SpaceMaker - Creating Space by Sketching It

Ming-Chun Lee, Ellen Yi-Luen Do
Design Machine Group
University of Washington

Abstract

Creating space is the essence of architectural design. Architects actually work with three-dimensional (3-D) space, although two-dimensional (2-D) drawings are perhaps the most commonly used design medium. It is thus valuable to help architects truly see 3-D space, while making drawings in 2-D. In addition, architects usually use symbols in their drawings to identify architectural concepts. By recognizing the symbols, it is possible to identify architectural configurations of the design. This paper introduces a symbol-based modeling tool—SpaceMaker—that allows architects to make freehand floor plans and apply symbols to identify different functional spaces. SpaceMaker then converts those floor plans into 3-D models according to the symbols.
1 Introduction

1.1 Space as the Essence of Architecture
Space is the essence of architecture (Zevi 1974). Louis I. Kahn stated a clear assertion that “architecture is the thoughtful making of spaces” (Perspecta IV 1957). Berlage also argued that “the aim of our creations is the art of space, the essence of architecture” (Grundlagen 1908). The essence of architectural design is to develop and arrange spatial relationships. Architects actually create void space. Although this void space is defined by solid elements such as walls, floors, and ceilings, it is through the space that people interact, inhabit, and engage with architecture.

1.2 Activities in the Early Design Process
In the early design phases, architects usually use different types of media to explore different design concerns. Among those, 2-D drawings are the most commonly used medium for the early schematic development. Architectural drawings reveal designer’s thinking graphically (Lawson 1994; Robbins 1994; Larkin & Simon 1987). By making drawings, architects have opportunities to re-examine initial ideas and further refine the ideas. The potential of graphic thinking lies in the continuous cycling of information from paper to eye to brain to hand and back to paper (Figure 1.1). Moreover, the activity of drawing includes both seeing and thinking about the subject being represented. Design drawing is an iterative and interactive act involving recording ideas, recognizing functions and meaning in the drawings, and finding new forms and adapting them into the design (Goldschmidt 1991; Schön & Wiggins 1992; Suwa & Tversky 1996). It serves as external symbol systems to facilitate thinking and support emergent ideas (Goel 1995).

Designing architecture is an action of creating space. Although architects actually work with 3-D space; drawings, however, are in 2-D. Architects see 3-D space in their mind’s eye when making 2-D drawings (Figure 1.2). The translation from 2-D drawings to 3-D space is usually done by architects themselves. However, the accuracy of the translation strongly depends on architects’ experience, knowledge, and capability. Novice designers might not be able to imagine space in 3-D as fluently as expert designers. Also, expert designers sometimes might enjoy seeing things in 3-D automatically without spending time to project 2-D drawings into imagined 3-D spaces. It is thus valuable to help architects see 3-D space during designing. Some architects do employ simple 3-D physical models in the early design. Architectural models are the best way to represent 3-D space (Koepke 1988). A 3-D architectural model tells a story that can be instantly understood and enables spatial relations to be perceived at a single glance (Koepke 1988; Kurrent 1999; Lucci & Orlandini 1990; Ervin & Hasbrouck 2001). 2-D drawing and 3-D modeling both have benefits to architectural design. Therefore, a tool for bridging these two design media seems important and necessary for architects developing early design concepts.

1.3 Use of Drawing Symbols
Architects usually apply symbols in their drawings to identify different design concerns such as functionality or architectural configurations (Herbert 1993; Fraser & Henmi 1994; Laseau 1989). They may use colors, different types of lines, or text to indicate different design concepts. By recognizing architects’ drawing symbols, it is possible to identify their initial ideas about spatial configurations of the design (Do 1998).

1.4 Conventional 3-D Modeling Programs
Users of conventional computer 3-D modeling programs such as Form-Z or 3-D Studio usually have to insert solid objects first to define a void space that they wish to create. The space will not be created until the architects have specified all necessary enclosing elements. A space is created from outside to inside, from solid boundaries to enclosed void (Figure 1.3).

However, architects typically do not use commercial modeling programs in the early design phases, because at this stage they usually do not deal with that specific level of detail about solid elements such as dimensions, shapes, or appearance. On the contrary, architects are concerned with void space. In fact, instead of using digital modeling tools in the early design, people tend to prefer making freehand drawings or simple massing models to study architectural relationships between spaces. A freehand drawing in the early phases indicates not only a designer’s decisions but also the associated degrees of ambiguity and vagueness. So a modeling tool for the early phases should also maintain the nature of early schematic design: abstraction, ambiguity, and vagueness.

1.5 SpaceMaker
In this paper, we introduce a prototype symbol-based modeling program—SpaceMaker—that allows architects to sketch 2-D floor plans to explore initial spatial layouts and draw symbols into the floor plans to identify different functional spaces. SpaceMaker then converts the floor plans into 3-D models using the drawing symbols to control the model detailing. The design of SpaceMaker is based on the study of the role of freehand
drawings in architectural design and the use of symbols in conceptual freehand drawings.

Compared to conventional computer modeling programs, SpaceMaker enables users to think about spaces first, to further decide the essential characteristics for each space, and then to assign appropriate enclosing elements for constructing spaces that users want to create. A space is defined from inside to outside, from the enclosed void to its solid boundaries (Figure 1.4).

2  Related Work

With the observation that drawing and modeling both play important roles in architectural design, the issue becomes how computational design tools can support freehand sketching and enable architects to generate 3-D forms easily.

2.1  Systems that Support Sketching

“Electronic Cocktail Napkin” (Gross & Do 1996), by Mark D. Gross and Ellen Yi-Luen Do, is a computer environment for making hand drawn sketches and diagrams. It aims to support freehand drawing and thinking of the sort that designers often do on the back of an envelope or cocktail napkin for creative design. Cocktail Napkin can recognize and interpret the elements and configurations of drawings made by designers, and then system retrieves relevant information. Cocktail Napkin recognizes multi-stroke glyphs and allows designers to customize the program to recognize personally defined symbols.

Do, in her doctoral dissertation, implements a prototype computational drawing environment called “Right-Tool-Right-Time” (Do 1998). Designers use different symbols in freehand drawings to explore design concepts. She argues that an intelligent computer-aided-design environment should be able to infer designer’s intentions from all drawn elements in the drawing and suggest designers with appropriate computational tools based on the inference for the task at hand.
2.2 Systems that Generate 3-D Forms by Sketching

“Digital Clay” explores the concept of the direct generation of 3-D form from freehand drawing input (Schweikardt & Gross 1998). Digital Clay derives 3-D geometry from a 2-D projected sketch. The program walks the lines of the sketch to identify convex and concave edges, then assigns 3-D coordinates to vertices, and finally generates a 3-D model that the designer can view and edit. Several similar projects are based on the same concept of deriving 3-D objects from 2-D projected drawings. Grimstead and Martin describe a method of reconstructing a Boundary-Representation (B-rep) model from a single hidden-line removed sketch of a 3-D object (Grimstead & Martin 1995). Lipson and Shpitalni introduce an Optimization-Based algorithm for reconstructing a 3-D model from a single 2-D edge-vertex graph (Lipson & Shpitalni 1996). “Quick-Sketch” by Eggli et al. also allows users to create 3-D models by making 2-D projected drawings (Eggli et al. 1995).

Do’s “Sketch VR” is a pen-based computing environment that generates 3-D objects in a VRML (Virtual Reality Modeling Language) format by recognizing simple geometric shapes in a 2-D view (Do 2001). To create an architectural space designers only simply draw lines and circles in a simple “Cocktail Napkin” 2-D sketch to indicate the appearances and locations of walls and columns. Designers draw in 2-D, and a 3-D world is created accordingly. Similarly, designers can generate 3-D models of interiors by drawing diagrams to indicate furniture placements. Igarashi’s “Teddy” is a sketching interface that allows users to draw 2-D freeform strokes on the screen, and then the system constructs polygonal surfaces to generate a 3-D model (Igarashi 2000). The system supports the modeling operation to construct a 3-D polygonal surface from a 2-D silhouette drawn by users. The system inflates the region surrounded by the silhouette to make wide areas fat and narrow areas thin in order to generate 3-D shapes.

2.3 Discussion

Electronic Cocktail Napkin can recognize drawn elements and configurations of freehand drawings and then retrieves relevant information for designers. Right-Tool-Right-Time infers designer’s intentions by recognizing all drawn elements and identifying the task context in a drawing. Based on this inference, the system then suggests appropriate computational tools for the task at hand. Both systems focus on information retrieval based on the recognition of drawn elements and configurations. However, 3-D modeling based on the recognition of drawing symbols has not been specifically implemented into either system.

On the other hand, Sketch VR does generate 3-D objects in a VRML format by recognizing simple geometric shapes in a 2-D view. The generation of 3-D objects is accomplished by an object-oriented translation from a 2-D drawn symbol to a 3-D object. Sketch VR interprets every single drawn element as an object that will be generated in 3-D format. The same concept of the object-oriented translation is also carried by Digital Clay and Teddy, both of which deal with drawing and modeling of objects. However, when designers draw symbols into a sketch floor plan to identify either the functionality or the basic spatial characteristics, designers may think about the void space instead of any single solid element. For the early schematic development, designers need a computer modeling tool that allows them to think of the basic characteristics of a space, instead of a single element, and use a symbol to represent the spatial configuration of this space. Then the tool creates a model corresponding to the spatial configuration that the symbol represents.

3 Fundamental Features in a Freehand Drawing

Freehand drawings usually consist of simple elements such as lines, simple geometric polygons, and symbols for labeling different design concerns (Do 1998). When designers start exploring the alternatives of spatial configuration, they usually decide the exterior boundary first, and then arrange the interior spaces. Typically, designers draw envelopes to encircle an area and identify the boundary. After the boundary has been defined, designers then use simple lines or bubbles to further partition the entire floor into individual rooms. By assembling lines and bubbles, architects compose the drawings and explore the spatial layouts (Figure 3.1).

Freehand drawings can be conceived as a flexible representation of designer’s thoughts about design. In a sketch floor plan, designers use a line to represent a wall, a bubble to represent an individual space, and further assemble lines and bubbles to create a floor plan that actually represents a building. Furthermore, the use of simple lines and bubbles can be a key clue of creating a 3-D modeling tool for the early schematic design. Because designers draw lines and bubbles to explore spatial configurations, a modeling tool should be able to recognize those lines and bubbles and generate a model that the designer’s drawing represents (Figure 3.2).

3.1 Use of Symbols in Freehand Drawings

A space is enclosed by defining elements, such as walls and columns, with a specific spatial configuration. When arranging spatial layouts through making drawings, designers may also start exploring the alternatives of spatial configurations of architectural elements. When designers label each space with a symbol, not only do they identify the function of the space, but they also remind themselves of the proposed configuration of architectural elements. A symbol can be interpreted as a representation of the spatial configuration in designer’s mind (Figure 3.3). By applying symbols into drawings, designers assign “meanings,” give “characteristics,” and imply “spatial configurations” to spaces.
This study mainly focuses on the idea that designers label interior spaces in floor-plan drawings with symbols to identify their functions and also indicate proposed spatial configurations of architectural elements. Furthermore, those symbols can make 3-D modeling easier. As discussed above, designers draw symbols to imply spatial configurations. Therefore, instead of drawing architectural elements to enclose spaces, designers can simply draw symbols and train the computer to recognize the spatial configurations that those symbols represent, and then have the computer create the appropriate architectural elements to enclose the space (Figure 3.4).

The computational implementation of SpaceMaker is mainly based on the studies of the fundamental characteristics of freehand drawings and the role of symbols in freehand sketches described in this section. SpaceMaker makes models by recognizing the fundamental features and symbols in designer’s freehand drawings.

4 Implementation of SpaceMaker

4.1 Scenario / SpaceMaker in Action

This section presents a scenario of how a designer uses SpaceMaker to develop a schematic floor plan. The scenario presented here helps to clarify and demonstrate the implementation of SpaceMaker.

A novice designer, Jamie decides to develop a conceptual scheme for a small office using SpaceMaker. He starts the design by clarifying the project requirement: two staff offices, a meeting room, a storage room, and a kitchen. Jamie starts to sketch a floor plan using a digitizing pen and tablet. First, he draws a rough rectangle on the tablet to indicate the boundary of the building. SpaceMaker responds by drawing a rectangular frame on the screen. Then he draws several lines to divide the floor into five rooms. Several guidelines for indicating the boundary and partition walls appear on the screen (Figure 4.1). However, Jamie understands that still no architectural elements have been displayed. Not until he labels the sketch with symbols to identify functions of space does SpaceMaker recognize the symbols and generate the corresponding architectural elements in the building.

Jamie starts to define symbols using SpaceMaker’s Symbol Definition system (Figure 4.2). After a symbol has been defined, Jamie then draws the symbol in one of the partitioned areas in the sketch, and SpaceMaker then constructs a room with the defined configuration in Space Window (Figure 4.3). Jamie continues to define other symbols for different functional spaces and goes on to complete the entire spatial layout. After completing the layout, Jamie wants to modify the space. Jamie uses the mouse to pick a guideline in Sketch Board and move it. As a result, all rooms in Space Window aligned with this line are modified (Figure 4.4).
Figure 4.1. SpaceMaker generates a floor plan in Space Window (on the right) while a designer is making a sketch in Sketch Board (in the middle). The figure also shows tool palettes on the left.

Figure 4.2. A designer draws a symbol (on the left) and defines the symbol by choosing wall elements and other parameters in Definition Window (right).

Figure 4.3. A designer draws a pre-defined symbol into Sketch Window (left); SpaceMaker then displays the setting in Space Window (right).

Figure 4.4 shows the result (on the right) of changing the spatial configuration of the entire building by moving guidelines in Sketch Board (left).

Figure 4.5. A user can change the characteristics of a wall element, for example, by opening a window in the wall.

Figure 4.6. A user views the resulting model through a VRML enabled web browser.
Figure 4.4 shows the result (on the right) of changing the spatial configuration of the entire building by moving guidelines in Sketch Board (left).

Jamie wants to change the characteristics of a wall. Jamie picks a wall from Space Window and assigns the wall an opening by drawing a rectangle (Figure 4.5).

Now, Jamie is done with the spatial layout. He saves the file in VRML format, and then views this space through the VRML enabled web browser in 3-D (Figure 4.6).

### 4.2 System Architecture

SpaceMaker consists of several parts: a freehand drawing environment with recognition, a symbol definition system, a 3-D model conversion system, and a model modification system. Figure 4.7 illustrates SpaceMaker’s system architecture.

![SpaceMaker System Architecture Diagram](image)

4.3 Freehand Drawing Environment

SpaceMaker is a freehand drawing environment allowing input from a mouse or a stylus with a digitizing tablet. Drawn input by users is categorized into three different types: exterior building boundary, interior partition walls, and drawing symbols. The three types of drawn input are then respectively handled by independent recognition systems: a building-boundary locator (Boundary Locator), a partition-wall locator (Partition Locator), and a symbol recognizer. The symbol recognizer also acts as an interior-space locator (Space Locator) (Figure 4.8).

![Freehand Drawing Environment Diagram](image)

Figure 4.9. Boundary Locator recognizes a freehand envelope stroke and identifies it as the building boundary.
After drawing the boundary envelope, designers can further divide the entire floor by drawing either lines or bubbles. When a user draws a line, Partition Locator identifies the orientation and the location of the line and marks it with a guideline (Figure 4.10).

Figure 4.10. Partition Locator recognizes several freehand lines and identifies them as the interior partition walls.

Alternatively, when a user draws a bubble, Partition Locator first makes a rectangular frame to fully contain the bubble and then checks the relative location of the partition frame to the boundary frame. If the partition frame is very close to the boundary frame, then Partition Locator will automatically attach the partition frame to the boundary frame. On the other hand, if the bubble is farther from the boundary by more than a certain minimum distance, Partition Locator will consider the bubble as an individual room. The partition frame will be placed into the interior with no attachment to the boundary frame (Figure 4.11).

Figure 4.11. Partition Locator recognizes several freehand bubbles and identifies them as the interior partition walls.

4.4.2 Locator for Space
After users complete the layout, several interior spaces have already been identified by SpaceMaker. When users then draw a symbol into one of the blank interior spaces, Space Locator identifies the space represented by the symbol. The space is identified by finding four guidelines that enclose the labeled area (Figure 4.12).

Figure 4.12. Space Locator recognizes the drawn symbol and identifies the space that the symbol represents.

4.4.3 Symbol Recognition
When designers draw symbols into the sketch, not only does Space Locator identify the location of the space, but it also tries to recognize each individual symbol. Symbol recognition is handled by Shape Recognizer, which compares the similarity of each input glyph with a set of previously defined templates created by users themselves, and then Shape Recognizer identifies the best match for the input symbol. Shape Recognizer uses basic features of the freehand input to identify symbols containing a single stroke with simple shapes, like freeform lines with limited segments, or simple geometric polygons. The features that Shape Recognizer uses include: number of corners and relative locations of starting and end points in relation to a 3x3 grid within a bounding box (Figure 4.13).

Figure 4.13. The features that Shape Recognizer uses, including number of corners and relative locations of both starting and end points in relation to a 3x3 grid.

4.5 Symbol Definition
SpaceMaker has Symbol Definition System that enables users to pre-define the architectural configuration associated with a symbol by assigning a set of four elements to the four sides of a space, the height of the space, and its condition of privacy (Figure 4.14).
Figure 4.14. Symbol Definition System has two pop-up windows: (a) symbol-sketch window and (b) symbol-definition window. A designer can draw a symbol to represent a combination of spatial boundary conditions and privacy properties.

Users first draw a symbol and then define it by choosing different elements for each individual wall. SpaceMaker so far offers three conceptual elements: a solid wall (representing a solid element without opening), a colonnade with three columns (representing an element with relative higher degree of enclosure), and a pair of columns at two ends (representing an element without enclosure) (Figure 4.15).

Figure 4.15 shows three different types of walls that designers can choose for defining a symbol.

After users have decided all symbol settings, Symbol Definition System associates that spatial configuration information with that particular symbol. When users then draw a pre-defined symbol into the sketch, Space Locator compares the features of the symbol to previously defined symbol templates and finds the closest match. Space Locator then retrieves the spatial configuration that this symbol carries and finally displays the space with pre-defined wall elements in the floor plan (Figure 4.16).

Figure 4.16. Space Locator recognizes the pre-defined symbol and inserts the elements in the Space Window according to the architectural configuration that the symbol represents.

4.5.1. Boundary Conflict

When designers draw a symbol into a sketch, SpaceMaker responds by inserting a set of four elements in the floor. However, there may be a conflict on the boundary between two adjacent spaces because two sets of four wall elements will be inserted into the floor when two symbols are applied to two adjacent spaces. The boundary between the two spaces will be specified by two potentially different elements (Figure 4.17). How do we solve the conflict?

Figure 4.17 shows the boundary conflict between two adjacent spaces.

SpaceMaker has Boundary Conflict Resolver that selects a wall element for the shared space boundary based on the privacy condition of each space to deal with the conflict. SpaceMaker allows users to decide whether a space is to be assigned “privacy priority.” Users must make this selection when they define a symbol. Boundary Conflict Resolver will decide which element is chosen according to the pre-defined settings of privacy priority. Basically, there are three rules with which
SpaceMaker handles the boundary conflict (Figure 4.18): (1) when one space has privacy priority, SpaceMaker picks the element containing less opening in order to maintain the privacy; (2) when two spaces both have privacy priority, the element containing more opening is chosen in order to ensure circulation between two rooms; (3) when neither space has privacy priority, the element containing more opening is picked.

Figure 4.18 shows the three rules that Boundary Conflict Resolver uses to handle the boundary conflict.

4.6 Modification Tools for Converted Space
SpaceMaker provides two modification tools for users, Moving Tool and Opening Tool. Moving Tool allows users to modify the model by adjusting the spatial configuration in the sketch. Users can move any guideline that controls the positions of boundaries between interior spaces. All rooms aligned with this selected guideline will be modified together (Figure 4.19).

Opening Tool allows users to directly modify the model. Users can either add an opening on the wall or replace the wall with a colonnade. Opening Tool enables a designer to change the nature of a wall and thereby to change the nature of the spaces adjacent to that modified wall (Figure 4.20).

Figure 4.19. Designers can change the entire configuration of space by moving guidelines on the Sketch Board.

Figure 4.20. Users can change the characteristics of the partition walls by (a) adding an opening or (b) switching it to a colonnade.

4.7 Conversion to VRML Format
SpaceMaker converts 2-D sketches into 3-D models in VRML format. After designers have completed a space layout, VRML Translator outputs a standard VRML format file that any VRML enabled web-browser can read. The program assumes that the world is flat and level and that all elements will be simply extruded to a height that users have assigned while defining symbols. After creating the virtual space, users can use a VRML enabled browser to navigate the space (Figure 4.21).

5 Conclusion
We have presented a prototype 3-D modeling tool, SpaceMaker, for early schematic architectural design. The computational implementation is based on studies of the features in sketch floor plans and the use of drawing symbols. This study reveals that it is possible to quickly build 3-D models by making freehand sketches and integrating designer-defined drawing symbols into sketches. By allowing users to employ a pen-based interface to explore spatial layout, SpaceMaker enables designers to work with spatial components without specifying their precise internal structure. Using SpaceMaker, designers can arrange the spatial layout based on the fundamental characteristics of space and quickly experience the space in 3-D. On the other hand, SpaceMaker also enables designers to modify and add more precise details into the model later in the design process.

Because designers can explore spatial layout without worrying about specific details when using SpaceMaker, designers can mainly focus on “void” space instead of “solid” architectural elements. You think of the space first, and then think of the configuration of solid elements that come together with the space, and finally draw a symbol to represent this space.
Designers draw to think of space, draw to explore the solution of spatial configurations, and draw to make space.

While SpaceMaker supports a convenient paper-like modeling environment, there are still some limitations due to the computational implementation. For example, the recognition systems currently only support single-stroke recognition. Users must complete any single task, such as enclosing a building boundary, partitioning the floor, and inserting a symbol, within only one single stroke input. In addition, SpaceMaker currently only recognizes a building boundary with a simple rectangular shape. It is obvious that many opportunities for extending the scope of this study remain. In the immediate future, usability tests or other formative evaluation techniques such as heuristic evaluation should be used to further refine the prototype. In addition, future versions of SpaceMaker should incorporate non-rectilinear recognition, multi-stroke input, high-level context recognition, and multi-floor layout.

SpaceMaker represents our intention to look at a different direction for establishing a 3-D modeling environment that is truly suitable to the early conceptual development for the architectural design.
References