yung-Kang City, Tainan, Taiwan.
t60011@ms.twcat.eu.tw

Abstract. This paper described a prototype of interactive space layout, a tool for primary stage of architectural space layout called Space Layout Game (SLG). Rather than focusing on the automation of space layout, we are interested in interactive responding in manipulative process and assisting teaching and representation of design skills and knowledge. Through composing a prototype of space layouts, reusing and modifying this prototype or other form teachers or other students to adapt to new conditions, students can present their design intensions, and learn layout skills and design knowledge through the manipulative processes.

1. Introduction

The space layout problem is a critical issue in education and practices of architectural design. Therefore many methods, algorithms, prototypes and systems have been discussed and developed since the first of CAD and AI developed in the 1970s. However, those fruitful results still seem to be left in a researching and developing state, and also never really enter most classrooms, or practical design studios. Discussions and developments about this issue became quieter more or less in the recent past. Although today we can complete very complex 2D plan drawing in AUTOCAD, create very fantastic 3D models and render extremely realistic visual simulation in formZ or 3DStudio. Most students and architects now spend much time in struggling with manual diagrams on paper when they allocate spaces in different design stages.

1.1. AUTOMATED APPROACHES OF SPACE LAYOUT PROBLEMS

Automation of design is one of the initial aspirations in developing CAAD and AI, and automated space layout is one of those first chosen tasks. There are many different approaches presented for solving this problem. Actually, almost every developing algorithm in AI, such as generative grammars, genetic algorithms, fuzzy logical, neural networks, etc. has already been applied to solving this problem.

However, the space layout problem, as most design problems, have two characteristics: (i) ill-defined and (ii) over-constrained.

Over-constrained problems have no single best solution and thus have too many possible solutions (Balachandran and Gero, 1987). This fact of too many possible solutions is a common disadvantage of most automated systems. Although many evaluating approaches are developed to filter out better or optimal results, most approaches are too complex for designers to understand or to connect with their experiences and knowledge.

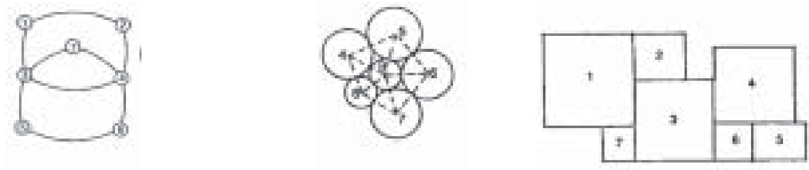
Furthermore, there would be a definite scope of solution if problem constraints were well-defined. But most design problems are ill-defined especially in early design state for beginners. To solve ill-defined problems is an interactive process between searching and redefining of problem constraints (Simon, 1973). Therefore problems constraints always vary with designer's intentions and temporary solutions. Nevertheless, another disadvantage of most automated systems is a complex, unnatural, inflexible and unidirectional generating process. When designers change intentions or want to modify problem constraints, they usually are scared by complex interface and afflicted by generating-testing process.

Therefore, although many achievements had been reached in the 1970s, very few automated systems have actually been applied in present education or practices of architectural design. One of the most notable and extensive research projects in recent years is SEED. The SEED system provided a module called SEED-Layout, which supports design solution exploration through an iterative, constraint-based approach that can be either manual or automated (Flemming and Chien, 1995). But tedious setting of constraints and annoying generating-testing process may still confuse beginners.

1.2. INTERACTIVE APPROACHES OF SPACE LAYOUT PROBLEMS

To avoid the problems of automated systems, a best alternative for assistant space layout system should be a responsive design process, which means a natural, intuitive, flexible, and most important, graphical and interactive process. Many methods have been proposed in graphical manner to help build up this kind of system from the 1970s. The virtual grid searching method (Mitchell) provided a simplified algorithm to find the relative location of spaces. The "Interactive Space Layout" (Ruch) approach proposed three-level schema of graphical representation: (i) network diagram, (ii) bubble diagram, and (iii) schematic plan to describe the problem of space layout (Figure 1).

However, these approaches do not really focus on the manipulative interactivity in design process because of constraints of early technology. These representations have become a basic and standard means of interpreting the space layout problem for educational and practical purpose.

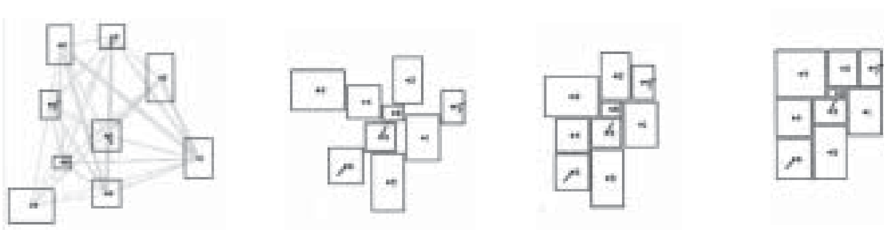


1.1. Network Diagram 1.2. Bubble Diagram 1.3. Schematic Plan

Figure 1. Three-level graphical representation of space layout problems (Ruch, 1978).

1.3. PHYSICALLY BASED APPROACH OF SPACE LAYOUT PROBLEMS

Although the "Interactive Space Layout" approach provided a schema of interactivity among spaces, the details were not very clear. A most interesting research about the interactive behaviours of spaces in recent years is the "physically based space planning" (Arvin and House, 2002). Rather than other automated methods applying design constraints to reduce or filter out solutions, this physically based method applied the law of physics to convert relation of spaces into various forces, such as applying elastic forces to stimulate spaces to dynamically move and change by those force (Figure 2).



2.1. Initial random position 2.2. Topological resolution 2.3, Geometric resolution 2.4. Alignment objective

Figure 2. Physically based space layout process (Arvin and House, 2002).

This physically based method provided a "living" design process by dynamic responses. However, this approach still attempted to automatically generate space layout from initial defined relation or given forces, thus the final state of layout actually had been determined by the initial state. Relation of spaces and design intentions usually varies with design process and temporary design results. How to provide a graphical and intuitive way to help designers to convert their intensions into appropriate physical forces, and to modify those forces to explore possible solutions would be a challenge.

2. Methodology of Space Layout Game

This paper describes a prototype of interactive space layout, called Space Layout Game (SLG). Through the survey of automated and interactive space layout methods, we attempted to integrate the fruitful results of graphical representation to help teaching design skills and knowledge of space layout. And we attempted to implement parts of physically based methods to provide a natural, intuitive, flexible, interactive and graphical interface to help designers to explore their design attentions.

2.1. DESIGN OBJECTIVES OF SPACE LAYOUT

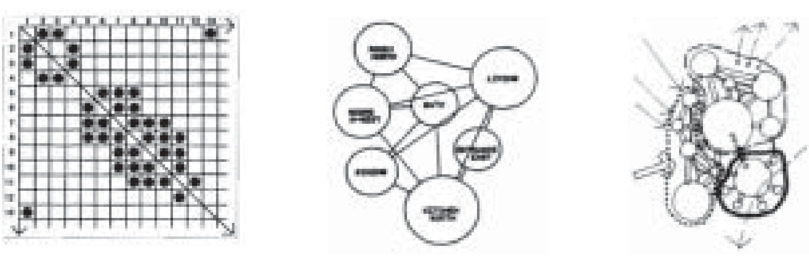
The design objectives of space layout can be concluded into two basic properties: topological and geometric (Arvin and House, 2002). Topological objectives reflect designers' intentions about relative positions and inner correlations of spaces. Geometric objectives reflect designers' intentions about the size and shape of spaces. Because our interests in the space layout problem are for educational purposes more than practical applications, we first modelled the most basic of topological and geometric objectives into SLG.

2.2. TOPOLOGICAL OBJECTIVES IN SLG

Although there are many different topological objectives necessary for fitting different design intentions, such as separation, orientation, interior and exterior, etc. (Arvin and House, 2002), the adjacency objective should be the most basic and important in primary state of space layout for educational and practical purpose. Therefore, we have modelled only the adjacency objectives into SLG for the moment.

2.2.1. Graphical Representations of Adjacency Relations

In traditional architectural design education and practice, space adjacency analysis is one of the most basic and important methods and processes for space planning. There are two basic representations for this analysis: (i) adjacency matrix, and (ii) bubble diagram (White, 1986). However, adjacency relation matrix usually is too complex for beginner to convert it into a topological or geometric diagram {Figure 3 (3.1, 3.2)}. Although a well–organized bubble diagram can present more information about topological or geometric intensions of design concept, a manual



3.1. Adjacency matrix 3.2. Random and ill-organized bubble diagram 3.3. Complex but will-organized bubble diagram

Figure 3. Graphical representations of space adjacency relations.

diagram cannot modify itself to fit the adjacency relation, and cannot remind designers where the conflicts are in the diagram. Actually, it usually needs many experiences and enough knowledge for a designer to draw a well-organized bubble diagram (Figure 3, 3.3). Therefore a bubble diagram usually is fit to present a concept by expert more than to study or explore it by beginners.

2.2.2. Boolean Adjacency Relations

There are also many different adjacency meanings and weighting techniques for interpreting the linked lines among bubble in bubble diagram. The most intuitive interpretation is the “Boolean” type adjacent relation, which means that two spaces should be adjacent directly with a linked line in bubble diagrams, but can be adjacency even without a linked line. This interpretation is easiest to understand by beginner and to catch conflicts by vision. And it is also very helpful for a teacher to explain and to teach guidelines or constraints of design acquirements. Therefore, we have now modelled only the Boolean adjacency objectives into SLG for teaching purpose.

2.2.3. Separation of Space as Rigid Body

Although we usually consider a space as a void area, and can also accept overlap condition of two spaces, spaces in SLG will be considered as a rigid body, like the metaphor of physically based method, and cannot be allowed to overlap.

2.3. GEOMETRIC OBJECTIVES IN SLG

Some approaches would ignore geometric problems to simplify the space layout problem. However, to solve the conflicts of topological objectives usually needs making many tests and decisions of geometric modifications such as changing shape, dimensions and relative positions. After simplifying manipulation of topological objectives, we also attempt to reduce decisions making and to encourage designers to test their intentions for learning purpose.

2.3.1. Simple Rectangle Space and its Design Information

There is only one shape of space in our prototype for the moment: a simple rectangle to present a space as most other systems and simplify shape and dimensional problems.

In primary design state, the dimensional and area information of space is more important than its exact location in coordinates system for designer. Many students would be perplexed by coordinates system in CAAD system such as AUTOCAD the first time. Therefore SLG will dynamically and immediately only provide dimensional and area information and will conceal the location information when a designer creates or modifies a space. Furthermore, SLG also uses a grid system to reduce dimensional and area decision-making.

2.3.2. Grid Snapping for Reducing Dimensional and Area Decision-Making

Although it may be very easy for an experienced designer to determine whether the dimensions and area of a given space are appropriate or not, these decisions are still difficult for beginners. Therefore, we applied an orthographic grid system in SLG to simplify this decision making. The orthographic grid system is just like graph papers familiar to students and the coordinate system of most CAAD software familiar to experienced designers. Any dimensional and location modification will be snapped on this grid system to reduce loads of both designers and SLG system.

SLG has a 90cm main grid and 30cm sub-grid system. The 30cm module, about one Taiwanese foot, is usually applied in education and practical convention and also conforms to the consideration of human factor engineer in many design methods, such as SAR method. Furthermore, the "Ping", an area unit equal to 36 square Taiwanese feet or about a square of 180cm x 180cm, is a popular area unit applied in Taiwan. Considering human factor engineer and local practical convention, we define SLG's grid system and provide dimensional and area information based on this grid system.

2.4. INTERACTIVITY IN SLG

2.4.1. Interactive Indication, Immediate Response and Conflict Reminding of Adjacency Relations

Because the SLG is established for a beginner or student to test and explore design intentions of space layout for educational purposes more than practical usage for now, we first focus on intuitive, graphical and interactive indication and modification of adjacency relations, rather than automated generating space layout.

For simplifying indication of adjacency relations, every space in SLG has a node at its centre to display its adjacency relations with other spaces. User can click it to link or unlink to other spaces, and the node and its linking lines connecting to other space's nodes also become a network diagram for visualizing space's adjacency relations.

Once a new adjacency relation has been indicated, SLG will begin to try to modify locations of relational spaces to satisfy their adjacency relations. Therefore, once the system or a user modifies a space's position, other adjacency spaces will be pushed away by the moving space to avoid overlapping. Therefore, these closing forces from the user or the separating forces from the system became two opposing powers to stimulate interactive behaviours of spaces in SLG.

However, we did not attempt to make SLG smart enough to find all appropriate positions to satisfy all adjacency relations. SLG will just attempt to move relational spaces to be closer and more adjacent, and will immediately and dynamically present this moving process to respond to user's adjacency intentions.

Because of SLG's simple algorithm, obviously there will appear more conflicts when more spaces and adjacency intentions are increased into SLG. Besides the immediate and dynamic response, one of SLG's advantages over manual diagram is that SLG will automatically remind designers of those conflicts and suggest what needs to be adjusted rather than to adjust them automatically for users.

2.4.2. Intuitive Manipulation of Geometric Modification

Although the most natural and intuitive design tool may still be a pencil and a paper, the modern computer GUI (graphic user interface) has developed over more than 10 years and the operations of GUI are be very familiar to today's students. When we considered a natural and intuitive manipulation of geometric modification, we immediately thought of the operational experiences of GUI. Therefore we applied basic operation of a "Window" in GUI system to model SLG's geometric modification, rather than the complex way of commercial CAAD systems such as AUTOCAD.

2.4.3. Interaction between Topological and Geometric Objectives

Because of the ill-defined characteristic of the space layout problem, the relation between the indications of topological objectives, as defining problem, and the modifications of geometric objective, as searching solutions, is also a mutually influencing process. Designers usually spend a total time solving the conflicts caused from this process. Although SLG leaves those conflicts to be solved by designers themselves rather than attempting to solve them automatically, through dynamic and immediate responses of results and conflicts rather than providing possible solutions, we built up SLG as an easy and interesting tool for teaching, questioning and exploring space layout problems.

3. Implementation of SLG

3.1. THE "PROCESSING" AND JAVA LANGUAGE

The first prototype of SLG was developed under the PROCESSING environment. PROCESSING is a java-like programming language and environment developed by MIT and built for visual art and design education based on JAVA, a popular object-oriented programming language. Through its simplified JAVA language and easily controlled interactive API in PROCSESS, we can quickly test our concepts about interactivity and graphical interface, but have no need to struggle with complex coding of GUI API in JAVA language.

PROCESSING could also produce Applet that can be included in web page and be executed in web browser to provide cross-platform and internet-ready ability as JAVA. Furthermore, the object-oriented concept and multi-threat ability of PROCESSING based on JAVA also provides us an entirely new approach to solve our problems.

3.2. INTERFACE OF SLG

Because of considering simple and intuitive manipulations for beginners, all manipulations in SLG will be controlled by the mouse cursor rather than complex keyboard input and dialogue windows. Necessary prompts and information will be attached with the mouse cursor or a space itself in SLG.

Spaces in SLG are displayed as a light gray rectangle with deeper gray boundary. For reminding the conflicts of topological or geometric objectives, SLG will change the colour of space from gray to green, yellow or red to present the modifying, overlapping and separating conditions of spaces. Furthermore, the adjacency relations of a space will be displayed as a simple network diagram when user moves the mouse cursor into it, and the lines of network will also change colour from green to red to prompt the separating conditions infringing the topological objectives (Figure 4).

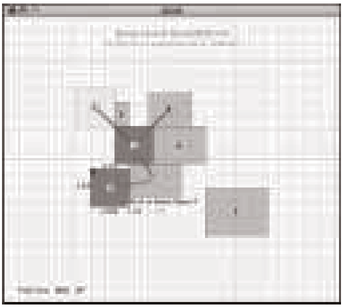


Figure 4. The interface of SLG.

3.3. MANIPULATIONS IN SLG

A user can click two diagonal corners on any void location to create a new space in SLG, and click on a space to move it. SLG will prompt dimensional and area information and topological networks of spaces when user moves the mouse cursor into a space or resizes it.

For modifying space, SLG will provide delete and resize icon on space when user moves the mouse cursor into a space to allow user to delete or resize it. Furthermore, SLG will also prompt a small node with space id at the centre of a space and display the connecting network diagram with other when user moves the cursor into a space or moves a space. Clicking on this node and connecting it with other space's node, user can easily indicate or cancel the adjacency objectives among spaces.

In conclusion, we attempt to keep all manipulations as simple, easy and intuitive as possible in SLG to help users to concentrate more on solving conflicts rather than complex drawing skills and system functions.

3.3. PRELIMINARY EVALUATION OF SLG

Although some attempts have been made in SLG, there are still some problems in our approach: (i) an editing machine for establishing a goal of SLG to make it more like a game, (ii) a score estimating machine for evaluating user's result, (iii) a simple and intuitive interface for multi-spaces topological and geometric objectives such as group or zoning, alignment and proportion.

One of the primary ideas in developing SLG is to expect it more like a game than a CAAD tool. Therefore there should be a goal for users to quest and a score-estimating machine to sense how far he is from the goal. However, a teacher can describe a goal in the design studio before students begin to play SLG, such as how many spaces and their properties, and leave score to evaluate by himself or students to discuss with each other.

Multi-spaces topological geometric objectives usually appear in more complex and advance design projects as a method to simplify complex design problems. Although it is easy to build up a hierarchical space system in our approach that allows a space to containt other spaces in order to constrain their dimensions and locations as a group or a zone. At present we are more interested in developing a simple tool for students to learn and explore primary concepts and methods for solving primary space layout problems such as a small house project than a complete but complex assistant environment. However, we will still try to integrate those issues into next or advanced version of SLG.

4. Discussion

4.1. AUTOMATED VS. INTERACTIVE CAAD TOOLS

Although CAAD tools have deeply influenced academic and professional fields of architecture and play an important role in those fields today, the primary goal to automatize design process still seems only a fantasy and faraway dream. Automated tools always need to model some design knowledge or process into systems in order to automatize them. However, the ill-defined characteristic of design problems results in that the knowledge in automated systems conflicts with designer's temporary beliefs, and thus the manipulations in automated systems become a complex, unnatural, inflexible and unidirectional generating process.

Designers especially beginner may not need automated tools but they do need an interactive environment to test, explore, learn and build up their own knowledge or beliefs. SLG provides a simple and interactive place for testing and exploring possible solutions under designer's intentions based on Boolean type adjacency objective rather than generating it automatically. Through the interactive manipulations in SLG, it should be a more natural, intuitive and interactive way of learning than using automated tools and commercial CAAD tools like AUTOCAD, and at least more interesting than manual diagram on papers.

4.2. TOPOLOGICAL VS. GEOMETRIC OBJECTIVE

Most approaches divide space layout problems into topological and geometric objectives in order to solve them individually and more efficiently. Nevertheless, the process to solve topological and geometric objectives is just like the process of searching for solutions and redefining problems in solving ill-defined problems. Therefore, to divide space layout problems will ignore the interactive influence between two objectives.

SLG is also an attempt to directly integrate both topological and geometric objectives into one simple graphical interface and to provide a dynamic representation of interactivity between topological and geometric objectives. This integration should be able to inspire designers to sense the interactivity between topological and geometric objectives and furthermore to discover possible methods to solve the conflicts between them.

4.3. PROTOTYPE VS. VARIATIONS

The concept of prototype and variation is also a very important approach in academic, educational and practical applications. The ordinary way to teach this method is to provide some similar cases for students to analyze, then to ask them to try to present what they had learned from those cases as a prototype. For testing this prototype extracted or abstracted from cases, students will usually be asked to transform this prototype into some variations to satisfy new constraints, such as different shape or context of site (Chiu, 1996).

However, the process of producing a prototype and its variations usually is a tedious and annoying experience for beginners and usually will constrain a student's idea. Furthermore, students usually forget or give up some important properties of their prototype for more easily satisfying new constraints. SLG provides a very easy method to build up and present a prototype based on adjacency relation of spaces in cases, and to record one of the variations with a schematic plan. By modifying this variation, students can still keep their prototype identity through the entire design process and have no need to worry about forgeting their prototype (Figure 5).

Although a prototype should contain more requisite knowledge appropriate to a specific design situation (Gero), the adjacency relations and one of its variations with schematic plan should still be one of the most important knowledge in a prototype. Therefore SLG also can be considered as a visual presentation of prototype and its variations.

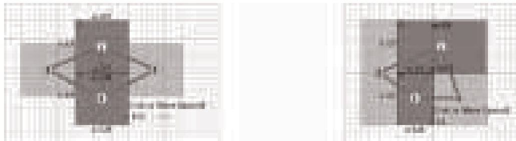


Figure 5. A Prototype of Space Layout (Left) and Its One Variation (Right) in SLG.

5. Conclusion

We do not attempt to provide entirely new methods or algorithms to solving the old issue of space layout problems. But through integrations of some past fruitful results and testing new programming techniques to improve manipulative process, SLG demonstrates a possible approach to improve old method with new technology.

Except the score evaluation device and the interface for multi-spaces objectives discussed before, one of the most important improvements that we wish to reach in future is to connect SLG to a database on Internet. Connecting to a database can expand the ability of SLG only from recording final results to recall the detailed process of user's manipulations, thus we can improve the educational usage of SLG. Furthermore, connecting a database over Internet can allow students to access and to share their works anywhere.

Furthermore, for applying the potential ability of SLG in presenting the graphical and prototypical knowledge of spatial layouts, we plan further to integrate SLG into our CBA system, the Case Library for Architecture (Lin and Chiu, 2003), to develop a simple but powerful analytic tool for the graphical contents of CBA such as space planning of cases in the future.

Acknowledgements

The author wishes to thank Dr. Sheng-Fen Chien for instructing the PROCESSING environment and programming skills to help us to build the prototype of SLG.

References

Lin, C.J. & Chiu, M.L. 2003, Information Mining within a Case Library, *Proceedings of CAADRIA 2003*, Bangkok, Thailand, pp. 249–262.

Arvin, S.A. & House D.H. 2002, Modeling architectural design objectives in physically based space planning, *Automation in Construction*, vol. 11, no. 2, February 2002, pp. 213–225.

Chiu, M.L. 1996, Prototypes, Variation, and Composition: A Formal Design Approach in Urban

Housing Design with Computer Assistance; Proceedings of CAADRIA 1996, Hong Kong, pp. 287–298.

Flemming, U., & Chien, S.F. 1995, Schematic layout design in SEED environment, *Journal of Architectural Engineering* vol. 1, no. 4, pp. 162–169.

Balachandran, M. & Gero, J.S. 1987, Dimensioning of architectural floor plans under conflicting objectives, *Environment and Planning B 14*, pp. 29–37.

White, E.T. 1986, Space Adjacency Analysis, *Architectural Media LTD.*, Tucson, pp. 32, 117.

Simon, H.A.1973, The Structure of Ill-structured Problems, *Artificial Inteligence* vol. 4, pp. 181–201.