

## Chapter 4

### A Generative Design Grammar Framework

Generative design grammars are applied by the GDAs to generate virtual world designs. This chapter presents a framework which provides guidelines and strategies for developing generative design grammars by defining the general structure of a grammar and the general structure of its basic components: design rules. Using the generative design grammar framework, designers define grammars that produce different design languages for virtual worlds, rather than predefine every detail of all possible virtual world designs. On behalf of the virtual world occupants, the GDAs carry out the design tasks by applying the grammars during real-time interactions in virtual worlds. The application of each grammar is directed to ensure the generated virtual world design satisfies the GDA's current goals that are related to designing virtual worlds. In this manner, virtual worlds are dynamically designed as needed.

Section 4.1 analyses the design problems of virtual worlds, and elaborates on the reasons why generative design grammars are suitable for designing virtual worlds. Section 4.2 presents a general framework for developing generative design grammars. Section 4.3 looks at the basic components of a generative design grammar: design rules. Section 4.4 highlights the characteristics of generative design grammars based on a comparison to shape grammars. The final section discusses general issues for applying the framework. Applying the generative design grammar framework with the integration of different design and domain knowledge, different generative design grammars can be developed to provide virtual world designs for different purposes and with different stylistic characterisations.

#### 4.1 GENERATIVE DESIGN GRAMMARS FOR FORMALISING VIRTUAL WORLD DESIGNS

The concept and development of generative design grammars for this study are inspired by the notion of shape grammars (Stiny and Gips 1972). The inspiration comes directly from shape grammars as a design formalism for describing and generating designs in general. Over the last three decades, the theories and applications of shape grammars have been applied to a wide range of design areas; for example, fine arts, architecture, industrial design, engineering design, design computing and so on. As reviewed in Chapter 2, Knight (2000) summarises that a shape grammar is a set of shape rules that can be applied in a step-by-step manner to generate a set, or language, of designs. The nature of shape grammars is both descriptive and generative:

- Shapes (points, lines, planes or volumes) as the basic components of shape rules are descriptions of the designs that the grammars generate.
- The applications of the shape rules generate designs via shape operations and spatial transformations.

Inheriting the descriptive and generative nature of shape grammars, our generative design grammars are capable of describing virtual world designs using components of the design rules and generating virtual world designs via design rule applications. The descriptive and generative qualities of generative design grammars well serve the purposes for designing virtual worlds.

#### **4.1.1 A Virtual World Design as “Objects in Relations”**

Shape grammars provide a formal approach for describing and generating designs. The way that shape grammars function lies in the unique view of designs. A design generated by a shape grammar is viewed as “elements in relations” (Stiny, 1999, 1990). To apply a shape grammar for design generation is basically to change (add, subtract or replace) the “elements” and define or alter the “relations” among the “elements” via shape rule applications. In this manner, shape grammars can generate rather complex designs based on simple design elements. This view of designs is accordant with the object-oriented nature of virtual worlds. A virtual world design can be viewed as “objects in relations”.

Most virtual worlds are object-oriented systems. They are among the examples designed using various commercial platforms, such as Active Worlds<sup>1</sup>, Blaxxun Platform<sup>2</sup>, SecondLife<sup>3</sup>, and Virtools<sup>4</sup>. An object-oriented virtual world is constructed through the placement and configuration of objects. An object in a virtual world may refer to a virtual place; for example, a virtual gallery or a virtual meeting room. An object may also refer to an entity in a virtual place, or form a part of the place; for example, a digital picture in a virtual gallery or a wall of the virtual meeting room. Each object can have an appearance of a 3D model in a virtual world, and together these models provide the visualisation of the environment. The objects then can be configured or programmed to have certain behaviours that allow the occupants to interact with the world and with each other. Therefore, a virtual world design basically comprises various objects in terms of the following two aspects (Gu and Maher 2003).

- Visually/spatially, via the use of architectural metaphors, the 3D models are composed to form an ambient environment which virtual world occupants can inhabit and where the intended activities can take place online.
- Functionally, selected objects are ascribed with behaviours to support the intended activities online. Therefore, the interactions between the occupants and the virtual world become possible.

Similar to the way shape grammars formalise designs in general, in designing virtual worlds generative design grammars describe various virtual world objects and their properties, define the relations among them, and generate virtual world designs based on these elements. This compositional characteristic of design generation makes generative design grammars an appropriate design formalism for virtual worlds.

#### **4.1.2 Design Problems of Virtual Worlds**

Generative design grammars are able to handle visual/spatial problems as well as non-visual/spatial problems in designing virtual worlds. In a virtual world, as described earlier, forms of objects are visualised using 3D models. Because of the use of architectural metaphor, these 3D models visualise architectural or architecture-like forms. The design problems that are related to the visual/spatial aspect of virtual worlds are analogical to the spatial problems of architectural designs. There are many successful shape grammar examples in the physical world; for example, the Palladian grammar (Stiny and Mitchell 1978), the Mughul Gardens grammar (Stiny and

---

<sup>1</sup> <http://www.activeworlds.com>

<sup>2</sup> <http://www.blaxxun.com>

<sup>3</sup> <http://www.secondlife.com>

<sup>4</sup> <http://www.virttools.com>

Mitchell 1980), the Prairie Houses grammar (Koning and Eizenberg 1981) and the Siza Houses grammar (Duarte 1999) that have been developed to generate and/or analyse architectural designs. Inheriting the capabilities of design description and design generation from shape grammars, generative design grammars are capable of handling similar kinds of visual/spatial problems in designing virtual worlds.

The design problems that are related to the functional aspect of the virtual worlds are non-visual/spatial problems. As described earlier, these problems are about ascribing behaviours to selected objects to support the intended activities online. For example, a simple behaviour might be to display a digital image in the virtual world when an object is mouse-clicked. While the visualisation of an object and the behaviours of the object in the world have no causal relations, the combination of them is determined by the designers' preferences, following the references to the metaphors and other design guidelines for the consistency of the virtual world design. Similar kinds of design problems have been addressed in shape grammars. Although shape grammars are spatial, Stiny (1981) demonstrates the use of description functions to address the composition of designs in other terms that are non-spatial; for example, those related to the functions, purposes, uses and meanings of the designs. Knight (1999) further suggests the linkage of an original shape grammar with a parallel description grammar to generate designs (compositions of shapes) and other non-spatial descriptions in parallel. Similarly, generative design grammars are more than visual/spatial grammars. The design rules of a generative design grammar are divided into different categories to be fired at different stages to address visual/spatial problems as well as non-visual/spatial problems for designing virtual worlds. Details of these design rule categories are discussed in section 4.2.

#### **4.1.3 Stylistic Characterisations of Virtual World Designs**

A specific style is exemplified when several designs "each create a similar impression" (Stiny and Mitchell 1978). As discussed in Chapter 2 section 2.2, stylistic characterisations of virtual world designs can be considered in terms of visualisation (layout and object design), navigation and interaction. They are three inseparable parts for providing an integral "impression" of virtual worlds. Compared to many novice designs, virtual worlds designed with a specific style in mind will achieve better consistency, and this good sense of consistency provides a strong basis to assist the virtual world occupants to orientate and interact in the world.

A shape grammar is capable of generating design instances that belong to an existing language of design as well as defining a new language of design. One might not instantly recognise the new design styles defined by shape grammars like the kindergarten grammar (Stiny 1980); however, the applications of shape grammars have provided examples that analyse many well-known styles of paintings, furniture designs and architectural designs, as well as generate design instances from these well-known styles, or even extend the styles. Examples of well-known architectural styles studied using shape grammars include the Palladian villas (Stiny and Mitchell 1978), the ancient Mughul gardens (Stiny and Mitchell 1980), the Prairie style houses by Frank Lloyd Wright (Koning and Eizenberg 1981), Malagueira houses by Alvaro Siza (Duarte 1999) and so on. March and Stiny (1985) use "syntax" and "semantics" as the two major factors to distinguish designs from one another. "Syntax" determines how the shapes are composed to represent the designs. "Semantics" describe the designs in terms of functions, purposes, uses or meanings other than shapes. Design styles that are described and generated using shape grammars are considered in terms of these two factors. The studies of shape grammars correspondingly address them through the original shape grammar formalisms and the use of description functions (Stiny 1981).

Like shape grammars, generative design grammars as a design formalism are able to describe and generate languages of designs that capture specific stylistic characterisations. As described earlier, a generative design grammar has different categories of design rules and the rules are applied at different stages to handle visual/spatial and non-visual/spatial problems in

designing virtual worlds. The rules that handle visual/spatial problems are syntactic rules, and the rules that handle non-visual/spatial problems are semantic rules. The different categories of design rules are developed to control the way virtual world designs are composed according to specific design considerations in terms of syntax (visualisation: layout and object placement), and semantics (navigation and interaction). In this manner, virtual world designs are generated with stylistic characterisations in mind.

## 4.2 GENERATIVE DESIGN GRAMMAR FRAMEWORK

The generative design grammar framework aims at providing guidelines and strategies for developing generative design grammars for the purpose of designing virtual worlds. This section presents the general structure of a generative design grammar. The general structure of the grammar's basic components - design rules - is presented in section 4.3.

A generative design grammar  $G$  is comprised of design rules  $R$ , an initial design  $D_i$ , and a final state of the design  $D_f$ .

$$G = \{R, D_i, D_f\} \quad (4.1)$$

The basic components of a generative design grammar are design rules  $R$ . The general structure of a generative design grammar for virtual worlds comprises four sets of design rules: layout rules  $R_a$ , object placement rules  $R_b$ , navigation rules  $R_c$ , and interaction rules  $R_d$ .

$$R = \{R_a, R_b, R_c, R_d\} \quad (4.2)$$

This general structure of a generative design grammar is determined by the four design phases of virtual worlds. As discussed in Chapter 2 section 2.2.1, these four phases are:

- To layout virtual places/areas in the world: each virtual place/area has a purpose that accommodates certain intended activities online.
- To configure the virtual places/areas: each virtual place/area is then configured with certain objects which provide visual boundaries of the place/area and visual cues for supporting the intended activities.
- To specify navigation methods: navigation in virtual worlds can be facilitated to consider the use of way finding aids and hyperlinks for assisting the occupants' movements among different virtual places/areas.
- To establish interactions: in general, this is a process of ascribing behaviours to selected objects in each virtual place/area so that the occupants can interact with the virtual place/area and with each other.

The four set of design rules: layout rules, object placement rules, navigation rules and interaction rules address the above four design phases of virtual worlds accordingly. The generative design grammar framework is illustrated in Figure 4.1. The firing sequence of the design rules follows the order of layout rules, object placement rules, navigation rules and finally interaction rules. These four sets of rules are discussed in section 4.3.

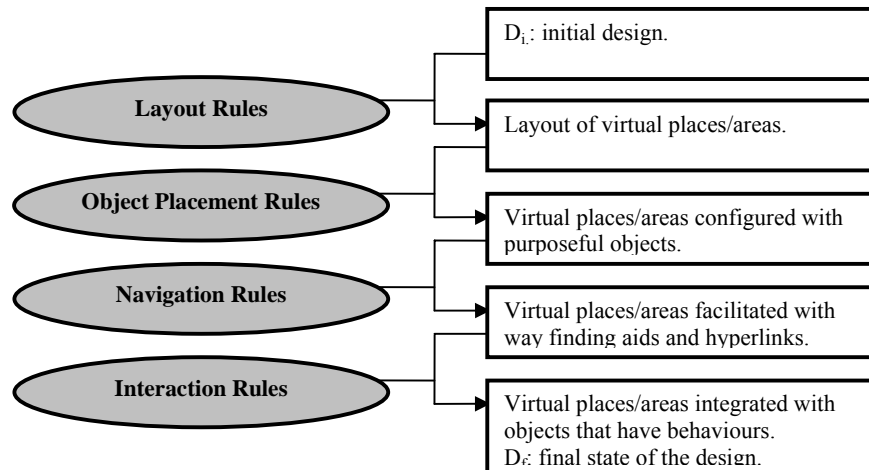


Figure 4.1. Generative design grammar framework.

Integrated with relevant design and domain knowledge, generative design grammars can be developed by following this general structure to generate virtual world designs for different purposes. The stylistic characterisations of virtual world designs - in terms of the syntax (visualisation: layout and object placement) and in terms of the semantics (navigation and interaction) - are defined accordingly in these four sets of rules.

### 4.3 GENERAL STRUCTURE OF DESIGN RULES

The basic components of generative design grammars are design rules. The general structure of design rules is similar to the general structure of shape rules. In shape grammars, a shape rule can be defined as:

$$\text{LHS} \rightarrow \text{RHS} \quad (4.3)$$

which specifies that when a left-hand-side shape (LHS) is found in the design, it will be replaced by a right-hand-side shape (RHS). The replacement of shapes is usually applied under a set of shape operations or spatial transformations. The shapes are labelled (the use of spatial labels and state labels) for controlling the shape rule applications.

Similarly, a design rule of generative design grammars is defined as:

$$\text{LHO} + \text{sL} \rightarrow \text{RHO} \quad (4.4)$$

which specifies that when a left-hand-side object (LHO) is found in the virtual world, and the state labels sL are matched, the LHO will be replaced by a right-hand-side object (RHO). The term “object” used here can refer to a virtual world object, a set of virtual world objects or virtual world object properties.

The general structure of design rules implies the following two aspects:

- State labels are singled out and expressed explicitly as sL in the structure. The use of state labels is essential to the application of generative design grammars as they direct the application to ensure that the generated virtual world design satisfies the GDA’s current design goals, or parts of the goals, that are related to designing virtual worlds. Each design rule is associated with certain state labels representing specific design contexts that can relate to the GDA’s design goals. In order for a design rule to be fired, a virtual world object, a set of virtual world objects or virtual world object properties need to be found in the virtual world that match the LHO of the design rule, and the design contexts represented by the sL of the design rule need to be related to the GDA’s current design

goals.

- The basic components of design rules are objects and their properties, not shapes. Therefore, they are not entirely visual/spatial. For the interaction rules and parts of the navigation rules, the replacement of LHO with RHO is applied under a set of general transformations.

#### 4.3.1 Layout Rules

Layout rules are the first set of design rules to be fired in the application of a generative design grammar. They are visual/spatial rules that generate the layout of virtual places/areas according to the kinds of activities supported by the virtual world. The use of different virtual places/areas for different activities provides a way of organising and allocating activities in the virtual world, and creates a sense of movement for the virtual world occupants when changing from one activity to another. Because of the use of the architectural metaphor, layout problems in virtual worlds can have similar solutions to the ones in built environments. However, designers may have more freedom in making design decisions since virtual worlds do not have to obey physical constraints. In virtual worlds, many layout-related issues; for example, adjacency, do not strictly follow their physical counterparts because virtual places/areas can be hyper-linked.

Figure 4.2 illustrates an example layout rule taken from a generative design grammar for virtual gallery designs.

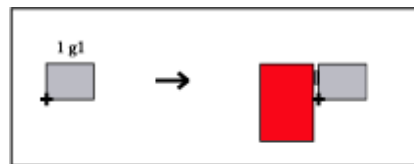





Figure 4.2. An example layout rule for generating virtual gallery layouts.

In this design rule, the LHO of the design rule is a reception area of the virtual gallery. After the rule is applied it will be replaced by the RHO: a gallery area placed spatially adjacent to the reception area.

Symbol  represents a gallery area in the virtual gallery for displaying exhibitions.

Symbol  represents a reception area of the virtual gallery.

State label  $sL=1$  indicates that layout rules will always be fired first in a generative design grammar. The meaning of state label  $sL=g1$  is to generate a certain gallery area in the virtual world. In order for this design rule to be fired the following two conditions need to be met:

- A reception area  is recognised in the virtual world by the GDA.
- The design contexts represented by  $sL=g1$  (to generate a certain gallery area in the virtual world) are related to the current design goals set up by the GDA in the process of hypothesising.

#### 4.3.2 Object Placement Rules

Object placement rules are fired after layout rules, they are also visual/spatial rules. After a layout is produced, object placement rules further configure each place/area to provide visual boundaries of the place/area and visual cues for supporting the intended activities through object placements. Because of the use of the architectural metaphor, the development of object placement rules can also refer to many design principles and examples in built environments. Depending on the designers' preferences, they may choose to simulate a built environment, or to experiment with other alternatives since virtual worlds are in fact free from any physical constraints. However,

virtual worlds designed with a consistent use of forms, colours and other visual/spatial elements are arguably more effective in assisting the occupants' orientation and interactions in the worlds.

Figure 4.3 shows an example object placement rule that generates the visual boundaries for a gallery area in the virtual gallery. Figure 4.4 shows an example object placement rule that arranges the interior of the gallery area with digital picture frame objects for displaying digital images.

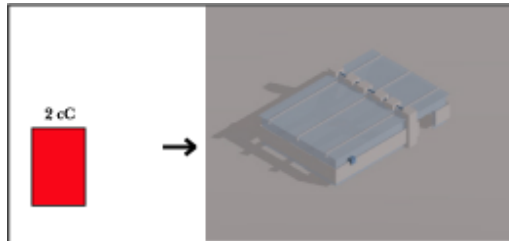


Figure 4.3. An example object placement rule that generates visual boundaries for a gallery area.

In Figure 4.3, state label  $sL=2$  indicates that object placement rules will be fired second, after the layout rules. State label  $sL=cC$  means to apply a cold colour scheme for the interior of the generated area. In order for this object placement rule to be fired, the following two conditions need to be met:

- The layout of a gallery area is generated in the virtual world.
- The design contexts represented by  $sL=cC$  (to apply a cold-colour scheme for the generated area) are related to the GDA's current design goals (in terms of design preferences).

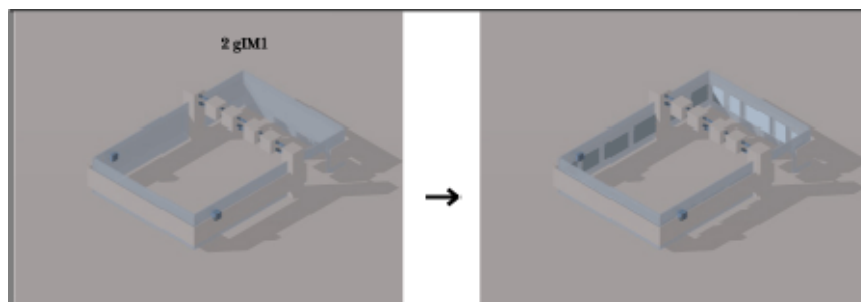


Figure 4.4. An example object placement rule that arranges the interior of a gallery area for displaying digital images.

In Figure 4.4, state label  $sL=gIM1$  means to arrange a gallery area for displaying digital images using configuration 1. In order for this object placement rule to be fired the following two conditions need to be met:

- The visual boundaries of a virtual gallery area are defined.
- The design contexts represented by  $sL=gIM1$  (to arrange the gallery area for displaying digital images using configuration 1) are related to the GDA's current design goals (in terms of the requirements for displaying exhibitions).

### 4.3.3 Navigation Rules

Navigation rules are fired next in a generative design grammar, after layout rules and object placement rules. Navigation rules provide way finding aids and hyperlinks in the generated places/areas to assist the occupants' navigation. As discussed in Chapter 2 section 2.2.3, way finding aids in virtual worlds have been studied with direct references to those in built

environments (Vinson 1999, Darken and Sibert 1996, 1993). There are at least two kinds of way finding aids that can be integrated into virtual worlds from built environments:

- The use of spatial elements; for example, paths, openings, hallways, stairs, intersections, landmarks, maps, signs and so on.
- The use of social elements; for example, the assistance gained from tour guides or other occupants.

Besides these way finding aids originating from built environments, virtual worlds also have their unique forms of navigation since virtual places/areas are hyper-linked. Most virtual worlds allow occupants to move directly between any two locations using hyperlinks. For example, in virtual worlds implemented using Active Worlds there are two kinds of hyperlinks for navigation purposes:

- Teleport portals: hyperlinks that take occupants from one location to another location in a virtual world, or from one virtual world to another virtual world without transitive movements.
- Warp portals: hyperlinks that take occupants from one location to another location in a virtual world with transitive movements.

The origin of hyperlinks used in virtual worlds can be traced back to the navigation in hypertext systems (Dourish, 1999, Ruddle et al, 1997).

Navigation rules are not entirely visual/spatial. The application of the rules indeed involves object placements for defining way finding aids and hyperlink portals in the generated virtual places/areas, which is related to the visual/spatial aspect of virtual worlds. However, before these object placements are made, navigation rules are mainly about recognising the connections among these generated virtual places/areas and finding appropriate navigation methods for the occupants to access these places/areas.

Figure 4.5 is an example navigation rule. The LHO of this rule shows that two virtual gallery areas are separated without direct access to each other. The RHO of this rule shows that a pair of hyperlink portals is created inside these two gallery areas which allows visitors to travel back and forwards freely. State label sL=3 indicates that navigation rules are the third set of rules to be fired in a generative design grammar, after layout rules and object placement rules.

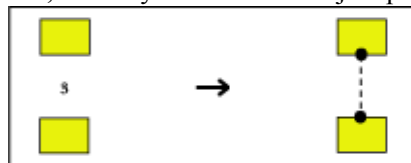


Figure 4.5. An example navigation rule for connecting two separated virtual gallery areas.

Figure 4.6 shows the effect of the navigation rule illustrated in Figure 4.5. The left-hand-side image is the interior of one of the virtual gallery areas. The right-hand-side image shows that a hyperlink portal is created and appears as a colour stone on the floor. After appropriate behaviours are ascribed, the portal will take the visitor directly to the other gallery area when it is “stepped” on.

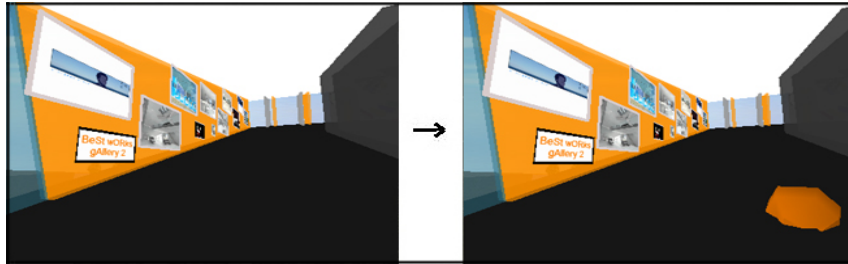


Figure 4.6. The effect of the example navigation rule.

#### 4.3.4 Interaction Rules

Interaction rules are the final set of design rules to be fired in a generative design grammar. The application of interaction rules ascribes appropriate behaviours to selected objects in each virtual places/areas. Therefore, virtual world occupants can interact with the world by triggering these behaviours.

Interactions in virtual worlds are not well developed. There are examples that derive from various gaming environments. There are also new attempts; for example, the use of computational agents, which are influenced by artificial intelligence and artificial life. The interaction rules of generative design grammars do not intend to develop new interactions in virtual worlds. The kinds of interactions generated by the interaction rules will be mainly supplied by the existing virtual world design platforms realised in the forms of object behaviours. By developing the interaction rules, conventions of triggering object behaviours during the interactions between virtual worlds and their occupants can be defined. For example, in Active Worlds there are three different ways to trigger an object behaviour: create, activate and bump. Taking an example of displaying a digital image on the surface of a virtual world object, command “create picture” allows the digital image to be shown on the surface of the virtual object as soon as the occupants connect to the world. Command “activate picture” requires the virtual object to be mouse-clicked in order for the image to be shown, and command “bump picture” requires the occupants’ avatars to collide with the object in order for the image to be shown. These three different methods demand different levels of involvement from the virtual world occupants. Designers can use them selectively to serve different design purposes.

Interaction rules are non-visual/spatial rules that recognise selected objects in the virtual world and ascribe appropriate behaviours to these objects. There are two different types of interaction rules. One supplements object placement rules and the other supplements navigation rules. Object placement rules define visual boundaries for each generated place/area and place purposeful objects in the place/area. The first type of the interaction rule ascribes behaviours to relevant objects in order to support the intended activities in these virtual places/areas. The other type of interaction rule looks for way finding aids and hyperlinks generated by navigation rules and ascribe appropriate behaviours to activate them.

Because interaction rules do not operate on a visual/spatial level they are not appropriate to be expressed using illustrations. Examples of interactions rules can be found in Chapter 5 section 5.6. They are expressed in the form of “IF... THEN...”, taken from a generative design grammar developed for virtual gallery designs. Without getting into the technical details of ascribing behaviours to virtual world objects, Figure 4.7 shows the effect of an example interaction rule of the first type for supplementing object placement rules. The left-hand-side image is the exterior of a virtual building with an empty advertisement board. The right-hand-side image shows the same advertisement board displaying digital images in an animated sequence, after the interaction rule is fired, which configures the object properties of the advertisement board object using a scripting language to enable the animation to be shown.

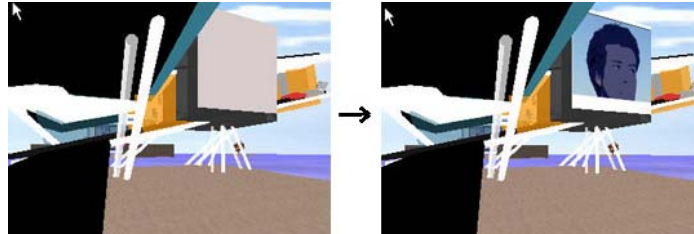


Figure 4.7. The effect of an example interaction rule.

#### 4.4 CHARACTERISTICS OF GENERATIVE DESIGN GRAMMARS

Generative design grammars both adopt the descriptive and generative nature of shape grammars, and modify some of the original shape grammar properties to suit the purpose of designing virtual worlds. As design formalisms they have similarities as well as differences. We highlight the characteristics of generative design grammars based on a comparison to shape grammars in general. The comparison is drawn in terms of the following three aspects: grammar components, meeting design goals and computation.

Grammar components: in shape grammars, a set of shape rules, an initial shape and a final state constitute a shape grammar (Knight 1994). Shape rules are applied recursively in a step-by-step manner to generate designs. Spatial and state labels are used to control the application of shape rules.

- In generative design grammars, similarly, a grammar comprises design rules, an initial design and a final state of the design. More specifically, a generative design grammar has four sets of design rules: layout rules, object placement rules, navigation rules and interaction rules. They can be categorised as syntactic and semantic rules. Each set of the design rules corresponds to a design phase of designing virtual worlds.
- Generative design grammars also use spatial labels and state labels to control the application of design rules. The original use of state labels in a shape grammar is to control the sequence of shape rule applications. In a generative design grammar, on one hand, this original purpose is maintained so that the design rules can be applied in the sequence of layout rules, object placement rules, navigation rules and interaction rules. On the other hand, a special set of state labels are developed to represent a set of design contexts that can relate to the GDA's design goals of designing virtual worlds. Using these special state labels, the application of the generative design grammar is directed to ensure that satisfactory virtual world designs are generated.

Meeting design goals: Knight (1999) points out that there are generally two different ways to connect shape grammars with design goals. The first approach is to integrate foreknowledge into the shape rules so that the generated designs will meet the given goals. The key process here is to add constraints to the original, unrestricted shape grammars, to therefore increase the predictabilities of the designs they generate. The second approach is to apply an automated search and test strategy to the generated designs. The search and test approach does not require constraints for the shape grammars since the generated designs are searched and tested to select the ones that satisfy the design goals.

- Generative design grammars are restricted. They are similar to set grammars (Stiny 1982). Set grammars are restricted kinds of shape grammars. According to Knight (1998b), the main difference between set grammars and shape grammars is the way designs are decomposed. In a set grammar, each generated design can only be viewed and decomposed in one definite way. In a shape grammar, each generated design can be viewed and decomposed in unlimited way. This liberty of shape grammars on one hand supports two very important aspects of shape grammars' generative power: emergence

and ambiguity. On the other hand it also makes it very difficult to predict the behaviours and design outcomes of shape grammar applications. Similar to designs generated by set grammars, a virtual world design composed using generative design grammars can be decomposed uniquely into a set of purposeful virtual world objects and their properties. The application of a generative design grammar also needs to be controllable and predictable to a certain extent so that the generated designs can be useful for the purposes of this study, since virtual worlds in the context of this study are essentially functional online places. The purpose of generating virtual world designs that meet the GDAs' design goals is given higher priority over the urge of maintaining emergence and ambiguity for generative design grammar applications.

- As discussed earlier, to direct the application of a generative design grammar relies on the use of a special set of state labels. Each design rule in a generative design grammar is associated with one of these state labels which represent a specific design context of designing virtual world. In order for a design rule to be fired, besides the LHO of the design rule needed to be found in the virtual world, the design context represented by the state label of the rule also need to be related to the GDA's current design goals. In this manner, the application of the generative design grammar is directed to generate virtual world designs that meet the GDA's current design goals.
- Generative design grammar applications are carried out by GDAs during real-time interactions in virtual worlds. The generated designs are implemented and put into use immediately. Each virtual world design needs to meet the GDAs' current design goals, and to be generated with efficiency in order to catch up with the pace of real-time interactions in virtual worlds. The search and test approach is impractical in this respect, and therefore was not considered further from the beginning.

Computation: in the initial studies of shape grammars, the design generation is performed by hand. To manually perform the step-by-step application and examine and present different generated design instances, is very time-consuming. Computers appear to be a perfect tool for handling these tasks. Nowadays, according to Gips (1999), the computer implementations of shape grammars have at least four different purposes. The most common purpose is to assist the generation of designs from a shape grammar. The other three purposes are to analyse if a given design belongs to the language generated by a certain grammar, to generate a shape grammar based on a corpus of given designs, and to assist designers in designing shape grammars. Gips further points out that the challenge in computer implementation of shape grammars lies in the tension between the spatial nature of shape grammars and the symbolic nature of the underlying computer representations and processing. As a result of this tension, in many cases the original designs of shape grammars need to be restricted and adapted for computer implementations; for example, to be implemented as set grammars.

- In contrast to shape grammars, generative design grammars are made for computer implementations. The GDAs who apply the grammars, and the virtual world designs that generate by the grammars, are both immersed in computer environments. In this respect, computer implementations are not options for generative design grammars, but the only forms of generative design grammars. As discussed earlier, generative design grammars are restricted, which reflects the essence of designing virtual worlds as place marking, as well as software system designing.

## **4.5 APPLICATION ISSUES OF THE GENERATIVE DESIGN GRAMMAR FRAMEWORK**

The generative design grammar framework has been introduced in the sections above. Following the general structure of a grammar and its basic components - design rules - generative design grammars can be developed for the purpose of designing virtual worlds. The application of such a grammar begins when the GDA hypothesises design goals that are related to designing virtual worlds and recognises the initial design  $D_i$  in the virtual world. The appropriate design rules are then selected and fired in the order of layout rules, object placement rules, navigation rules and interaction rules. The application is terminated when there are no rules to be fired; that is, no LHO and sL can be matched. The final state of the virtual world design  $D_i$  is generated for the current moment. When new design goals are hypothesised the application of the generative design grammar will start all over again.

The following summarises issues for applying the generative design grammar framework.

- The generative design grammar framework is presented at a general level. The development of specific generative design grammars should apply this framework by integrating appropriate design and domain knowledge for implementing the four sets of design rules. Different generative design grammars are able to generate virtual world designs for different purposes and with different stylistic characterisations. Different GDAs may have different generative design grammars reflecting their own identities in virtual worlds. In Chapter 5, an example grammar for dynamic design of a virtual gallery developed using the generative design grammar framework is presented, and the application of this example grammar is then demonstrated in Chapter 6 using a design scenario.
- Generative design grammars are inspired by the notion of shape grammars, yet are different from shape grammars. Their similarities and differences have been discussed earlier in section 4.4. The development of generative design grammars should bear these comparisons in mind.
- A generative design grammar is applied by a GDA. The application of the grammar is closely related to the GDA's computational processes. For example, through the process of interpretation, the GDA recognises virtual world objects and their properties for matching the LHOs of design rules, and through the process of hypothesising the GDA sets up design goals so that the sLs of design rules can function. This requires the development of generative design grammars that are compatible with the development of GDAs so that data can flow between these two systems.