EXPRESSING PARAMETRIC DEPENDENCE IN SHAPE GRAMMARS,
WITH AN EXAMPLE FROM TRADITIONAL CHINESE
ARCHITECTURE

ANDREW I-KANG LI
Department of Architecture
The Chinese University of Hong Kong
Shatin, N.T., Hong Kong, China

Abstract. Shape grammars traditionally generate one product at a time. This leads to
difficulties when dependent parameters are involved. Parallel grammars are proposed as
a solution. As an example, a grammar is shown which generates plans according to the
12th-century Chinese building manual Yingzao fashi.

1. Introduction

Shape grammars have demonstrated that they can characterize constructively a
wide range of styles: from Palladian villas to Wrightian prairie houses, from
traditional Taiwanese houses to Hepplewhite chair backs. Despite this variety,
these grammars have one important similarity: a computation begins with one
object and ends with one object.

However, shape grammars organized in this way can be unsatisfying
when they involve dependent parameters. Take, for example, Flemming’s (1987)
Queen Anne grammar, which generates plans by instantiating rooms around an
initial entrance hall. Flemming assumes that the constraints on room sizes can be
added without structural changes to the grammar.

It should be noted that to generate realistic layouts, information about the context,
especially the shape and size of the lot, would have to be taken into account. For
example, it must be assured that the layout under development actually fits
within the given boundaries and, possibly, keeps desired setbacks. In addition,
the dimensions of the spaces allocated would have to be considered in an explicit
way…. Such aspects could be incorporated easily into the grammar but are
omitted here because they are rather trivial and would only distract from more
basic issues, in this case, from a concentration on basic spatial relations between
spaces. (Flemming 1987, 333, original emphasis)

In fact, these aspects are not trivial. The problem is that we do not know
whether the plan fits on the site until we have finished the computation; we work
by trial and error. To put it another way, knowing the size of the site before the
computation is begun is no help, because the grammar does not tell us how to
use that knowledge to set the room sizes appropriately. This is because the
parameters involved are dependent: the sizes of the rooms depend on the number of rooms in the plan, which is known only after the computation has been finished.

However, the grammar requires us to set the sizes of rooms before we have finished the computation. We need a way to handle the parameters separately, so that we first set the values of the independent parameters (numbers of rooms) and only then set those of the dependent parameters (sizes of rooms).

One way of doing this is to have grammars in parallel, which operate on shapes in parallel (or, to use a CAD analogy, in different layers). The parameters can then be assigned to the various shapes according to the nature of their dependence. This is a direct application of algebras combined in Cartesian products as developed by Stiny (1992). As an example, I offer a partial and
simplified grammar of the twelfth-century Chinese building manual *Yingzao fashi*.\(^1\)

\[ A^8 \rightarrow \]

\[ A^7 \rightarrow \]

\[ A^8 \rightarrow \]

\[ A^9 \rightarrow \]

*Figure 1. continued. Part A.*

### 2. Parametric dependence in the *Yingzao fashi*

The *Yingzao fashi* treats a wide range of issues in the design and construction of buildings, but here we will consider only the constraints on the size of the plan, in terms both of the number of bays (in width and depth) and of the dimensions of the bays. In this simplified account,\(^2\) there are four parameters:

:: **Number of bays in width;** this number must be odd. This is ensured by the production algorithm which starts with a center bay and adds two bays at a time. Buildings are symmetrical about the central axis.

---

\(^1\) The *Yingzao fashi* was written by Li Jie (d. 1110), architect in the Song court, and published in 1103. For a brief introduction in English, see Glahn (1984).

\(^2\) There are three simplifications. First, I do not observe the distinction among building types (*dian* halls, *ting* halls, and other buildings). Second, the grammar does not generate grids smaller than $5 \times 3$. These can be generated as special cases with additional rules, which I omit for clarity. Third, I have omitted two subparameters of bay width: the number of bracket sets (*puzuo*) and their spacing.
:: **Number of bays in depth.** There are no constraints.

:: **Width of bays.** Bays are at least 200 fen and at most 300 fen wide.\(^3\) They are named according to their position in the front façade: the center bay

---

\(^3\) The fen is a modular unit of length, with eight possible values ranging from 0.96 cm to 1.92 cm. The value of the fen is related to the size and type of the building and the rank of its occupant.
EXPRESSING PARAMETRIC DEPENDENCE IN SHAPE GRAMMARS

(xin jian) is in the center; the end bays (shao jian) are at the ends; all others, if any, are succeeding bays (ci jian), i.e., they succeed the center bay. Bay width can decrease from the center to the end bays. That is, a bay can be narrower than the next bay on the inside.

:: Depth of bays. Bays are at most 300 fen deep.

Figure 2. continued. Part B.

3. The grammar

The grammar has two parts and two products: a diagram, which has no dimensions, and a plan, which does. Part A generates the grid, which shows what Flemming (1987, 333, emphasis removed) calls the “basic spatial relations between spaces.”

Part B, which generates the plan, comprises two grammars in parallel. The left-hand grammar takes as its initial shape the diagram generated in part A. The diagram remains essentially unchanged, but guides the right-hand grammar in generating the plan. Part A can generate many diagrams, any one of which can lead in turn to many plans.

4. Part A: setting the numbers of bays

Part A of the grammar (figure 1) uses the inverted-T algorithm seen in Stiny and Mitchell’s (1978) Palladian grammar: start with the front center bay, add bays in
width and in depth to form an inverted T of the desired size, and fill out the corresponding rectangular grids.

Rule 1 instantiates the front center bay at the point \((x_1, y_1)\). With this first bay come labels and counters:

- **Triangles**. These indicate where bays can be added.
- **Bay label**, set to \(c\). The grammar applies the appropriate label – \(c\), \(s_i\), or \(e\) – to bays when they are instantiated.
- **State label**, set to \(A\). This keeps track of where we are in the generation process. The state label has the value \(A\) in part A, the value \(B\) in part B.
- **Concentric square and line segment labels**. These mark progress through the diagram in part B.

Rules 2 and 3 increase the width by 2 bays; the new bays are labeled \(s_1, s_2, \ldots, s_n\). Rule 4 increases the depth by 1 bay. Rule 5 fills out the rectangular diagram 2 bays at a time. Rules 6 and 7 fix the width and depth of the diagram by changing the triangular labels at the ends of the inverted T to square ones; rules 2, 3, and 4 can no longer be applied. Rules 8 and 9 check that the diagram is well-formed, i.e., that it is rectangular. Rule 8 changes triangular labels to square ones. If the lines of square labels meet at corner bays, then the diagram is rectangular, and rule 9 changes the state label from \(A\) to \(B\), allowing it to be passed into part B.

5. Part B: setting the sizes of bays

Part B of the grammar (figure 2) receives as one of its two initial shapes the diagram generated in part A. The left-hand grammar walks us (with our position marked by the concentric square) through the dimensionless bays of the diagram. To ensure that we visit each bay, we use an algorithm that takes us from front to back in each “column” of bays, starting at the center column and working out towards the end columns.

As we visit each bay in the diagram, we instantiate the corresponding bay in the plan. Each new bay has the same position as its counterpart in the diagram, but now has dimensions; these are the dependent parameters. The diagram ensures that the plan has the right number of bays; the parametric constraints in the right-hand grammar ensure that the bays have appropriate dimensions. Rules 1, 2, and 3 instantiate the center bays of the plan. Rules 4, 5, and 6 instantiate the succeeding bays of the plan. Rules 7, 8, and 9 instantiate the end bays of the plan.

6. Example
Suppose that we want to generate a plan that meets the following criteria: the center bay has the maximum allowable width of 300 fen; the end bays have the minimum allowable width of 200 fen; and the succeeding bays are sized so that their widths change in equal increments. We first need to know how many bays in width there are in the diagram. Let us use part A of the grammar to generate a $5 \times 3$ diagram, derived as shown in figure 3. Now that we know there are 5 bays, we can easily calculate that they should become narrower, from center to end, in increments of 50 fen. We use part B of the grammar to generate another $5 \times 3$ diagram, but we instantiate the bays with the desired widths (figure 4). Rules B1, B4, and B7 instantiate the front center, front succeeding, and front end bays with widths of 300, 250, and 200 fen, respectively. As for the depth (200 fen in this example), we set it with rule B1, and it applies to all bays.

Figure 3. Derivation of the diagram.
Figure 4. Derivation of the plan.
Figure 4, continued. Derivation of the plan.
7. Conclusion

A parallel grammar accomplishes what a single grammar cannot: it keeps us from having to assign values to parameters that are dependent on other parameters whose values we have not yet assigned. It does this by using more than one product, and the parameters can be associated with one or another of them, depending on their relation. In our example, the number of bays in width and the number of bays in depth – the two independent parameters – are associated with the diagram, which is generated first, by part A of the grammar. Bay width, which is dependent on the number of bays in width, is associated with the plan, which is generated second, by the right-hand grammar in part B. The plan “inherits” the correct number of bays from the diagram.

Parallel grammars help us to express the relations among interdependent (as opposed to mutually independent) parameters. One common type of interdependence is hierarchical, where parameters range from the global to the local. In the vertical dimension of buildings, there are number of floors, floor-to-floor height, window height, and so on.

We can have any number of grammars in parallel, depending on how many objects are appropriate for the task. In the Yingzao fashi grammar, I anticipate two more grammars (parts C and D), which will generate in turn an orthogonal three-dimensional grid and an adjusted three-dimensional grid. These will set the values of the building’s global parameters, within which there will be many local variations.

The diagram is a control structure for the generation of the plan. The idea that a diagram can control a process is attractive indeed, and suggests that further investigations will be richly rewarded.

Acknowledgements

I would like to thank Professors George Stiny, Terry W. Knight, and William J. Mitchell for their guidance and encouragement.

References