RULE-BASED COMPOSITION

SOTIRIOS D. KOTSOPoulos
Massachusetts Institute of Technology, USA.
skots@mit.edu

Abstract. A framework of rule-based composition, appropriate for the architectural studio, is outlined. Rules, grammars and a shape grammar digital interpreter are put into use in the generation of designs.

1. Introduction

The paper outlines a framework of rule-based composition for the architectural studio. How can rules, grammars, and digital tools become part of the core studio teaching in the schools of architecture? How can rules be organized to perform goal-driven design tasks? And how can analogue and digital systems coexist in the design process?

This study is conducted along the lines of the formal design theory introduced in Stiny (1976, 1980, 1991). The proposed approach generates compositions without analysis of pre-existent designs. A housing project based on a design competition is used as example. Given the building program, the objective is to determine sequences of rules for the generation of 2D plan arrangements and 3D models. It is proposed that the heuristics of the process be organized in three levels: First, formation rules produce partis. Second, transformation rules produce wall-layouts from a chosen parti. Third, refinement rules determine certain tectonic details (windows, doors, and stairs) in chosen wall layouts.

Initially, possible sets of rooms, spatial relationships and rules are formulated with analogue means, as a hypothesis. The evaluation of spatial relations is based on the produced room-adjacencies. The rules are tested using a digital parametric 2D shape grammar interpreter. The interpreter provides computer aid in clarifying the ramifications of the hypothesis at the levels of formation and transformation. If a rule-set is not producing interesting arrangements it is modified and re-tested.

2. The Project

A housing competition sponsored by the Habitat For Humanity (HFH) in the summer of 2002 in Boston, Massachusetts was used as experimental project. The HFH described the goal of the competition as: “the building of simple, decent, affordable
houses”. The program called for adaptable types of 2, 3 and 4-bedroom houses without determining the square-footage of rooms or house types. All houses included: primary covered entrance, circulation, dining area, living area, at least one full bathroom, kitchen, and bedrooms. You can see examples of HFH housing in East Boston, Dorchester and Roxbury in Figure 1.

![Figure 1. Examples of HFH housing in East Boston, Dorchester, and Roxbury.](image)

A minimum living space limit for all types was suggested: 900 s. f. for 2-bedroom apartments, 1050 s. f. for 3-bedroom apartments, and 1150 s. f. for 4-bedroom apartments. The organizing committee did not specified sites, but offered possible ones: Small, quadrilateral lots less than 5000 s. f. were an option, but lots larger than 20000 s. f. were also typical, In Figure 2 are diagrammatic representations of two typical HFH sites.

![Figure 2. Two examples of typical HFH sites.](image)

3. The Method

The aim of the proposed design process is to develop rule-systems able to generate houses of variable size and morphology. The systematization of the ground plan is the medium used for the attainment of the objective.

The computational apparatus defined in Stiny (1980) within which shapes that
belong in some algebra $U_{ij}$ are composed with rules of the form $x \ y$, is employed in the production of house designs.

For $i = 0$, the $U_{ij}$ algebras contain points. For $i = 1, 2, 3$ the algebras contain lines, planes and solids. Lines belong to the $U_{ij}$ algebras. Each shape is defined as a finite set of lines of finite and possibly zero length maximal with respect to one another, manipulated on a line ($U_{11}$), a plane ($U_{12}$), or, in space ($U_{13}$). Planes belong to the $U_{2j}$ algebras. Each shape is defined as a finite set of planes of finite and possibly zero area, maximal with respect to one another, manipulated on a plane ($U_{22}$), or in space ($U_{23}$). Solids belong to $U_{33}$ algebra. Each shape is a finite set of solids of finite and possibly zero volume, maximal with respect to one another, manipulated in 3D space.

Similar rule-based models for the treatment of symbols can be found in Carnap (1937), and Chomsky (1957). The possibility of establishing analogous methods in design was first discussed in Stiny and Mitchel (1978) in the production of Palladian villa plans. Numerous papers have followed describing the generation of Frank Loyd Write’s prairie houses (Koning and Eizenberg, 1981), Japanese tea-room designs (Knight, 1981), Queen Ann houses (Flemming, 1967), Taiwanese houses (Chiou and Krishnamurti, 1995), Yingzao fashi houses (Li, 2000), and Siza’s houses (Duarte, 2001).

The novelty of the proposed approach is that it attempts to capture the exploratory effort of an intuitive design process. A process of this kind involves selection among rules, where the designer tests their possible outcomes. The heuristics of the process are organized in three levels. At the first level of formation, the rules apply on the finite set of rooms to produce diagrammatic parts for designs. At the second level of transformation, a specific parti is selected and transformed to a wall layout. From one parti infinite wall layouts can be produced. At the third level of refinement, the rules apply on wall layouts to determine certain details such as doors windows, stairs, etc. The framework can be described as follows:

$$
\Sigma : \{ \text{finite set of rooms} \}
$$

$$
R : \{ \begin{array}{ccc}
\text{Formation} & & \text{Transformation} & & \text{Refinement} \\
A_1 \rightarrow F_1 & & G_1 \rightarrow M_1 & & N_1 \rightarrow W_1 \\
\vdots & & \vdots & & \vdots \\
A_n \rightarrow F_n & & G_k \rightarrow M_k & & N_r \rightarrow W_r
\end{array} \}
$$

where $A_1, \ldots, A_n, W_1, \ldots, W_n$ are elements in $\Sigma$.

At each level the rules produce descriptions. The transformation and refinement levels require some input from their preceding level. Samples from the three levels are presented next in Figure 3.

Analogue and digital means coexist in the design process. The analogue part involves the articulation of spatial relations and rules. The digital part involves the testing. Through an iterative process of formation, transformation and refinement
the rules are evaluated and redefined according to their compliance to programmatic, intuitive, or other criteria.

4. The Interpreter

A shape grammar interpreter (Liew, 2003) written in VisualLISP uses a scripting language based on LISP to describe a rule. The digital interpreter is used in the exploration of partis, and wall layouts. Using the interpreter the designer determines
if a set of rules produces any desired arrangements. If not, the rules are modified
and re-tested. In the evaluation the designer selects among the alternative rules.
The interpreter requires conversion of the rules into LISP scripting format.

Each rule includes left-hand schema, right-hand schema, transformation
mapping, and variable mapping. A rule is composed in three parts: The first part
describes the geometry of the left-hand schema of a rule. The second part describes
the geometry of the right-hand schema of a rule. It also includes the transformation
and variable mapping, which determines the transformation and the parameter
relationships between left and right-hand schemata. The third part of a rule links
the previous three parts.

A vector description format (Nagakura, 1995) is used to describe the geometry
and variables of a schema. The transformation mapping determines any
transformation changes between the left-hand schema and the right-hand schema.
The variable mappings define a relationship between the parameters of both
schemata. A schema is composed of two parts, the geometry and the constraints on
the geometry variables. The geometry of a schema is described by a series of vector
displacements. Each vector has 3 components: action, vector and label. The action
component determines if the shape is a line or a point. The vector component
describes the x and y displacement of the shape. The label component determines
the name of the label. For example, a horizontal parti line that is 5 units long is
described as:

```
((action "line") (vector 5 0) (label "parti"))
```

A shape is described as a series of vector displacements that are connected
from end to end. For example, the following describes a "parti" square that is 5
units by 5 units in size.

```
(((action "line") (vector 5 0) (label "parti"))
 ((action "line") (vector 0 5) (label "parti"))
 ((action "line") (vector -5 0) (label "parti"))
 ((action "line") (vector 0 -5) (label "parti")))
```

To describe a parametric shape, the numbers in the vector displacement
description are substituted with variables. The following describes a schema that
finds all parti rectangles.

```
((action "line") (vector 1 0) (label "parti"))
 ((action "line") (vector 0 w) (label "parti"))
 ((action "line") (vector (- 1) 0) (label "parti"))
 ((action "line") (vector 0 (- w)) (label "parti"))
```

Restrictions can be set on the geometry variables to limit the type of sub-shapes
found. Theses restrictions are added in the binding-constraints component of the
schema. The following example restricts the size of the square to be less than 10
units.

((binding-constraints
  (l (< l 10))
  (w (< w 10)))

To apply a rule of the form $x \rightarrow x + y$,

the interpreter recursively searches the input shape for all instances of the left-hand schema and presents the possibilities through an interactive menu that highlights the embedded schemata. Once the user selects an embedded schema, the rule application is completed by subtracting the selected schema from the input shape and adding the right-hand schema of the rule.

5. Example

Each house unit is approached as an arrangement of rooms. The possible combinations of rooms are treated by shape rules. First, spatial relations and rules are examined abstractly. Then, they are applied on specific rooms to generate designs.

5.1. SPATIAL RELATIONS AND SHAPE RULES

Two parametric rooms and their boundaries can form four general relations: (i) they can be placed next to each other to share one common boundary, (ii) they can be placed so that they do not touch, (iii) they can meet in a corner, or, (iv) they can be placed one inside the other, or they can share some area. These generic spatial relationships are presented in the next Table 1.

TABLE 1. Generic spatial relations.

The relation 1, of Table 1 depicts two adjacent rooms having a common boundary. For each parametric rectangle made out of lines (Figure 4) two parameters
$L_j$ and $W_j$ are defined: $L_j$ represents the length, and $W_j$ the width. It is $L_j W_j$.

Figure 4. Parametric rectangles made out of lines are used in the composition.

For $L_2 = W_1$ the relation 1 of Table 1 forms the relation A (Table 2). Four more spatial relations, between the same pair of rectangles, are also distinguished.

TABLE 2. The five spatial relations A, B, C, D, E.

<table>
<thead>
<tr>
<th>relation A</th>
<th>relation B</th>
<th>relation C</th>
<th>relation D</th>
<th>relation E</th>
</tr>
</thead>
</table>

The relation A produces an I-shape, the relations B, C, D produce L-shape and the relation E a T-shape arrangement.

The five spatial relations are generated by a rule of the form $x \rightarrow x+y$. The five spatial relations of Table 3 are produced by five instances of the rule. These are presented in the next Table 3.

TABLE 3. Five instances of the rule R1: R1A, R1B, R1C, R1D, R1E.

<table>
<thead>
<tr>
<th>R1A</th>
<th>R1B</th>
<th>R1C</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>R1D</th>
<th>R1E</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The five rules R1A, R1B, R1C, R1D, R1E can be expressed by two parametric rules. First, the rule R1AB 9 (Figure 5) indicates that for $D_w < W_1$, a rectangle (or rectangular solid) is added on the short side of an initial shape.
Second, the parametric rule R1CDE (Figure 6) indicates that for $D_L < L_1$ a rectangle (or rectangular solid) is added on the long side of an initial shape.

5.2. DERIVATION

Spatial relationships and rules from the previous section are put into use to generate designs. At the stage of formation labelled rectangles in $(V_{02}, U_{12})$ algebra, and labelled solids in $(V_{03}, U_{33})$ algebra are put into use. In the transformation and refinement all labels are omitted. The 2D descriptions use lines manipulated on the plane in $(U_{12})$ algebra, while 3D representations use solids manipulated in space, in $(U_{33})$ algebra.

5.2.1. Formation

The first group of rules produces partis. Given a finite set of labelled rectangles $\Sigma$, representing rooms: living area “li”, bathroom “ba”, kitchen “ki”, bedrooms “be” and auxiliary spaces “au”. An initial shape $I \in \Sigma$ is also designated. In the working example the shape representing the living room initiates the process.

$\Sigma : I$

The room-adjacencies as shown in Table 4 serve as criterion for the generation of partis according to rules. A set of restricted shape rules is presented in Table 4.
TABLE 4. Set of restricted shape rules.

Rule 1 is an instance of the rule R1A. Rule 2 is an instance of rule R1C. The shape for “au” is specified here as a square, while the symbol specifies the direction of room-addition and prevents undesirable overlapping. Rules 3 and 4 specify how bedrooms can be added. Rule 5 prevents the generation of partis without bedrooms. Rule 6 allows the addition of floors. Restricted 3D rules form a parallel rule-set (Table 5).

TABLE 5. Restricted 3D shape rules.

The derivation of a parti in 2D and 3D is presented below (Figure 7). For simplicity name labels are omitted in 3D derivations.
Figure 7. Example of parti derivation in 2D and 3D.

The produced parti is presented next in 2D and 3D (Figure 8).

![Figure 8. Example of a produced parti.

The partis are generated by rules that describe how to compose them. During the process of formation several rules are tested. They are restricted with labels to generate the preferred partis. The criteria of evaluation vary. In the example the evaluating basis was the adjacencies among rooms.

5.2.2. Transformation

In transformation an input parti is used and boundary layouts are produced. The transformation rules can often convert the input into a new arrangement with a new derived parti. Or, they produce alternative boundary layouts without changing the
Two descriptions A and B are presented next in 2D and 3D (Table 6). Description A includes the input *parti* as it is produced at the level of *formation* as shown in Table 6. Description B includes a possible wall layout.

TABLE 6. The input *parti* (left) and the produced boundary-layout (right).

<table>
<thead>
<tr>
<th>Description A (formation)</th>
<th>Description B (transformation)</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image1" alt="Diagram" /></td>
<td><img src="image2" alt="Diagram" /></td>
</tr>
</tbody>
</table>

The diagram of Figure 9 exhibits an overview of how the wall-layout of description B is produced after the application of the parametric rule $r$:

![Diagram](image3)

The rule substitutes the volume of a room with the corresponding wall boundary. At the root of the tree we see the *parti* before the application of the transformation rule. The transformed shapes appear at the bottom leaves.
Further, another very common transformation-rule is one that creates openings in the wall layout. The distribution of solid and void is thus specified. The openings are not specified as doors, windows, etc. The rules of transformation were encoded into the scripting meta-language and 2D variations were derived from a single parti. The produced arrangements were extruded manually in AutoCAD to generate 3D forms.

The next variations (i–viii) of Table 7 are some of the produced wall layouts with different kinds of openings.
TABLE 7. Alternatives Bi-viii of derived boundary-layouts.

<table>
<thead>
<tr>
<th>B_i</th>
<th>B_ii</th>
<th>B_iii</th>
<th>B_ivi</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image1" alt="Diagram" /></td>
<td><img src="image2" alt="Diagram" /></td>
<td><img src="image3" alt="Diagram" /></td>
<td><img src="image4" alt="Diagram" /></td>
</tr>
<tr>
<td>B_v</td>
<td>B_vi</td>
<td>B_vii</td>
<td>B_viii</td>
</tr>
<tr>
<td><img src="image5" alt="Diagram" /></td>
<td><img src="image6" alt="Diagram" /></td>
<td><img src="image7" alt="Diagram" /></td>
<td><img src="image8" alt="Diagram" /></td>
</tr>
</tbody>
</table>

The selected boundary-layout appears in the next Table 8, in description B. The parallel descriptions A and B are different representations of the same design in 2D (algebra $U_{12}$) and in 3D (algebras $U_{23}$ and algebra $U_{33}$).
To summarize, in transformation alternative options of boundary layouts are produced from a parti, and the distribution of solid-void is defined.

5.2.3. Refinement

In the process of refinement we gradually determine some of the tectonic details of the produced wall layout arrangements. For example, refinement can include the definition of a construction grid \( C_1 \), the specification of window zones, and window-door openings \( C_{ii} \), or the specification of stairs and other functional details \( C_{iii} \), etc. These descriptions appear in Table 9.

<table>
<thead>
<tr>
<th>Description ( C_1 )</th>
<th>Description ( C_{ii} )</th>
<th>Description ( C_{iii} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>U_{33}</td>
<td>U_{33}</td>
<td>U_{33}</td>
</tr>
</tbody>
</table>

The refinement process requires an input wall-layout on which the refinement rules apply. The rules do not convert this input into one with a new parti. The entire
process is performed in AutoCAD manually, without using the interpreter. It is presented here briefly. The three parametric rules (a–c) of Table 10 specify what kind of opening is applied on an existent void. Three opening-types are used: door (a), window (b), door-window (c).

TABLE 10. Three parametric rules for the specification of openings.

<table>
<thead>
<tr>
<th></th>
<th>(a)</th>
<th>(b)</th>
<th>(c)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><img src="example1.png" alt="Image" /></td>
<td><img src="example2.png" alt="Image" /></td>
<td><img src="example3.png" alt="Image" /></td>
</tr>
</tbody>
</table>

An overview of the developments caused by the rules (a–c) appears in the next Figure 10. The input arrangement appears at the root of the tree.

*Figure 10. Developments caused by the rules (a), (b), (c) of opening specification.*
6. Results

The descriptions of Figure 11 show three concluding representations: *parti*, wall-layout, and final design that correspond to the three levels, *formation*, *transformation* and *refinement*. A view of a 3D model is also provided in Figure 5.

<table>
<thead>
<tr>
<th>Formation</th>
<th>Transformation</th>
<th>Refinement</th>
<th>Model</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image1" alt="Formation" /></td>
<td><img src="image2" alt="Transformation" /></td>
<td><img src="image3" alt="Refinement" /></td>
<td><img src="image4" alt="Model" /></td>
</tr>
<tr>
<td><img src="image5" alt="Formation" /></td>
<td><img src="image6" alt="Transformation" /></td>
<td><img src="image7" alt="Refinement" /></td>
<td><img src="image8" alt="Model" /></td>
</tr>
<tr>
<td><img src="image9" alt="Formation" /></td>
<td><img src="image10" alt="Transformation" /></td>
<td><img src="image11" alt="Refinement" /></td>
<td><img src="image12" alt="Model" /></td>
</tr>
<tr>
<td><img src="image13" alt="Formation" /></td>
<td><img src="image14" alt="Transformation" /></td>
<td><img src="image15" alt="Refinement" /></td>
<td><img src="image16" alt="Model" /></td>
</tr>
</tbody>
</table>

*Figure 11. Examples of designs produced by the same set of rules, in axonometric.*
7. Conclusion

The paper outlined an educational framework of rule-based composition for the architectural studio. The framework uses rules, grammars, and digital tools. A housing project is used as example. The aim is to combine theoretical devices such as rules and grammars and digital tools such as scripting and modelling, to treat common studio projects. Another objective is to show how some known studio techniques can be approached computationally without losing their expressive power.

The rules perform simple design tasks that are organized to achieve three objectives: First, formation rules generate diagrammatic arrangements (partis). Second, transformation rules generate wall layouts from a chosen parti. The transformation rules also organize the distribution of solid and void. Third, refinement rules determine some of the tectonic details (windows, doors, and stairs) in chosen wall layouts.

Analogue and digital means coexist in the design process. In all three levels, the rules are formed with analogue means and then are tested digitally. A shape grammar 2D digital interpreter is used for this purpose. The digital tool is particularly useful in the exploration of 2D partis, at the stage of formation. In the transformation stage the symbolic expression of rules become increasingly complex. The refinement process is performed manually in AutoCAD, without using the interpreter. Additional 3D descriptions are executed manually in Auto-CAD without the interpreter.

The compositions are generated without analysis of pre-existent designs. In order to identify the appropriate rules one examines what they produce. In the example, the selection of spatial relationships and rules happens on the basis of the produced room-adjacencies. This is only one of the many possible ways to form spatial relationships and rules.

Spatial relationships and rules are initially formed with pencil and paper as a hypothesis. The first hypothesis is facilitated by analogue representation: values and variables within the parametric rules remain undetermined, and open. The digital representation is more efficient in deriving the consequences of a hypothesis, by allowing the mechanical execution of large number of tests. In this way, the digital exploration helps to determine if a particular rule-set produces any desired results. If not, the rules can be modified and re-tested.

In the digital format the rules are translated in the symbolic meta-language, which is based on VisualLisp. The values of the variables and the proportions need to be specified before the rules can take digital form. This is usually possible only after some limited testing with analogue means.

References


