A shape grammar for teaching the architectural style of the *Yingzao fashi*

Volume 1, Text

by Andrew I-kang Li

March 2001

A shape grammar for teaching the architectural style of the *Yingzao fashi*

by

Andrew I-kang Li

L.Mus., distinction in performance, McGill University, 1975
A.B. *cum laude* in East Asian languages and civilizations, Harvard College, 1977
M.Arch., Harvard University, 1983

Submitted to the Department of Architecture in partial fulfillment of the requirements for the degree of Doctor of Philosophy in Architecture: Design and Computation at the Massachusetts Institute of Technology

June 2001

Signature of Author: ______________________________
Department of Architecture
15 March 2001

Certified by: ______________________________
George Stiny
Professor of Design and Computation
Thesis Supervisor

Accepted by: ______________________________
Stanford O. Anderson
Professor of History and Architecture
Chairman, Committee for Graduate Students

Committee members

George Stiny, Professor of Design and Computation

Terry W. Knight, Associate Professor of Design and Computation

William J. Mitchell, Professor of Architecture and Media Arts and Sciences

A shape grammar for teaching the architectural style of the *Yingzao fashi*

by

Andrew I-kang Li

Submitted to the Department of Architecture
on 15 March 2001
in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in Architecture: Design and Computation

Abstract

The *Yingzao fashi* [Building standards] is a Chinese building manual written by Li Jie (d. 1110) and published in 1103. I present a shape grammar for teaching the architectural style – the language of designs – described in this manual. This grammar is distinguished by two objectives, and the technical means used to accomplish them.

First, the grammar is for teaching. Usually, the author of a grammar of a style aims to generate *all and only* the designs in the language. To do this, he not only writes the grammar, but also judges whether the designs it generates are members of the language. In the *Yingzao fashi* grammar, on the other hand, I want to generate all and *more than* the designs in the language. It is then the student who evaluates the designs – does this design belong to the language? – and adjusts the grammar accordingly. Thus the student participates actively in defining the language of designs, and learns that style is a human construct.

Second, the grammar is designerly. As already observed, most authors of style grammars focus on the language of designs; they do not consider how to structure the user's interaction with the grammar. By contrast, I consider explicitly what the user decides and when he decides it, and organize the grammar accordingly. In other words, I consider process as well as products.

The grammar exploits several technical devices for the first time: the design as an *n*-tuple of drawings, descriptions, and other elements; the generation of descriptions in the *n*-tuple; and techniques that are made possible by these devices.

Thesis supervisor: George Stiny
Title: Professor of Design and Computation

Biographical note

Andrew I-kang Li has been teaching since 1991 in the Department of Architecture, The Chinese University of Hong Kong, where he is now an associate professor. Before that, he taught in the Department of Architecture, Tunghai University, Taichung, Taiwan (1988–1991), and practiced as an architect at Architectural Resources Cambridge, Cambridge, Mass. (1983–1984, 1986–1988). He has degrees in music (L.Mus., distinction in performance, McGill University, 1973–1975), Chinese (A.B. *cum laude* in East Asian languages and civilizations, Harvard College, 1975–1977), and architecture (M.Arch., Harvard University, 1979–1983), and has pursued graduate studies in Chinese architecture (Nanjing Institute of Technology, now Southeast University, Nanjing, China, 1985–1986).

Contents

	7	Introduction
Chapter 1	17	The language of the *Yingzao fashi*
Chapter 2	23	Creating the plan diagram subdesign
Chapter 3	29	Creating the section diagram subdesign
Chapter 4	37	Creating the plan subdesign
Chapter 5	41	Creating the partial elevation subdesign
Chapter 6	43	Completing the design
	47	Discussion
	51	References

Introduction

THE *YINGZAO FASHI*

The *Yingzao fashi* [Building standards] was written by Li Jie[1] (d. 1110), court architect during the late Northern Song dynasty (960–1127), and published in 1103. Li evidently meant to educate government officials who commissioned buildings and to set standards for the builders who built them. He sets out rules for designing foundations, masonry buildings, wood-frame buildings (*da muzuo,* or structural carpentry), finish carpentry (*xiao muzuo*), and painted decoration. He also defines terms and provides methods for estimating materials and labor. The book includes numerous drawings, but these reflect a much later style – probably Ming (1368–1644) or Qing (1644–1911) – and so can be used as references for the Song only with caution.

In the classical Chinese literature, the *Yingzao fashi* is one of only two books that deal with architecture; the other is the *Gongcheng zuofa zeli* [Structural regulations], published in 1733. Architecture – or, perhaps more properly, building – was not an appropriate subject for literati, so the books are important simply for existing. However, they are interesting on their own account, since they document what had developed as, and probably still was, an oral tradition of structural carpentry.[2] In the case of the *Yingzao fashi,* that tradition was rule-based and parametric.

The *Yingzao fashi* has played a uniquely important role in the study of Chinese architectural history. Its discovery and reprinting in 1919 led directly to the establishment of the field, headed by Liang Sicheng (in Wade-Giles romanization, Liang Ssu-ch'eng, 1901–1972). He was the first to study the book; indeed, he organized his whole career around it. This in turn set the agenda for the field for decades to follow. At first, Liang found the book opaque, particularly the terminology. So he took an indirect route, and studied the *Gongcheng zuofa zeli.* Since this book was 600 years closer in time, more relevant buildings had survived, and there were even living craftsmen who knew the tradition. As he put it,

> [w]ith the [*Gongcheng zuofa zeli*] as the textbook, the carpenters as teachers, and the [Qing] palaces in [Beijing] as teaching material, the study of the methods and rules of [Qing] architecture began to have a solid basis (Fairbank 1994, 52; original Chinese at Liang 1984, 357).

---

1 For biographical information, see Glahn (1976) or Li Mingzhong (1930).

2 The main features of this tradition can be traced back at least to the Warring States period (475–221 BC), and are generally consistent with even earlier remains.

At the same time, Liang did inspired fieldwork and discovered numerous buildings related to the style of the *Yingzao fashi.* He eventually deciphered most of the text and wrote his magnum opus, the *Yingzao fashi zhushi* [The annotated *Yingzao fashi*] (Liang 1983), published posthumously in 1983.[3]

Chen Mingda (Ch'en Ming-ta, 1914–1997) had been Liang's assistant and continued Liang's work on the *Yingzao fashi* (Chen 1993). Chen perceived implicitly that the system of the *Yingzao fashi* was parametric and rule-based (although he did not use those terms), and strove to interpret the book in that way. Let us take as an example the modular unit *fen.* It can have eight different values, from 9.6 mm to 19.2 mm, depending on the grade (*deng*) of the building. So, for example, a modular dimension of 10 *fen* has eight possible absolute values, ranging from 3 *cun* (96 mm) at the eighth grade to 6 *cun* (192 mm) at the first grade. Li Jie stresses that the *fen* is fundamental and usually gives dimensions in *fen.* The user applies the relevant grade by using the appropriate scale or ruler, and determines the absolute dimension. But sometimes Li gives an absolute dimension instead. He writes, for instance, that rafters (in horizontal projection) should not be longer than 6 *chi* = 60 *cun* = 1,920 mm. Does he mean that rafters should never be longer than 60 *cun,* regardless of the grade of the building? This would seem to contradict his general rule. Or does the length of rafters vary with the grade? If so, then what grade does he have in mind: 60 *cun* is 200 *fen* at the eighth grade and 100 *fen* at the first. Chen (1993, 6–11) assumed the latter, studied extant buildings,[4] and concluded that Li probably meant the sixth grade, at which 60 *cun* = 150 *fen.* Thus Chen's contribution was to restate the *Yingzao fashi* in a consistently parametric way.[5]

However, many people still think of the *Yingzao fashi* as a laundry list of obscure names like *mud coat arm* (*nidao gong*). Thus, the next step is to demonstrate how the parameters and their constraints are related. In my teaching, I have taken such an approach, with gratifying results (Li and Tsou 1996). Here I undertake to characterize rigorously and in a unified way the language of designs defined by the structural carpentry system of the *Yingzao fashi.* My tool is shape grammar.

---

3 For a memoir of Liang, see Fairbank (1994).

4 The buildings that are clearly related to the Song style date from the Tang (618–907) through the Yuan (1271–1368) and into the first years of the Ming (1368–1644). Chen's (1992) list of 47 buildings, which includes dimensions and other valuable information, stops at the Yuan. Liang (1984) includes 3 more, going as late as 1391.

5 For a general introduction in English to the *Yingzao fashi,* see Liang (1984, 14–21) or Glahn (1984). Guo (1999) offers a translation but little explanation of chapters 4 and 5 on structural carpentry. The authoritative studies, in Chinese, are Liang (1983) and Chen (1993).

DEFINING LANGUAGES OF DESIGNS

Shape grammars have been used in many studies of languages of designs, from Palladian villas (Stiny and Mitchell 1978) to Wrightian prairie houses (Koning and Eizenberg 1981), from traditional Taiwanese houses (Chiou and Krishnamurti 1995) to Hepplewhite chair backs (Knight 1980). Their purpose is to elucidate the languages by articulating complete generative definitions of those languages. A complete definition specifies all and only the designs in the language. A generative definition specifies the member designs by generating them; this contrasts with an enumerative definition, which lists the member designs.

Stiny and Mitchell (1978) propose three criteria for evaluating a characterization of a language of designs:

1 It should specify new designs in the language;
2 It should evaluate whether a newly obtained design is a member of the language; and
3 It should explain the perceived likeness of the designs.

As Stiny and Mitchell point out, a complete generative definition meets these criteria. First, it can generate all (and only) the designs in the language. Second, if a design can be generated by the definition, then it is a member of the language. And third, because the definition is generative, it is itself the explanation.

Contrast this with a complete enumerative definition – an exhaustive list. To specify a new design, pick one off the list. To judge whether a design is a member of the language, see whether it is on the list. However, the list has no explanatory ability.

When we think about making a complete definition, it is helpful to consider what information we have at hand. There are two basic possibilities, reflecting the distinction between generation and enumeration. The first is a partial enumeration, partial because it lists only but not all designs in the language. This partial enumeration is the *corpus* of designs. The other basic possibility is a partial generative definition, partial because it does not specify some designs in the language (not all), specifies some designs not in the language (not only), or both (not all and not only).

Most published analytical grammars are of the first type. Some rely additionally on limited generative evidence. The prairie house grammar, for example, refers to Wright's own design prescriptions (Koning and Eizenberg 1981). The grammar presented here belongs to the second type, relying almost exclusively on generative evidence. This evidence is incomplete and often ambiguous, and the language of designs it defines is necessarily incomplete. The task here is to complete the generative definition. I will have more to say about this shortly.

HUMANS AND LANGUAGES

I have sketched a framework of how, with only partial information, a language of designs can be completely defined. There is still another element of the framework to be discussed: the human. As Knight (1999–2000) points out, there may be more than one human role: the author and the user(s) of a grammar are not necessarily the same person. I see at least three roles.

First, when the source is a corpus of designs, there must be, as Stiny and Mitchell (1978) point out, some perceived likeness. That likeness is perceived by a human; without the human, there is no likeness. Which designs belong to the corpus, which do not, which are representative, which are exceptional, are all decided by a human.

Second, the grammar is a hypothesis, which is also a human construct. No human, no hypothesis. (However, with a grammar, the hypothesis can be formally – that is, objectively – articulated.)

Third, as with any hypothesis, the grammar's predictive power must be tested: does the newly generated design belong to the language? In design, the test is almost always the judgement of a human. Contrast this with scientists, who use experiments to manipulate nature into judging their hypotheses. Downing and Flemming (1981) once wished for the same, to discover a unknown but genuine member design. Linguists seek native speakers to judge whether new sentences are grammatical. Duarte's (1999) rare fortune is to have a native stylist to judge his grammatical predictions. At the other extreme is Flemming's (1981) study of Terragni's Casa Giuliani Frigerio. With a corpus of one, the grammar is only as reliable as the human who judges its designs.

Traditional analytical grammars have not emphasized these human roles. The authors have played all three roles: they perceive the likeness, they write the grammar, and they judge the results. This authoritative approach is reasonable, since they are striving for the completeness of the definition. In this project, I take a different approach: I use a grammar to define two authors and a judge. Again, more on this shortly.

HUMANS AND GRAMMARS

Humans have another role, as users of grammars. Unless it is entirely deterministic, a grammar needs the user to choose which rule to apply, under which transformation or assignment. In effect, it asks the user to make certain decisions in a certain sequence, and if those decisions or sequences do not make sense to a user, then the grammar may surprise and even disappoint. It may, for instance, generate designs that do not fulfill the requirements of a given problem, even when those requirements are known and articulated at the beginning of the computation. This issue is worth examining. I present three cases of fitting a building onto a given site: two examples and one counterexample.

The Queen Anne grammar

Take for example Flemming's Queen Anne grammar, which generates plans by instantiating rooms around an initial entrance hall. It does not handle the dimensions of rooms, but Flemming asserts that it is "rather trivial" to parameterize the grammar to assure that "the layout under development actually fits within the given boundaries and, possibly, keeps desired setbacks" (Flemming 1987, 333).

In fact, the grammar needs more than parameterization. The user of the grammar does not know whether the plan fits on the site until he has finished the computation. To put it another way, it makes no difference that the user knows the size of the site before beginning the computation, because the grammar does not show him how to use that knowledge to set the room sizes appropriately. Instead, the grammar requires him to set the room sizes *before* he has finished the computation. Generating a plan that fits on the site is a matter of trial and error.

Why is this so? Consider the plan in width. It embodies two parameters: the number n of rooms and the widths r_i of rooms, where $1 \le i \le n$. These determine the width b of the building, which is the sum of the widths of the rooms:

$$b = r_1 + r_2 + \ldots + r_n.$$

If the building is to fit on the site, then it must be no wider than the buildable site, s: $b \le s$. the relation among s, n, and r_i is this:

$$r_1 + r_2 + \ldots + r_n \le s. \tag{1}$$

For any s, n is constrained by r_i, and r_i is constrained by n; we will say that n and r_i are mutually constrained.[6] This is because they are embodied in a single object, the plan. In addition, they are manipulated simultaneously: a rule that affects r_i also affects n. Now even if the user knows s, he does not know how to use s to choose – simultaneously – appropriate values for n and r_i. The result it is that he can only generate and test: he uses the grammar to generate a plan with parameters n and r_i, and tests it against s.

The Mughul garden grammar

Like the Queen Anne grammar, the grammar of paradise (Stiny and Mitchell 1980) involves mutually constrained parameters, although it does not claim to fit a plan to a site. This grammar generates Mughul garden plans by recursively subdividing a square into four subsquares. With

---

6 I say *mutually constrained* rather than *interdependent*, which might suggest that the value of each parameter is *determined* by the value of the other. The values are not determined each by the other, but merely constrained. One could also say that it is the parameters' constraints that are determined, not their values.

each subdivision, an optional border can be added inside the perimeter of the square, if a border was added at the previous subdivision. Each successive border is either always narrower or always wider than the previous border.

If the borders are successively narrower, they could eventually become “too narrow.” No minimum width is given, but we will assume one, since a border, to exist, must have nonzero width. If, on the other hand, the borders are successively wider, then they could become wider than the later, smaller subsquares; some subsquares could, in effect, be all border and no garden. Whether this should be permitted is a question of interpretation for the authors of the grammar. However, we will assume that the widest border must be narrower than the narrowest subsquare, both for the sake of argument and because the examples given in the article appear to follow this constraint.

Let us formalize these constraints. Let n be the number of subdivisions; s_n, the width of the subsquares formed by the nth subdivision; b_i, the width of the ith border, where $1 \le i \le n$; $b_{\min}$, the minimum width of a border; and $b_{\max}$, the maximum width of a border, where $b_{\max} < s_n$. We have, for successively narrower borders,

$$b_{\max} \ge b_1 > b_2 > \ldots > b_n \ge b_{\min}, \tag{2}$$

and, for successively wider borders,

$$b_{\min} \le b_1 < b_2 < \ldots < b_n \le b_{\max}. \tag{3}$$

The successive values of b_i must be chosen so that, in the first case, $b_1 \le b_{\max}$ and $b_n \ge b_{\min}$ or, in the second case, $b_1 \ge b_{\min}$ (which is trivial) and $b_n \le b_{\max}$. As in the Queen Anne grammar (1), the successive parametric values have known constraints. But, where s in the Queen Anne grammar is fixed, $b_{\max}$ in the Mughul grammar is a moving target: it becomes smaller with every successive subdivision.

The Taiwanese house grammar

Now take as a counterexample the traditional Taiwanese house grammar (Chiou and Krishnamurti 1995). Like the Queen Anne grammar, it generates plans by incrementally adding rooms but, unlike the Queen Anne grammar, it does not involve mutually constrained parameters.

The Taiwanese grammar first instantiates a center room and then adds two rooms at a time: one on the left, one on the right. The number n of rooms is odd and may not exceed 9: $n = 2m + 1$, where $0 \le m \le 4$. The width r_0 of the initial room is subject to constraints that are not explicitly given. The width r_i of each succeeding room, where $1 \le i \le m$, is constrained by the width r_{i-1} of the previous room; it is narrower by a decrement factor d_i ranging from 0.8 to 0.9: $r_i = d_i r_{i-1}$, where $0.8 \le d_i \le 0.9$ and $1 \le i \le m$.

However, there are two differences between the Queen Anne and the Taiwanese grammars. First, in the Queen Anne grammar, the computation itself can be seen largely as a recursive process for setting the number n of rooms. The Taiwanese grammar, on the other hand, sets n in a single step: rule 1 instantiates the scale bar, ΠΠ, which encodes n graphically.

Second, and more important, the parameters n and r_i are not mutually constrained: r_i is constrained by r_0, d_i, and n, but not vice versa. There is no s or $b_{\max}$. Thus, for any legal values of r_0, d_i, and n, we expect, since there is no statement to the contrary, that r_i will also be legal; it should not be "too narrow." For example, the narrowest possible room will be the outermost room in that house which has the narrowest initial room, the greatest number of rooms ($n = 9$), and the greatest decrement in width ($d_1 = d_2 = d_3 = d_4 = 0.8$); it will be about four-tenths as wide as the initial room:

$$r_4 = d_1 d_2 d_3 d_4 r_0 = 0.8 \times 0.8 \times 0.8 \times 0.8 \times r_0 = 0.4096 r_0.$$

If $r_4 = 0.4096r_0$ is found to be too small, even for a legal value of r_0, then a new condition has been introduced, one which causes r_0, d_i, and n, on the one hand, and r_0, on the other, to be mutually constrained. However, since there is no indication that this is, or even should be, the case, it seems fair to assume that $r_4 = 0.4096r_0$ is a legal value. The parameters are simply, not mutually, constrained.

¬ ¬ ¬

The three grammars discussed here show why it is sometimes difficult to generate objects that meet the requirements of a given problem. In the Queen Anne and Mughul grammars, some parameters are mutually constrained. Users use these grammars to generate designs and test them against the requirements of the problem – whether to fit the building to the site or to create borders that are neither too narrow nor too wide – with no guarantee of success. By contrast, the Taiwanese grammar, which has simply constrained parameters, does not have this difficulty.

The problem arises because the grammar asks the user to make design decisions in a foreign way. The solution is straightforward, at least to understand: the grammar should be constructed so the user – the designer – makes decisions in a way that makes sense to him. We can call this accommodation *designerliness.*

THE GRAMMAR OF THE *YINGZAO FASHI*

To investigate the ways that a human interacts with a grammar, I here propose a project in which I define clear and essential roles for people who write and use the grammar. The project is a teaching grammar of the *Yingzao fashi.*

In the light of the discussion above, this project has three salient features. First, the source – the text of the *Yingzao fashi* – is not primarily an enumeration, but a partial generative definition. There are extant buildings, of course, but they are few (not more than 50), and none follows all the rules. My approach is, to the extent possible, to follow the generative definition, the text. I avoid the enumerative definition – the corpus of extant buildings, however defined – which consists of imperfect examples, at least as far as the *Yingzao fashi* is concerned. Nevertheless, I rely on Chen's hypothetical rules. I take this approach because I do not aim for a complete definition; instead, I want to characterize the text as it is, sometimes vague, sometimes silent. And the reason for this is the second salient feature.

This grammar is for teaching. My goal, as teacher and author of the grammar, is to engineer a useful experience for the student learning about the style of the *Yingzao fashi.* I believe that the most useful such experience is to participate in composing and testing the hypothetical definition. To do this, I write the grammar so that it generates all and not only (in other words, more than) the designs that are likely to be in the language. As the grammar reflects the imperfections of the text, so do its products, which the student can evaluate.

This differs from the usual analytical approach, in which the author is also the judge, because he is aiming for an authoritative definition. We might call this the expert approach. The advantage of my approach is that the student, not the teacher, aims for the authoritative definition. I give him no more information than there already is, so he must confront the gap between what he knows and what he needs to know. What information is missing, and why? Was it knowledge common to Li Jie and his readers, but now lost to us? Was it overlooked by Li? Was it specialized builder's knowledge that, whether by design or by ignorance, Li omitted? What assumptions are needed, and are they justified? We might call this the naïve approach.

The third salient feature is that the grammar is designerly. I try to organize the user's interaction with the grammar so that it seems natural to him.[7] Not only do I eliminate difficulties like those associated with mutually constrained parameters, but I give the user – the designer, really – the sense he is designing a building. In this way, the grammar is more of a guide for the designer than a machine for producing designs. This is a subjective standard, but appealing to the user is surely one of the goals of shape grammars: Stiny (1977) calls his ice-ray grammar "simple and intuitively appealing."

7 Knight (1999–2000) discusses the user's possible roles with respect to a grammar.

To achieve designerliness, I expand the grammar in two dimensions. The first is in the number of elements that make up the design: seven drawings and nine descriptions. The second is in the computation: the user creates the 16-part design not in a single step, but in seven steps. The technical devices have long been available: multipart designs, descriptions, and algebras combined in cartesian products (Stiny 1981, 1990, 1992).

Chapter 1

The language of the *Yingzao fashi*

The design and its elements

The designs have the form proposed by Stiny (1990, 97): a design λ is a member of "an n-ary relation among drawings, other kinds of descriptions, and correlative devices as needed." Here I employ 16 elements – 7 drawings and 9 descriptions (figures 1a, 1b):

o, plan diagram;
r, section diagram;
p, plan[8];
e, partial elevation;
d, roof section;
s, section;
f, (complete) elevation;
u, number of bays (= width of diagram, in bays);
v, number of rafters (= depth of diagram, in rafters);
w, number of storeys (= height of diagram, in storeys);
b, disposition of beams;
c, number of columns in depth;
$x = (\xi_1, \xi_2, \ldots, \xi_m)$, widths (in *fen*[9]) of bays, $m = (u + 1) / 2$;
y, length (in *fen*) of rafters;
z, height (in *fen*) of columns; and
$l = (l_1, l_2, \ldots, l_n)$, elevations (in *fen*) of purlins, $n = v / 2$.

I distinguish the three sets Λ'', Λ', and Λ.

8 I distinguish *diagrams* and *scale drawings.* Thus, for example, the *plan diagram* records only the numbers of bays and rafters but not their lengths; the *plan* records both. For the sake of brevity, I omit *scale* wherever clarity permits.

9 The *fen* is a modular unit of length. Its absolute value depends on the grade or rank (*deng*) of the building. There are 8 grades of building and 8 values of the *fen,* ranging from 9.6 to 19.2 mm. For the sake of simplicity, I take the *fen* as an absolute unit.

- Λ'' is the language of legal *Yingzao fashi*-style designs. It contains all and only such designs, and is a subset of …

- … Λ', the language of well-formed designs. The drawings and descriptions of a well-formed design "match." So, for example, a design λ_1 with a 7×6 plan diagram and descriptions $u_1 = 7$ and $v_1 = 6$ is a well-formed design, and is a member of Λ'. A "design" λ_2 with the same plan diagram and descriptions $u_2 = 9$ and $v_2 = 4$ is not well-formed, and is not a member of Λ'. In fact, λ_2 is not even a design, only a 16-tuple. Λ' contains all but not only legal designs, and is a subset of …

- … Λ, which is the cartesian product $O \times R \times P \times E \times D \times S \times F \times U \times V \times W \times B \times C \times X \times Y \times Z \times L$, where O, R, …, L are respectively the sets of values of o, r, …, l. Not all 16-tuples in Λ are designs, so Λ is not a language. It is of interest to us only insofar as it is defined by O, R, …, L, which must be large enough to define a useful Λ'.

The distinction between Λ' and Λ'' allows the student to have a role in defining Λ''. I write the grammar so that it reflects the imperfections of the text and thus specifies an imperfect version of Λ'', namely Λ', which I call the student's *working language.* The student, by evaluating the designs generated by the grammar, tries to specify Λ'', the *target language.* He can do so generatively, by refining the grammar. In particular, he can define or modify constraints on or among the 16 elements of the design. This contrasts with the expert approach, in which there is no distinction between the language specified by the grammar and the language under study.

The design and its subdesigns

I have shown how Stiny's definition can combine 16 elements into a design. We can understand this definition recursively: a design is a member of an *n*-ary relation among drawings, other kinds of descriptions, correlative devices as needed, *and other designs.* In my case, the 16-part design λ is then reconceived as a member of a relation among the following seven subdesigns:

ω, plan diagram;
ρ, section diagram;
π, plan;
ε, partial elevation;
δ, roof section;
σ, section; and
ϕ, (complete) elevation.

Each of the 7 subdesigns is a member of an m-ary relation among one drawing (after which it is named[10]) and $m - 1$ descriptions. Thus, for example, the plan diagram subdesign ω is an element in a ternary relation among the plan diagram (drawing) o, the number u of bays, and the number v of rafters. If we specify the subdesigns ω, ρ, π, ε, δ, σ, and ϕ, then we specify the design λ.[11]

As each subdesign κ is to the design λ, so are Κ, Κ′, and Κ″ to Λ, Λ′, and Λ″. Κ is the cartesian product of the sets of values of the constituent elements of κ. Κ′ is the student's working sublanguage, consisting of all and not only legal subdesigns. And Κ″ is the student's target sublanguage, consisting of all and only legal subdesigns.

The subdesigns ω, ρ, π, ε, δ, σ, and ϕ and the design λ are shown here with their constituent elements.

ω	*o*	·	·	·	·	·	·	*u*	*v*	·	·	·	·	·	·	·
ρ	·	*r*	·	·	·	·	·	·	*v*	*w*	*b*	*c*	·	·	·	·
π	·	·	*p*	·	·	·	·	*u*	*v*	·	*b*	*c*	*x*	*y*	·	·
ε	·	·	·	*e*	·	·	·	*u*	·	*w*	·	·	*x*	·	*z*	·
δ	·	·	·	·	*d*	·	·	·	*v*	·	·	·	·	*y*	·	*l*
σ	·	·	·	·	·	*s*	·	·	*v*	*w*	*b*	*c*	·	*y*	*z*	*l*
ϕ	·	·	·	·	·	·	*f*	*u*	·	*w*	·	·	*x*	·	*z*	·
λ	*o*	*r*	*p*	*e*	*d*	*s*	*f*	*u*	*v*	*w*	*b*	*c*	*x*	*y*	*z*	*l*

Specifying the subdesigns rather than the design provides a way of combining and separating elements as desired, which corresponds nicely to the way that a designer reduces his problem. He uses representations that contain information in only two of three dimensions, but he can choose among them in order to combine some elements (and exclude others) or separate some elements. So, for example, when he considers a building layout, he wants to include information in the x- and y-dimensions (like the location of a wall) and exclude information in the z-dimension (like the height of the wall); he uses a plan as his primary tool. And when parameters are mutually constrained, he wants to separate them; this is why I have two plans and two sections.

Using subdesigns expands both the design and the computation (the process). This is clear when we look at other grammars. Stiny's (1992) example of algebras combined in cartesian products consists of three drawings (a plan and two elevations), but these are generated simultaneously. They are equally important; there is no hierarchy. Flemming's (1987) Queen Anne grammar generates two drawings: first a plan, then a three-dimensional view. But the first

10 I omit the words *subdesign* and *drawing* where possible.
11 We might also distinguish drawings and descriptions, at least in this case. See note 13 below.

is transformed into the second, so they cannot exist simultaneously as complete, different representations of the design.

The two-dimensionally expanded approach presented here gives us an important way to accommodate the user and how he works. He creates the design by creating one subdesign at a time. When he creates a subdesign, he is not just specifying a member of the sublanguage; he is also defining in the working language an equivalence class of designs. As he creates each succeeding subdesign, he defines a smaller working language which ultimately contains a single design. He has now specified the design.

In greater detail, the process is this. First, the user creates the plan diagram subdesign $\omega_1 = \langle o_1, u_1, v_1 \rangle$. This defines in Λ' the equivalence class Q_1 of designs $\lambda = \langle o, u, \ldots, l \rangle$ for which $o = o_1$, $u = u_1$, and $v = v_1$; Q_1 is the user's new working language. Next, he creates the section diagram $\rho_2 = \langle r_2, v_1, w_2, b_2, c_2 \rangle$; he already has $v = v_1$, from the plan diagram subdesign. This defines in Q_1 the equivalence class Q_2 of designs for which $o = o_1$, $r = r_2$, $u = u_1$, $v = v_1$, $w = w_2$, $b = b_2$, and $c = c_2$. Similarly, he creates the plan subdesign $\pi_3 = \langle p_3, u_1, v_2, b_2, c_2, x_3, y_3 \rangle$, which defines Q_3; the partial elevation subdesign $\varepsilon_4 = \langle u_1, w_2, x_3, z_4 \rangle$, which defines Q_4; the roof section $\delta_5 = \langle v_2, y_1, l_5 \rangle$, which defines Q_5; the section $\sigma_5 = \langle v_1, w_2, b_2, c_2, y_1, z_4, l_5 \rangle$, which defines Q_6; and the complete elevation subdesign $\phi_7 = \langle f_7, u_1, w_2, x_3, z_4, l_5 \rangle$, which defines Q_7. This last equivalence class Q_7 contains the single design $\lambda = \langle o_1, r_2, p_3, e_4, d_5, s_6, f_7, u_1, v_1, w_2, b_2, c_2, x_3, y_3, z_4, l_5 \rangle$.

ω_1	o_1	·	·	·	·	·	·	u_1	v_1	·	·	·	·	·	·	·
ρ_2	·	r_2	·	·	·	·	·	·	v_1	w_2	b_2	c_2	·	·	·	·
π_3	·	·	p_3	·	·	·	·	u_1	v_1	·	b_2	c_2	x_3	y_3	·	·
ε_4	·	·	·	e_4	·	·	·	u_1	·	w_2	·	·	x_3	·	z_4	·
δ_5	·	·	·	·	d_5	·	·	·	v_1	·	·	·	·	y_3	·	l_5
σ_6	·	·	·	·	·	s_6	·	·	v_1	w_2	b_2	c_2	·	y_3	z_4	l_5
ϕ_7	·	·	·	·	·	·	f_7	u_1	·	w_2	·	·	x_3	·	z_4	·
λ	o_1	r_2	p_3	e_4	d_5	s_6	f_7	u_1	v_1	w_2	b_2	c_2	x_3	y_3	z_4	l_5

The grammar

There are seven stages, corresponding to the seven subdesigns. Each stage is deterministic or nondeterministic, according as whether or not, for any given input, there is only one possible output. The first four stages are nondeterministic; the user decides which subdesign is created. The last three stages are deterministic; given the subdesigns that the user has already created, each stage generates exactly one subdesign. Each stage consists of substages, each of which is similarly either deterministic or nondeterministic.[12]

There seems to be a general pattern. For any nondeterministic stage (or substage or even rule), there may be one deterministic stage (or substage or rule) preceding it and one following it. The first, deterministic, phase “reads in” values from another subdesign and instantiates them in the current subdesign; this phase *prepares* the design. An example is rules B1–B10, which prepare a section diagram with the same number of rafters as the plan diagram already created. In the second, nondeterministic, phase, the user chooses values for parameters, observing any relevant constraints; in this phase, he *designs* the output. For example, in stage D he applies the single rule D1 to decide the column height. In the third, deterministic, phase, the grammar fills in necessary information or transforms it into some required form; it *completes* the design. In the partial elevation subdesign, for example, rules D2–D4 complete the partial elevation drawing, based on the column height decided in rule D1.[13]

Example

As an example, we follow our hypothetical student as he creates a design as follows: the plan diagram is 5 bays × 6 rafters; the bays are 300, 250, and 200 *fen* wide; the rafters are 100 *fen* long; the columns are 200 *fen* high; 6-rafter building, centrally divided with a 1-rafter beam in front and in back, with 5 columns; and purlin elevations of 65, 50, and 35 *fen.* That is,

$\langle u, v, w, b, c, x, y, z, l\rangle =$ ⟨5, 6, 1, (6-rafter building, centrally divided with a 1-rafter beam in front and in back, with 5 columns), 5, (300, 250, 200), 100, 200, (65, 50, 35)⟩.

We see how, in each of the nondeterministic stages (A through D), he can generate a number of subdesigns, each of which defines a smaller equivalence class for the next stage. Some of the subdesigns may be illegal; the student decides.

Stage A (plan diagram subdesign) is the beginning of the computation, so the student is not constrained by any previous actions, and can design anything in the working sublanguage. He creates a 7 × 4 plan diagram, which probably is not legal. Instead, he creates a 5 × 6 plan diagram, which probably is.

12 This enables us to say that a stage with at least one nondeterministic substage is nondeterministic.

13 The distinction between deterministic and nondeterministic seems to reflect the difference between the current language of designs and the universe of shapes. For example, in the language of the *Yingzao fashi,* both a 7-tuple of drawings and a 9-tuple of descriptions can specify the same 16-element design. However, where the descriptions distinguish the design only within the language of the *Yingzao fashi,* the drawings distinguish it in the universe of, say, designs consisting of two-dimensional line drawings. The universe of shapes is larger and, importantly, is the one that designers work in. This would explain why there are so many more rules making up deterministic processes: they enrich the specification so that it makes sense in a larger frame of reference.

In stage B (section diagram subdesign), the student defines the disposition of beams (i.e., the spatial subdivision). Our knowledge about spatial subdivision comes from a very small corpus of descriptions, so the grammar is especially tentative, and the student's judgement is critical. A disposition of all 1-rafter beams fills the interior with columns and is probably not legal. He creates other, more plausible alternatives.

Stage C (plan subdesign) is more straightforward. The student instantiates the dimensions of the bays and the rafters, observing any constraints that follow from u and v in the plan diagram. Mutually constrained parameters are separated.

Stage D (partial elevation subdesign) is simpler still. The student creates the partial elevation drawing by setting the column height against the number and widths of bays.

Stages E, F, and G (roof section, section, and complete elevation subdesigns) are deterministic. The student makes no decisions; he follows instructions, as it were. The result is one design, with all 16 elements specified.

Chapter 2

Creating the plan diagram subdesign

The sublanguage of plan diagram subdesigns

The plan diagram subdesign ω is a member of a ternary relation among the plan diagram (drawing) o, the number u of bays (width of diagram), and the number v of rafters (depth of diagram).

I distinguish Ω, Ω', and Ω'', which are respectively the cartesian product $U \times V \times W$, the language of well-formed plan diagram subdesigns, and the language of legal plan diagram subdesigns. These are analogous to Λ, Λ', and Λ''.

What we know

The first question is: what are legal values of $u \in U$ and $v \in V$? The author Li Jie said nothing, probably because he thought it too obvious to mention. Indeed, all the available evidence suggests that u is odd and v is even, and few people would suggest it could be otherwise. I concur.

Next, what are the limits of u and v? Again, Li said nothing. Chen (1993, 31), playing the role of the historian, concludes that $u \leq 9$ and $v \leq 10$. I propose $u \geq 5$ and $v \geq 4$, with no upper limit. (The high lower limits simplify the grammar; to get $1 \leq u < 5$ and $2 \leq v < 4$, more rules should be added in stage C.)

Finally, are all $\langle u, v \rangle \in U \times V$ legal? To this question, the historian would offer an answer. As a teacher, however, I want my student to be the one to attempt an answer. Again, Li said nothing directly, but there is information, in the form of incomplete enumerations, that is probably relevant. I summarize it in four charts (figure 2), and the student can evaluate it himself. The diagrams show:

1 Textual references to plan diagram sizes of *dian* halls[14] (upper left). These references indirectly sanction certain plan diagram sizes; they are a partial enumeration (only but not all). For example: "For a hip-roofed *dian* hall of 4 or 6 rafters by 5 bays, or of 10

14 The *dian* hall is a higher-ranked type than the *ting* hall, which is the subject here.

rafters by 9 bays, connect the corner beam to the ridge purlin" (Liang 1983, 159). There are no such references for *ting* halls (upper right).

2 The plan diagram sizes of extant *dian* and *ting* halls (black dots) (Chen 1992). The match between buildings and text is not perfect, which could mean that the text was sometimes reinterpreted, ignored, or unknown by the builders. This should not be surprising, since the extant buildings range over a period of more than 500 years and a similarly large geographical area.

3 An assumption that we might make about the sublanguage of legal *dian* hall plan diagram sizes, that is, a hypothetical complete enumeration (all and only) (lower left).

4 An assumption that we might make about the sublanguage Ω'' of legal *ting* hall plan diagrams, extrapolating from the comparable sublanguage of *dian* hall plans (lower left). Again, this is a hypothetical complete enumeration (all and only). The student is free to reach a different conclusion.

With this information, which is about all the information there is, the student can decide what size plan diagram subdesign ($\omega \in \Omega'$) to create and evaluate ($\omega \in \Omega''$?). He creates $\omega_1 = \langle o_1, u_1, v_1 \rangle$, which defines in the working language Λ' the equivalence class of designs λ for which $o = o_1$, $u = u_1$, and $v = v_1$. This is the new working language Λ'_1.

The plan diagram vs. the scale plan

There are two mutually constrained parameters in the x-dimension: the number u of bays and the width x of bays.

1 **Number *u* of bays**: $u = 2m - 1$, $m \geq 3$.

2 **Width *x*** (in *fen*) **of bays**, $x = (\xi_1, \xi_2, \ldots, \xi_m)$. Bays are at least 200 *fen* and at most 300 *fen* wide. A bay may not be wider than the preceding bay (the one on the inside).[15]

That is,

$$300\,fen \geq \xi_1 \geq \xi_2 \geq \ldots \geq \xi_m \geq 200\,fen,\ m \geq 3.$$

Compare this with the constraints of the Mughul grammar:

$$b_{\max} > b_1 > b_2 > \ldots > b_n > b_{\min}. \quad (2)$$

15 For more details, see stage C.

The maximum width in the *Yingzao fashi* is fixed (at 300 *fen*), while the Mughul maximum changes during the computation. However, the overall forms of the constraints are identical, and the two parameters u and x of width in the *Yingzao fashi* are mutually constrained. So we set u and x separately. First, in stage A, we set u, as part of the plan diagram subdesign ω. Then, in stage C, we simultaneously "read" u from the plan diagram and set x, as part of the plan subdesign π.[16]

The algorithm

Stage A is nonparametric. It uses the inverted-T algorithm used by Stiny and Mitchell (1978) in their Palladian grammar: it generates first the front center cell, adds cells in width and in depth to form an inverted T of the desired size, and fills out the corresponding rectangular diagram. The cells have not yet acquired their final dimensions ξ_i and y; they are all of a single standard size.

The initial shape (figure 3) is a point (α_1, β_1) on an axis parallel to the y-axis, with the indices $i = 0$ and $j = 0$, which count the bays and the rafters respectively, and assign the values of u and v. Rule A1 instantiates the front center bay at the point (α_1, β_1) and increments i by 1 and j by 2. We use the following labels and markers:

1 **Expansion markers** (triangles) and **infill markers** (long line segments), which indicate where cells can be added, either at the ends of the inverted T (triangles) or at the inside corners (long line segments).

2 **State label**. This ensures that only rules belonging to the current stage are applied. The state label is the symbol A in stage A, the symbol B in stage B, and so on.

3 **Cell marker** (square) and **rafter and bay markers** (short line segments), which are not used until stages B and C, where they "walk" through the plan diagram.

Rules A2 and A3 increase the width and depth of the plan diagram, respectively, and increment the appropriate index. Rule A4 fills out the diagram two bays at a time. Rules A5 and A6 fix the width and depth at $u = i$ and $v = j$, respectively; rules A2 and A3 may no longer be applied. Rules A7 and A8 check that the diagram is rectangular in the following way. Rule A7 changes infill markers to square markers. If the lines of square markers meet at corner cells, then the

16 In other circumstances, we might want to set x before u. If, for example, we were designing a hotel, with many identical rooms, we might first design the individual room (i.e., set x), and then decide the number of rooms to fit on the site (set u). Prof. George Stiny suggested the two-step, two-drawing approach.

diagram is rectangular, and rule A8 may be applied to change the state label from A to B. The diagram is passed into stage B.

Examples

We return to our student, who creates two plan diagram subdesigns, one probably illegal. To judge from the chart (figure 2), a 7 × 4 plan diagram is probably illegal, but the student is free to create it anyway (figure 4). A more probably legal plan diagram is 5 × 6 (figure 5). In each case, seeing the plan diagram o may help the student evaluate whether it is legal or not; for a designer, seeing the drawing is likely to be more useful than contemplating the parameters u and v. Also, as the student considers whether each subdesign is legal, he can see the relation Ω'' take shape on the chart.

The plan diagram subdesign defines the equivalence class of designs λ with, in the first case, a 7 × 4 plan diagram o, $u = 7$, and $v = 4$, or, in the second, a 5 × 6 plan diagram o, $u = 5$, and $v = 6$. Which subdesign he selects now determines the equivalence class of designs he ultimately chooses from. Let us assume that he chooses the second.

Let us see how separating the mutually constrained parameters u and x solves the problem discussed above. Suppose that the student wants to create a plan that meets the following criteria: the center bay has the maximum allowable width; the end bays have the minimum allowable width; and the intermediate bays have widths that change in equal increments; that is,

$$\begin{aligned} &\xi_1 = 300\,\textit{fen}, \\ &\xi_m = 200\,\textit{fen}, \text{ and} \\ &\xi_{i+2} - \xi_{i+1} = \xi_{i+1} - \xi_i, \text{ where } 1 \le i \le m - 2. \end{aligned} \tag{4}$$

First he needs to know how many bays, $u = 2m - 1$, there are in the plan diagram. He does not need to know the value of u until he comes to rule A5, which asks in essence: is this the number of bays that you want? If the answer is no, he can ignore rule A5 and apply rule A2 to add 2 more bays. If the answer is yes, then he applies rule A5 to fix the width of the plan diagram at $u = i$.

He has set the diagram width ($u = 5$ or $m = 3$), so he can easily calculate that the bays should become narrower, from center to end, in decrements of 50 *fen.* That is, he satisfies the constraints (4) like this:

$$\begin{aligned} &\xi_1 = 300\,\textit{fen} \\ \ge\ &\xi_2 = 250\,\textit{fen} \\ \ge\ &\xi_3 = 200\,\textit{fen}. \end{aligned}$$

This information he leaves until stage C, when he creates the scale plan. But the plan diagram drawing itself he will use twice more, to guide him in generating the section diagram, in stage B, and then the scale plan, in stage C.

Chapter 3

Creating the section diagram subdesign

The sublanguage of section diagram subdesigns

The section diagram subdesign ρ is a member of a 5-ary relation among one drawing (the section diagram drawing *r*) and four descriptions (the depth *v* of the diagram, the height *w* of the diagram, the disposition *b* of beams, and the number *c* of columns).

We distinguish $\underline{\mathrm{P}}$, $\underline{\mathrm{P}}'$, and $\underline{\mathrm{P}}''$, which are respectively the cartesian product $R \times V \times W \times B \times C$, the language of well-formed section diagram subdesigns, and the language of legal section diagram subdesigns.[17] These are analogous to Λ, Λ′, and Λ″.

In stage A, the student created the plan diagram subdesign $\omega_1 = \langle o_1, u_1, v_1 \rangle$. This defines $\underline{\mathrm{P}}_1$, $\underline{\mathrm{P}}'_1$, and $\underline{\mathrm{P}}''_1$ as follows:

$\underline{\mathrm{P}}_1 = R \times V_1 \times W \times B \times C$, where $V_1 = \{v_1\}$,
$\underline{\mathrm{P}}'_1 = \underline{\mathrm{P}}' \cap \underline{\mathrm{P}}_1$, and
$\underline{\mathrm{P}}''_1 \subseteq \underline{\mathrm{P}}'_1$.

The sublanguage $\underline{\mathrm{P}}'_1$ of well-formed section diagram subdesigns is the equivalence class in $\underline{\mathrm{P}}'$ of section diagram subdesigns with $v = v_1$; it is the student's current working sublanguage. The sublanguage $\underline{\mathrm{P}}''_1$ of legal section diagram subdesigns is a subset of $\underline{\mathrm{P}}'_1$; it is the student's current target sublanguage. By reducing $\underline{\mathrm{P}}'$ to $\underline{\mathrm{P}}'_1$, the student reduces his problem.

What we know

The structural frame of a *ting* hall is composed of repeated transverse frames (*liangjia*) perpendicular to the front elevation. Each of these transverse frames is composed in turn of columns (*zhu*) and transverse beams (*fu*). The *Yingzao fashi* shows 18 transverse frames drawn in section, each with a terse written description (figure 6) (Liang 1983, 313–321). This description is the disposition *b* of beams, and for us is no mere counter of features. As we will see, it offers insight into the definition of the sublanguage $\underline{\mathrm{P}}''$ of legal section diagram subdesigns.

---

17 Distinguish $\underline{\mathrm{P}}$ and P. $\underline{\mathrm{P}}$ is the set related to the plan diagram subdesign ρ; P is the set related to the plan diagram (drawing) *p*.

It is well to remember that, unlike western buildings, in which the rafters support the purlins, in Chinese buildings the purlins (*tuan*) support the rafters (*chuan*). The rafters are segmented and make possible the characteristic curved roof section. A rafter is not more than about 1.80 meters long in horizontal projection, and is used as a unit of length for beams and of depth for buildings. Hence we speak of three-rafter beams (*sanchuan fu*) or four-rafter buildings (*sijia chuan wu*).

Each description has the three parts described below. Each part characterizes one aspect of the transverse frame.

1 **Depth v** (in rafters). This is an even number. The student set this in stage A.
2 **Disposition of beams**, expressed in various combinations of three terms: *clear span*, *central division*, and *beams.* Of the 18 descriptions, none containing the term *clear span* also contains the terms *central division* or *beams.* A description not containing *clear span* contains *central division* or *beams* or both.
   a **Clear span** (*tong yan*). In a clear-span building, there are no interior columns, only the two in the front and back walls.
   b **Central division** (*fen xin*). In a centrally divided building, there is a column in the central position, below the ridge purlin.
   c **Beams** (*fu*). The length of the beam indicates the size of the bay it spans. Only the outermost beams are specified; the inner beams are merely implied.
3 **Total number c of columns**. The minimum is two, in a clear-span building. The maximum is one more than the number of rafters, but this possibility is not seen among the 18 variations.

As an example, consider this description:

> 6-rafter building, centrally divided, 1-rafter beam in front and back, with 5 columns (*liujia chuan wu, fen xin, qian hou zhaqian, yong wu zhu*).

The building has four divisions which, from front to back, are one, two, two, and one rafter deep. The two outside divisions are specified; the two inside divisions are merely implied.

The 18 descriptions form a partial enumeration of the sublanguage they belong to. From this we can hypothesize a generative definition of the set of descriptions. This takes the form of an initial description and a set of description functions, each one associated with a shape schema.[18]

18 I use the term *description function* differently from Stiny (1981). He means the ensemble of the initial description and the individual functions, analogous to a shape grammar. I mean the individual function.

Which means that the algorithm to generate the descriptions also generates the shapes. This is valuable, because the description counts things that we don't see.

The algorithm

The grammar is parametric. The algorithm has three substages:

1 Prepare the section diagram, consistent with the plan diagram subdesign $\omega_1 = \langle o_1, u_1, v_1 \rangle$. That is, generate a section diagram with depth $v = v_1$. This substage involves shape schemata only and is deterministic.
2 Set the salient parameters. That is, create the characteristic features of both the section diagram ρ and its descriptions *b* and *c*; each step involves a shape schema and an associated description function. This substage is nondeterministic.
3 Complete the subdesign. This means completing the section diagram (shape schemata only), and reducing the description from its characteristic form to that found in the text (description functions only). This substage is deterministic.

Substage 1, preparing the subdesign

The algorithm in this substage generates a base and two columns, one in the front wall and one in the back wall. This partial section diagram has the same depth v_1 as the plan diagram o_1. The rafter marker "walks" through the plan diagram, 2 rafters at a time from front to back. With each "step," the section diagram is enlarged by 2 rafters. Then the purlins are located; these coincide with the ends of the rafters. These are located two at a time, from the outside in.

The input of the plan diagram *o* is o_1, as created in stage A (figure 7). The initial shape of *r* is the point (β_2, γ_2). Both have the state label B. The descriptions b_c (Chinese) and b_e (English) each have three parts; each part is empty, shown by the symbol ∅. The following labels are used:

1 **Front marker**. This is a hollow triangular label, and marks the front of the section.
2 **Column markers**. These are hollow circular labels below the base, and mark potential column positions. They are spaced one rafter apart.
3 **Purlin markers**. These are hollow circular labels in the roof, and mark the locations of the purlins (and the ends of the rafters).
4 **State label**.

Each schema is of the form $\langle o, r, v, w, b, c \rangle \rightarrow \langle o', r', v', w', b', c' \rangle$, where *o* and *r* are drawings and *v*, *w*, *b*, and *c* are descriptions. It may be applied to the design $\langle o_0, r_0, v_0, w_0, b_0, c_0 \rangle$ when appropriate transformations of *o* and *r* are parts of o_0 and r_0, respectively. For convenience, let us adopt the following convention. For a schema B*n*, comprising the schemata $\langle o, r, v, w, b, c \rangle \rightarrow \langle o',$

r', v', w', b', $c'\rangle$, we will refer to the schema $o \rightarrow o'$ as schema B*n*o, the schema $r \rightarrow r'$ as schema B*n*r, and so on.

Schemata B1o, B2o, and B3o all move the rafter marker back 2 rafters. Schema B1r instantiates the base, with two columns 2 rafters apart, at $\langle \beta_2, \gamma_2 \rangle$; there are three column markers. Schema B2r expands the 2-rafter diagram to 4 rafters. Schema B3r is the recursive case, deepening by 2 rafters any diagram of more than 4 rafters. Schemata B4o and B5o identify the back end of the plan diagram and delete the rafter and cell markers; schema B4o is for $v = 4$, and schema B5o is for the general case. The plan diagram is not needed until stage C, so the state label is changed to C. Schemata B4r and B5r expand the section 2 rafters and change the outermost column markers to solid circles, preparing for the next substage.

Schema B6 instantiates the first two purlin markers, at the tops of the two columns; the two solid column markers "move" one rafter's length toward the center. Schemata B7, B8, and B9 instantiate the next two purlin markers and move the solid column markers toward the center. Schema B7 is for the 4-rafter case. Schema B8 is for the general case, except for the penultimate instantiation. Schema B9 is for the penultimate instantiation. Schema B10 instantiates the ridge purlin, "clears" the solid column marker, and replaces the outermost column markers with solid squares. The preliminary section diagram is ready for substage 2.

Substage 2, instantiating the disposition of beams

At this point the section diagram consists of a base, two columns, the column positions, and the purlin positions. The algorithm instantiates each beam specified in the description and the columns that support it. It works from the outside in: first the front, then the back, repeating once if necessary. With each shape schema are associated two description functions. One operates on the Chinese description b_c; the other operates on an equivalent English description b_e (figures 8a–b).

Schema B11 (clear span, *tong yan*) can be applied only to the section diagram with solid squares. It does not alter the drawing – it bypasses all other instantiation of beams – and goes straight to complete the section (substage 3). It alters the description, removes the solid square labels, and fills in the outermost purlin markers. Having this schema prevents the clear span from being a default condition.

Schema B12 (central division, *fen xin*), like schema B11, can be applied only to the drawing with solid squares. However, in this case, beams can still be instantiated afterwards with schemata B16–B23. Schema B12 instantiates the central column, thus dividing the interior into equal front and back halves. The solid squares become hollow. If there is no further subdivision of space, then we apply B14, which, like schema B11, removes the hollow squares and fills in the outermost purlin markers.

If we want to subdivide the space further, then we apply schema B13, which leaves the purlin markers hollow and changes the hollow squares to triangular markers. Both markers point to the back of the section; the solid one is below the front column and the hollow one is below the back column. If we wish to instantiate beams without central division, we apply B15 directly to the section with solid squares; this leaves the outermost purlin markers untouched and changes the solid squares to the triangular markers.

Schemata B16 through B23 instantiate beams of lengths measured in rafters. We start from the outsides (front and back) of the building and work towards the center. For each beam in front, we instantiate one in back. We can repeat the cycle once, if the section is deep enough. Schemata B16–B19 instantiate beams in front which are 1 through 4 rafters long (schemata to instantiate longer beams can easily be added), update the description, remove the solid triangle, and fill in the circular marker below the new column. This ensures that a corresponding beam is instantiated in back.

Schemata B20–B23 correspond to schemata B16–B19: they instantiate beams in back which are 1 through 4 rafters long, and update the description. They require a solid circle and restore the solid triangle.

After we have instantiated beams, the section has two triangular markers and one or more hollow column markers in between. Schemata B24–B26 terminate beam instantiation by removing these markers. Schema B24 moves the front solid triangle one rafter's length toward the center; schema B25 does the same for the back hollow triangle. When only the two triangles and one circle remain, we can apply schema B26 or B27, which removes the markers and fills in the outermost purlin markers. Schema B26 is for the non centrally divided case; schema B27 is for the centrally divided case. The subdesign is ready for substage 3.

Substage 3, completing the design

To complete the design, we must complete the drawing and reduce the descriptions; the order is not important. To complete the drawing, we instantiate those components not instantiated in substage 2; the descriptions are untouched. The algorithm is to move from the eaves to the ridge, at each purlin position (indicated by a black dot) instantiating a beam and a column as necessary (figures 9a–b).

Schemata B28–B31 are applied when the purlin position is above a beam and there is no column. They instantiate a beam and a column extending to the beam below, and add a hollow circular marker to show that there is no column on the base. At the far end of the beam, there may be no column (B28), a column of the same height (B29), a column that is higher (B30), or a column that terminates at the next purlin (B31).

Schemata B32–B35 are applied when the purlin position is at the top of a short column that stands on a beam below (not the base). Schemata B32–B34 are analogous to schemata B28–B30. Schema B34 is applied when there is already a beam.

Schemata B36–B39 are applied when the purlin position is at the top of a column that stands on the base. They are analogous to schemata B32–B35, but add a solid marker to show that the column reaches the base.

Schemata B40–B42 instantiate the topmost, 2-rafter beam and two columns. Of the two columns, none (B40), one (B41), or two (B42) may stand on the base. Schema B43 instantiates the king post, alters the labels, and changes the state label to C. The section diagram is complete; it will be used again in stage C.

As for reducing the descriptions, the goal is to transform the results of substage 2 (in their characteristic form) into the form seen in the corpus; the drawing is unchanged. We define an intermediate form thus: a_1 *in front,* a_2 *in back,* where a_1 and a_2 are the specified beams.

Schema B44 is for sections with four beams specified, i.e., with two cycles of instantiation. It combines the front beams and back beams into the intermediate form. Schemata B45 and B46 modify the intermediate form. If $a_1 = a_2$, then B45 is applied to get a_1 *in front and back.* If $a_1 + a_2 = v$ (number of rafters), then B46 is applied to get a_1 *abutting* a_2, and the number c of columns is reduced by one. In all other cases, the description is left unchanged. If $a_1 = a_2 = a_3 = a_4$, then B47 is applied to get *double* a_1 *in front and back.* If the front and back are symmetrical, i.e., if the description has the form a_1 *in front and back,* where a_1 is two beams, then it can be modified optionally.

If the form is $a_1 a_1$ *in front and back,* then B47 can be applied to remove the repetition, giving *double* a_1 *in front and back;* both forms are found in the corpus. If the form is $a_1 a_2$ *in front and back,* then B48 can be applied to add *both,* giving $a_1 a_2$ *in both front and back;* again, both forms are found in the corpus.

Examples

We return to our student, who in stage A created a 5 × 6 plan diagram. This defines the design space $\underline{\mathrm{P}}'_1 = \underline{\mathrm{P}}' \cap \underline{\mathrm{P}}_1$, where $\underline{\mathrm{P}}'$ is the sublanguage of well-formed subdesigns, $\underline{\mathrm{P}}_1 = R \times V_1 \times W_1 \times B \times C$, $V = \{6\}$, and $W_1 = \{1\}$. We see five section diagram subdesigns that he might create within this sublanguage. Only one is in the corpus; he must judge which of the other four, if any, is legal.

Substage 1 (figure 10) is common to all 6-rafter section diagram subdesigns. It generates a preliminary drawing with $v_1 = 6$ and the description *6-rafter building,* $\varnothing$*, with 2 columns.* Substages 2 and 3 differ for each subdesign.

1 A *6-rafter building, centrally divided, with double 1-rafter beams in front and back* (figures 11a–b, 16). This interior is full of columns. I think it is difficult to use and so probably illegal.
2 A *6-rafter building, with a 1-rafter beam in front and a 2-rafter beam in back* (figures 12, 16). Probably legal.
3 A *6-rafter building, clear span, with 2 columns* (figures 13, 16). Almost definitely legal.
4 A*6-rafter building, centrally divided, with 3 columns* (figures 14, 16). In the corpus.
5 A*6-rafter building, centrally divided, with a 1-rafter beam in front and back* (figures 15, 16). Probably legal.

The student is free to disagree with my interpretations and define his own by adding constraints on the application of the beam-instantiating schemata. He might, for instance, suppress the first section diagram by allowing 1-rafter beams to be instantiated only once and at the beginning. Thus he refines the generative definition of the sublanguage $\underline{P}''$ and the language Λ''.

Chapter 4

Creating the plan subdesign

The sublanguage of plan subdesigns

The plan subdesign π is a member of a 7-ary relation among one drawing (the plan p) and 6 descriptions (the number u of bays, the number v of rafters, the disposition b of beams, the number c of columns in depth, the width x of bays, and the length y of rafters).

The sublanguage of well-formed plan subdesigns Π is the cartesian product $U \times V \times B \times C \times X \times Y$. In stages A and B, the student created the plan diagram subdesign $\omega_1 = \langle o_1, u_1, v_1 \rangle$ and the section diagram $\rho_2 = \langle r_2, v_2, w_2, b_2, c_2 \rangle$. These define Π_2, Π'_2, and Π''_2 as follows:

$\Pi_2 = P \times U_1 \times V_1 \times B_2 \times C_2 \times X \times Y$, where $U_1 = \{u_1\}$, $V_1 = \{v_1\}$, $B_2 = \{b_2\}$, and $C_2 = \{c_2\}$,
$\Pi'_2 = \Pi' \cap \Pi_2$, and
$\Pi''_2 \subseteq \Pi'_2$.

The sublanguage Π'_2 is the equivalence class in Π' of well-formed plan subdesigns with $u = u_1$, $v = v_1$, $b = b_2$, and $c = c_2$; it is the student's current working sublanguage. The sublanguage Π''_2 is a relation on Π'_2; it is the student's current target sublanguage. By reducing Π' to Π'_2, the student reduces his problem.

In our continuing example,

$u_1 = 5$,
$v_1 = 6$,
b_2 = 6-rafter building, centrally divided, with a 1-rafter beam in front and back, and
$c_2 = 5$.

The student's current working sublanguage Π'_2. He has given himself the values of u, v, b, and c, and sets only the values of p, x, and y.

What we know

The constraints on x and y are as follows.

$$300\,\textit{fen} \geq \xi_1 \geq \ldots \geq \xi_m \geq 200\,\textit{fen},\ m = (u+1)\ /\ 2,$$ [19] and
$$y \leq 150\,\textit{fen}.$$ [20] (5)

The bay width x is clearly constrained by the diagram width u. This is why we separated them.

The algorithm

We generate the plan subdesign in steps, as follows. First, we generate a grid that embodies the actual dimensions ξ_i and y of the cells. We do this by referring to the plan diagram and manipulating the two drawings in parallel, as we did in stage B. We also manipulate the two descriptions x and y. Then, we read the section diagram to transfer the beam disposition to the plan; again, we use a parallel grammar.

Stage C comprises two grammars in parallel; following the convention described in stage B, let us call them stages Co and Cp. Stage Co handles the plan diagram generated in stage A; stage Cp, which is parametric, generates the plan. In addition, associated with each shape rule are description functions Cx and Cy which generate the descriptions x and y.

Substage 1, setting the sizes of cells

Stage Co (figures 17a–c) receives as its input shape the plan diagram o generated in stage A, and moves the cell marker through the grid; this is a more elaborate version of the walk through the plan diagram. To ensure that each cell is visited, an algorithm is used that walks the cell marker from front to back in each column of cells, starting at the center column and working out toward the end columns. As stage Co moves the cell marker through the grid, stage Cp instantiates the corresponding cell in plan, governed by the constraints given in (5). Each new cell in the plan has the same position as its counterpart in the plan diagram, but now embodies the dimensions which satisfy the constraints.

The schemata accomplish this in the following way. The initial shape in stage Cp is the point (α_3, β_3); with it is associated an index i, set to 0, which controls the assignment of bay width ξ_i. Schema C1o moves the cell marker and rafter marker to the next cell back. Schema C1p instantiates the front center cell at the initial point (α_3, β_3); this cell has the dimensions $\xi_1 \times 2y_3$. The index i is incremented by 1. The description functions C1x and C1y update the width and depth. Schema C2o does the same thing as schema C1o, but schema C2p must match a cell in the plan (and not the initial point). A new cell is instantiated behind the existing cell; it has the

19 This presentation of the constraints is simplified and hypothetical. Simplified, because the bay width has to do with two parameters which I have omitted: the number of bracket sets in the bay and the spacing between them. And hypothetical, because the text is vague. See Chen (1993, 11–15).

20 This is also hypothetical. See Chen (1993, 15–17).

same dimensions, namely $\xi_1 \times 2y_3$. Schema C3 is applied when we reach the backmost cell in the plan diagram (marked by the square label). Schema C3o moves the cell marker to the front cells in the next two columns out, and schema C3p instantiates a new cell, again with the dimensions $\xi_1 \times 2y_3$.

Schema C4p instantiates the frontmost cells in the two columns beside the center column. These two cells have the width ξ_2, which is subject to the constraints already given in (5); the depth is the same, $2y_3$. Schema C5 increments the index *i*; the plan diagram and the plan are essentially unchanged. Schema C6p is the same as schema C4p, except that it instantiates the frontmost cells in the columns not immediately next to the center column; schema C4 is applied when $u = 5$, while schema C6 is applied when $u \geq 7$. Schema C7, like schema C5, increments the index *i*. Schemata C8 and C9 operate like schemata C2 and C3. Schemata C10, C11, C12, and C13 instantiate the cells in the outermost columns of the plan; they operate like schemata C6, C7, C8, and C9.

To summarize, stage Co generates a plan diagram with the desired number of bays; this computation sets the values of *u* and *v*. Stage Co "reads" those values from the diagram, and guides stage Cp to produce a plan with those numbers of bays and rafters. In addition, those values are combined with the general parametric constraints (5) to give the specific constraints on the sizes of cells, x_i and y. This ensures that stage Cp instantiates cells with appropriate dimensions.

Substage 2, instantiating the disposition of beams

Now we modify the plan diagram *o* to reflect the disposition *b* of beams as embodied in the section diagram *r*. Each intersection in the plan diagram is potentially the endpoint of a beam and thus potentially the location of a column. If a beam is more than one rafter long, then no columns are needed beneath it. The disposition *b* of beams has this information. The strategy is to take each row of column positions in turn. If there are columns, then they are instantiated from the center out. If there are no columns, then the line is erased. The result is a plan that has cells with the correct dimensions and columns in the correct locations. This stage is deterministic; it mechanically transfers information from one drawing to another.

The schemata work like this (figure 18). Schemata C14, C15, C16, and C17 move us from one row to the next one back. There are four cases: both rows have columns (C14), the first has columns while the second does not (C15), the first has no columns while the second does (C16), and neither has columns (C17). The schemata governing the section diagram *r* (C14r, C15r, C16r, C17r) move a triangular row marker from row to row from front to back. If the row has columns, the marker is solid; if the row has no columns, the marker is hollow. The schemata governing the plan *p* (C14p, C15p, C16p, C17p) move corresponding triangular row markers along the two sides; these are hollow when the row is being transformed and solid when it is complete. These schemata additionally fill in the two triangular column markers at the front of the plan, which allows the schemata for working within the rows to be applied.

Schemata C18–C27 work on individual rows, adding columns or removing lines. They fall into four groups, according as the row is the front row, which always has columns (C18–C20); a middle row with columns (C21–C23); a middle row without columns (C24); or the back row, which always has columns (C25–C27). Within each group, the schemata governing the section diagram *r* are the same. For example, schemata C18r–C20r are all of the form $\delta \rightarrow \delta$; they are shown just once. Schemata C18p–C20p, on the other hand, are different and are all shown. Within each group of schemata, only the plan changes; the section diagram remains unchanged. This allows for loops, as we will see in the examples.

Let us discuss the first, second, and fourth groups of schemata, which deal with rows that have columns. Each group has three schemata, according as we are instantiating columns at the middle cell (C18, C21, C25), between the middle cell and the sides of the plan (C19, C22, C26), or at the sides of the plan (C20, C23, C27). The third group deals with a row with no columns. It consists of a single schema (C24), which removes the line.

Schema C28 terminates action in stage C. It is applied when all the columns in the back row have been instantiated; it removes unnecessary labels and changes the state labels so that the drawings are ready for their next use (the section diagram in stage F and the plan in stage D).

Examples

Recall that the student decided in stage A on bay widths $x = (300, 250, 200)$. He uses stage Co to create the plan (figures 19a–b). This plan has the same *u* and *v* as the plan diagram ω_1, but now has cells with the desired dimensions. Schemata C1, C4, and C10 instantiate the front bays with $\xi_1 = 300$, $\xi_2 = 250$, and $\xi_3 = 200$ *fen*, respectively. As for the rafter length (100 *fen* here), he sets it with schema C1, and it applies to all subsequent rafters.

The student now applies the description *b* to the partial plan. We will show two examples, both with $x = (300, 250, 200)$ and $y = 100$. They are taken from the five created in stage B. The first is the *6-rafter building, with a 1-rafter beam in front and a 2-beam rafter in back* (figures 20a–c). The second is the *6-rafter building, centrally divided, with a 1-rafter beam in front and back* (figures 21a–c). The section diagram changes less frequently than the plan. The derivation reflects this, with the section diagram moving from left to right along the top, and the plan dropping in a vertical line beneath each section diagram.

Chapter 5

Creating the partial elevation subdesign

The sublanguage of partial elevation subdesigns

The partial elevation subdesign ε is a member of a 6-ary relation among one drawing (the elevation *e*) and 5 descriptions (the number *u* of bays, the number *w* of storeys, the widths *x* of bays, the height *z* of columns, and the elevations *l* of purlins).

We distinguish $\underline{E}$, $\underline{E}'$, and $\underline{E}''$,[21] which are respectively the cartesian product $E \times U \times W \times X \times Z \times L$, the sublanguage of well-formed partial elevation subdesigns (the student's working language), and the sublanguage of legal partial elevation subdesigns (the student's target language). These are analogous to Λ, Λ′, and Λ″.

The student has created the plan diagram subdesign $\omega_1 = \langle o_1, u_1, v_1\rangle$, the section diagram $\rho_2 = \langle r_2, v_1, w_2, b_2, c_2\rangle$, and the plan subdesign $\pi_3 = \langle p_3, u_1, v_1, b_2, c_2, x_3, y_3\rangle$. These define $\underline{E}_3$, $\underline{E}'_3$, and $\underline{E}''_3$ as follows:

$\underline{E}_3 = E \times U_1 \times W_2 \times X_3 \times Z \times L$, where $U_1 = \{u_1\}$, $W_2 = \{1\}$, and $X_3 = \{x_3\}$,
$\underline{E}'_3 = \underline{E}' \cap \underline{E}_3$, and
$\underline{E}''_3 \subseteq \underline{E}'_3$.

The sublanguage $\underline{E}'_3$ is the equivalence class in $\underline{E}'$ of well-formed elevation subdesigns with $u = u_1$, $w = w_2$, and $x = x_3$; it is the student's current working sublanguage. The sublanguage $\underline{E}''_3$ is a relation on $\underline{E}'_3$; it is the student's current target sublanguage. The student, by reducing his working sublanguage from $\underline{E}'$ to $\underline{E}'_3$, has reduced his problem. In our continuing example,

$u_1 = 5$,
$w_2 = 1$, and
$x_3 = (300, 250, 200)$.

The student's current working sublanguage is $\underline{E}'_3$, and he needs to set only *z*. We put this together with x_3. We defer the calculation of *l*.

---

21 Again, distinguish $\underline{E}$ and *E*. $\underline{E}$ is related to ε; *E* is related to *e*.

What we know

About the height z of columns, Li says merely that "it may not exceed the width of the bay" (*bu yue jian zhi guang*) (Liang 1983, 153). Chen (1993, 17–19) concludes that this refers to the center bay; I follow his conclusion.[22]

The algorithm

Stage D is parametric. The designer selects the column height z in the context of relevant information that is already known, namely the bay widths x. This information resides in the plan, generated in stage C, so we transfer the information from there. We walk along the front of the plan, from the center bay toward the two sides (stage Dp). With each step (after the first step), we instantiate two new bays, with the existing bay widths and the new column height (stage De). The elevation generated in stage D is only partial; it does not include the roof, because the details are not yet worked out.

The initial plan in stage Dp is the plan p generated in stage C (figure 22). The initial elevation is the point (β_4, γ_4). Associated is a counter i, which keeps track of the bay width ξ_i. There are three cases for instantiating bays in elevation: instantiating the center bay, i.e., $i = 1$ (D1); instantiating two bays at neither the center nor the sides, i.e., $2 \leq i < m$, $m = (u + 1) / 2$ (D2); and instantiating the bays at the two sides, i.e., $i = m$ (D3). Schemata D1 and D2 hollow out the triangular markers so that schema D4 may be applied; schema D4 increments the counter i and resets the markers so that more bays may be instantiated (schemata D2 and D3, but not D1).

Example

The plan comes from stage C, with the solid bay markers already at the center bay (figure 23). The counter $i = 1$. Schema D1p moves the markers one bay outwards and hollows them. Schema D1e instantiates the elevation of the center bay, with width $\xi_1 = 300$ *fen* and column height $z = 200$ *fen.* Schema D4 advances the counter to $i = 2$, and "resets" the triangular markers. Schema D2p moves the markers from the penultimate bays to the outside bays. Schema D2e instantiates the next two bays, with width $\xi_2 = 250$ *fen* and the same column height $z = 200$ *fen.* Schema D4 advances the counter to $i = 3$, and resets the triangular markers. Schema D3p removes all markers and the state label, because the drawing will not be needed again. Schema D3e instantiates the two outside bays, with width $\xi_3 = 200$ *fen* and the same height column height $z = 200$ *fen*, and changes the state label to G, since it will be completed (the roof) in stage G.

22 Once again, I have simplified. The *Yingzao fashi* requires that column heights vary: the tallest are at the ends of the elevations, and the shortest are in the middles. This variation is called *shengqi:* see Liang (1983, 153–159, 260) and Chen (1993, 15–17). I have assumed that the columns are all the same height.

Chapter 6

Completing the design

The sublanguage of roof section subdesigns

The roof section subdesign δ is a member of a 4-ary relation among the roof section drawing *d*, the number *v* of rafters, the length *y* of rafters, and the heights *l* of purlins. The student has already set $v = v_1$ and $y = y_3$. The task is to set *l*, and since, as we will see below, *l* is a function of *v* and *y*, the student has merely to derive *l*, not to design it. We have:

$\Delta_4 = D \times V_1 \times Y_3 \times L$, where $V_1 = \{v_1\}$ and $Y_3 = \{y_3\}$,
$\Delta'_4 = \Delta' \cap \Delta_4$, and
$\Delta''_4 \subseteq \Delta'_4$.

Since *l* is determined by *v* and *y*, Δ'_4 is a singleton, and $\Delta''_4 = \Delta'_4$.

What we know: *juzhe*

The elevation *l* of purlins is determined by the process known as *juzhe* (literally, ‘raise and lower’). This is an indisputably generative definition, which makes sense for a book that is a guide for builders.

Before we go into the process, it is well to remember that in Chinese buildings, the purlins support the rafters. The spacing between purlins is even in the horizontal dimension but uneven in the vertical dimension. The rafters span from purlin to purlin, forming the characteristic curved section. This is the opposite of the western practice, where the rafters support the purlins, and span from ridge to eaves in a necessarily straight line, forming a triangular section. Here we are calculating the elevations, or vertical spacing, *l* of the purlins, where $l = (l_1, \ldots, l_m)$, $m = (u + 1) / 2$; $l_1 + \ldots + l_m =$ the overall height of the roof (from eaves purlin to ridge purlin).

The *Yingzao fashi* is explicit about *juzhe.* There are two steps: *ju,* ‘raise,’ and *zhe,* ‘lower’ [or depress].[23]

23 I have simplified *juzhe.* There are two other parameters, which I have omitted: the length of the eaves overhang and the type of roof tile. See Liang (1983, 182–184, fig. 26, 265).

1 **Raise**. Find the height of the ridge purlin (above the eaves purlin): it is one-quarter of the depth vy of the building. Call this the roof height. Connect this point and the eaves purlin; call this the working roof line.

2 **Depress**. Find the elevation of the first purlin below the ridge purlin, in the following way. The horizontally projected length of a rafter is y, so the next purlin is displaced by a distance of y. Find the intersection of the working roof line and the vertical line at a distance of y. From this point "push" down a distance of 1/10 of the roof height; the resulting point is the elevation of the purlin. From this point to the eaves purlin, draw a new working roof line. Repeat with the remaining purlins; each time halve the "depression": 1/20, 1/40, etc.

The algorithm

The grammar is parametric and deterministic (figure 24). The initial shape is a point (β_5, γ_5). The initial description $l = 0$. The descriptions $v = v_1$ and $y = y_3$ are set. Schema E1 raises the roof: it draws the roof outline and the first working roof line. The working roof line is marked at its lower end (the eaves purlin) by a solid square, the upper end by a solid circle. A counter i is set to 0. Schema E2 can be applied if there is at least one purlin to be located. The position of this purlin is depressed by the required distance $a_i = h_0 \ / \ (10 \times 2^i)$; the labels become hollow. When the labels are hollow, only schema E3 may be applied. This schema establishes the new working roof line, marked with solid labels. The counter i is incremented by 1, and the description l is updated by concatenating the value of l_i. If we have finished with all the purlins, then schema E4 may be applied. This updates l again and removes the markers on the working roof line. The segments form the roof section.

Example

Let us continue with our 6-rafter building with 100-*fen*-long rafters; that is, $v = 6$ and $y = 100$ *fen* (figure 25). We apply schema E1 to raise the ridge purlin to a height of 150 *fen* (above the eaves purlin). We apply schema E2 to find the elevation of the next purlin down. The intersection at the working roof line is 150 / 3 = 50 *fen* below the ridge purlin. Schema E3 increments the counter and calculates the new working height (150 – 65 = 85 *fen*) and erases the construction lines. The description $l = 65$. We still have another purlin to locate, so we apply schema E2 again. The third purlin is lowered 42.5 + 7.5 = 50 *fen.* Schema E3 establishes the new working roof line, erases the construction lines, increments the counter, and updates the description $l =$ (65, 50). Now we have finished with all the purlins, so we apply schema E4, which removes the labels, updates the description $l =$ (65, 50, 35), and changes the state label to F.

Creating the section subdesign (stage F)

Stage F generates the section subdesign σ. This is a member of a 8-ary relation among one drawing (the section drawing *s*) and 7 descriptions (*v*, *w*, *b*, *c*, *y*, *z*, and *l*).

$\Sigma_5 = S \times V_1 \times W_2 \times B_2 \times C_2 \times Y_3 \times Z_4 \times L_5$,
$\Sigma'_5 = \Sigma' \cap \Sigma_5$, and
$\Sigma''_5 \subseteq \Sigma'_5$; actually $\Sigma''_5 = \Sigma'_5$.

The working sublanguage Σ'_5 is, like Δ'_4, a singleton. The grammar (not shown) is parametric and generates the section drawing deterministically.

Creating the elevation subdesign (stage G)

Stage G completes the elevation subdesign ε begun in stage D. As in stage F, only the drawing remains to be specified.

$\underline{\mathrm{E}}_6 = E \times U_1 \times W_2 \times X_3 \times Z_4 \times L_5$,
$\underline{\mathrm{E}}'_6 = \underline{\mathrm{E}}' \cap \underline{\mathrm{E}}_6$, and
$\underline{\mathrm{E}}''_6 \subseteq \underline{\mathrm{E}}'_6$; actually, $\underline{\mathrm{E}}''_6 = \underline{\mathrm{E}}'_6$.

The working sublanguage $\underline{\mathrm{E}}'_6$ is a singleton. The grammar (not shown) is parametric and completes the elevation drawing deterministically.

The completed design

The completed design λ is the single member of the equivalence class $O_1 \times R_2 \times P_3 \times E_4 \times D_5 \times S_6 \times F_7 \times U_1 \times V_1 \times W_2 \times B_2 \times C_2 \times X_3 \times Y_3 \times Z_4 \times L_5$ (figures 1a, 1b).

Discussion

Designs as *n*-tuples of elements

One of the most conspicuous innovations of this grammar is that it uses descriptions. These devices are not new: Stiny (1981) formalized their relation with drawings and gave an example of descriptions that characterized certain formal aspects of the drawings, such as the number of courtyards. We might call these *formal* descriptions. As the drawings change, so do the descriptions; the two develop in step. In our grammar we use exactly the same type of formal descriptions, such the numbers u of bays and v of rafters.

The descriptions of the sections are unusual. These are also formal and no doubt were obvious to Li Jie and his readers, but they are not obvious to us modern readers. We have different habits of classification and so describe, even see, the same sections differently. For example, we would probably specify the central space in a Song section and are surprised that our counterparts did not. Using the Song descriptions is like putting on Song spectacles. This was an unexpected benefit.

A description is just one type of element in a definition of a design that is richer than a single drawing, which is what most grammars have consisted of. Recall Stiny's (1990) definition of a design as a member of an n-ary relation on drawings, descriptions, and other devices. The design at any stage is an n-tuple of these various elements and is manipulated by rules, schemata, or functions also bound in n-tuples. In our grammar, the drawings, not the descriptions, drive the derivation, because, as Stiny (1981, 265) says, they "depend on [the design's] constituent spatial elements and the way these are combined."

Can it happen the other way around? Can the descriptions drive the derivation? In a general way, the answer must be yes. The design problem is typically long on *functional* descriptions and short on drawings, while the reverse is true for the design solution. The trend of the process is from descriptions to drawings. An example from the *Yingzao fashi.* We know that the size and type of a building were related to, even determined by, the occupant's rank or job title. In other words, there was a mapping from a functional description (the rank or job title) to a formal description (of the appropriate building). So when the assistant deputy third secretary needed a new office, it was presumably a straightforward matter to determine that he would get, say, a 3

× 4 *ting* hall at the sixth grade.[24] This formal description would then have been passed to the builders, who translated it into a spatial object. Unfortunately, Li Jie says little about the mapping, and our grammar does not take it into consideration.

People and grammars

Our 16-tuple design includes descriptions, but is a richer approach for another reason. The 16 elements are not a monolithic bloc; they can be restructured in ways that make sense to us. This allows us to identify and solve smaller design problems, solve some before others, shift from one representation to another – in short, simply to work the way designers work. This feature requires no new technical devices, but is innovative in its motivation to make a grammar feel more natural to a designer, more *designerly.* The result is a grammar that reflects in some way what Schön (1987) would have called the conversation between designers and designs.

Designerliness suggests a new application of grammars. Grammars are production systems, and production systems, according to Akın (1986, 115), are good for characterizing process: "the properties of parsimony, flexibility, and open-endedness coupled with [their] general agreement with psychological attributes of information processing make production systems … ideal device[s] for modelling the behavior of designers and representing inference making." This suggests that grammars can characterize more than styles; they might also characterize process. Knight's (1999–2000) observation on a grammar's believability or accuracy in historical applications is usefully transposed to this new context (substituting *cognitive* for *historical*):

> [T]here is no way of verifying whether a grammar is [cognitively] accurate, even with the testimony of a living designer. Thus, shape grammars are not, to use AI terminology, strong theories of styles, but weak theories of styles. The more compelling a grammar is, though, the more it may seem to correspond to [cognitive] reality.

Now if a grammar can characterize process, then it follows that a designerly grammar can form the basis of the interface of an automated grammar. This is a new approach to automating grammars. Most automated grammars have been interpreters, and have aimed at the rule: how to define it and execute it.[25] The user specifies a rule in real time, and the interpreter applies it to the shape in real time. The advantages are two: the user can define rules spontaneously, and the interpreter uses the shape grammar mechanism, complete with emergence, to apply the rules, thereby sparing the user time and errors in the search-and-replace process. This highlights the definition and the before-and-after quality of the rules. We might call this a *rule-centered*

24 At my university, a professor is allotted an office of 22 m^2; a senior lecturer, 18 m^2; a lecturer, 15 m^2; an assistant lecturer, 10 m^2; and a junior academic or researcher, 3.5 m^2. Same thing.

25 Tapia (1999) and Wang (1998) are representative.

interface; it is a bottom-up approach, in the sense that it is faithful to the shape grammar mechanism.

The disadvantage is that it doesn't help much with style grammars, with design process as we have discussed above. There are several reasons for this. First, focusing on the individual rule means that it is more difficult to get a sense of the whole grammar: we miss the forest for the trees. This probably encourages few rules, repeated many times. But style grammars have all had many rules, some of which do not transform the shape, but only control the process. Which brings us to the second reason, that currently available interpreters do not yet support all the technical devices that enable designerliness: parameterization, descriptions, weights, multiple drawings, and most types of labels.

These will come in time, but in the meantime we can think about automating grammars to emphasize the user's interaction with the grammar – what he decides and when he decides it – and suppressing the mechanics of how the decisions are executed. We might call this a *process-centered* approach. If we suppress the details of the execution, we need not actually implement the shape grammar mechanism; we can merely simulate it. This allows the user to concentrate on the overall structure and logic of the grammar as a characterization of the style. The underlying tree of possible derivations is fixed, so it can be worked out beforehand and simulated; as far as the user is concerned, there is no need to have it worked out in real time. Similarly, the rules are fixed and known in advance.

A process-centered simulation allows the user to work with a grammar that is as complex and rich as any on paper. It also allows us to examine whether the designerly process proposed in the grammar actually feels natural to users. Certainly there are disadvantages. The larger and more complex the grammar, the more conditions that have to be simulated. All possible derivations have to be worked out both by hand and in advance. Each simulation has to be custom written and, once written, is fixed. But this is merely provisional until we have developed interpreters that can execute any grammar that can be written.

As we discussed in the introduction, humans do more than just decide which rule to apply under which transformation. They play a critical role in defining style, and our grammar reflects that role. As a definition of style, the *Yingzao fashi* is incomplete, and our grammar reflects that too. If the student wants to know what the style of the *Yingzao fashi* "really is," he must complete the definition himself. This he can do only by examining all the evidence, formulating and testing a hypothesis, and coming to his own conclusion; this is what the connoisseur does, and it is important for the student to know this. The lesson here is that style is not "out there"; it is a human construct. As March and Stiny (1985, 52) say: "The designed world is one of our own making, and its making is our responsibility alone." What better lesson for a student of design?

A computational approach to Chinese architectural history

Our grammar points the way for further work in Chinese architectural history as well. One possibility is the automated grammar of the type described above, which would help students experience the style both from the outside (as a connoisseur) and from the inside (as a designer).

Another possibility is a comprehensive comparative study of wood frame architecture. The *Yingzao fashi* prescribes a style that evolved until just after the beginning of the Ming. At that point, there was a great stylistic break, after which the style changed markedly and virtually ceased to evolve.[26] Coincidentally, for this period we have the *Gongcheng zuofa zeli* of 1733. This sets up a series of comparisons which can be done with shape grammar. For instance, now that we have a grammar that generates buildings in the style of the *Yingzao fashi*, we can formalize the relation between the manual and the extant pre-Ming buildings: how does the grammar have to be modified to produce those buildings? Then, since the extant buildings change through time, we can see how the grammars evolve, as Knight (1994) does. Similarly, we can construct grammars of the style of the *Gongcheng zuofa zeli* and of that of Ming–Qing buildings. We can compare them with each other and with their pre-Ming counterparts.

Thus we can do a shape grammatical study of Chinese wood-frame architecture from the eighth to the twentieth century; if we consider indirect evidence, we can begin even earlier. This would be a complete formal statement of a long tradition, and an appropriate extension of the studies of what Liang Sicheng called the "grammar book" of Chinese architecture.

---

26 See Liang (1984, 103).

References

Akın, Ömer. 1986. *Psychology of architectural design*. London: Pion.

Chen Mingda. 1992. Tang Song mu jiegou jianzhu shice jilu biao [Field measurements of Tang and Song wood frame buildings]. In *Jianzhu lishi yanjiu* [Studies in architectural history], edited by He Yeju, 231–261. Beijing: Zhongguo jianzhu gongye.

———. 1993. *Yingzao fashi da muzuo zhidu yanjiu* [A study of structural carpentry in the *Yingzao fashi*]. 2nd ed. Beijing: Wenwu.

Chiou, Shang-chia, and Ramesh Krishnamurti. 1995. The grammar of Taiwanese traditional vernacular dwellings. *Environment and planning B: planning & design* 22: 689–720.

Downing, F., and Ulrich Flemming. 1981. The bungalows of Buffalo. *Environment and planning B: planning & design* 8: 269–293.

Duarte, José Pinto. 2000. *Malaguiera: Alvara Siza Vieira – 1977/99.* 1999 [cited 2000]. Available from http://destec.mit.edu/malag/.

Fairbank, Wilma. 1994. *Liang and Lin: partners in exploring China's architectural past.* Philadelphia: University of Pennsylvania Press.

Flemming, Ulrich. 1981. The secret of the Casa Giuliani Frigerio. *Environment and planning B: planning & design* 8: 57–96.

———. 1987. More than the sum of parts: the grammar of Queen Anne houses. *Environment and planning B: planning & design* 14: 323–350.

Glahn, Else. 1976. Li Chieh. In *Sung biographies,* edited by Herbert Franke, 523–529. Wiesbaden: Franz Steiner.

———. 1984. Unfolding the Chinese building standards: research on the *Yingzao fashi*. In *Chinese traditional architecture,* edited by Nancy Shatzman Steinhardt, 47–57. New York: China Institute in America.

Guo, Qinghua. 1999. *The structure of Chinese timber architecture.* London: Minerva.

Knight, Terry W. 1980. The generation of Hepplewhite-style chair-back designs. *Environment and planning B: planning & design* 7: 227–238.

———. 1994. *Transformations in design: a formal approach to stylistic change and innovation in the visual arts.* Cambridge, England: Cambridge University Press.

———. 2000. *Shape grammars in education and practice: history and prospects.* 1999–2000 [cited 2000]. Available from http://www.mit.edu/%7Etknight/IJDC/.

Koning, Hank, and Julie Eizenberg. 1981. The language of the prairie: Frank Lloyd Wright's prairie houses. *Environment and planning B: planning & design* 8: 295–323.

Li, Andrew I-kang, and Jin-Yeu Tsou. 1996. Using virtual models to teach traditional Chinese construction. In *The introduction of technology: CAAD teaching and research directions in Asia. Proceedings of the first conference of the Association of Computer Aided Architectural Design Research in Asia* (*CAADRIA*), edited by Thomas Kvan, 119–27. Hong Kong.

Li Mingzhong babai ershi zhou nian jinian, The 280th anniversary of Li Chieh. 1930. *Zhongguo yingzao xueshe huikan, Bulletin of the Society for the research in Chinese architecture* 1 (1): 1–24.

Liang Sicheng. 1983. *Yingzao fashi zhushi* [The annotated *Yingzao fashi*]. Beijing: Zhongguo jianzhu gongye.

———. 1984. Zhongguo jianzhu zhi liangbu "wenfa keben" [The two "grammar books" of Chinese architecture]. In *Liang Sicheng wenji* [The collected works of Liang Sicheng], 357–363. Beijing: Zhongguo jianzhu gongye.

Liang Ssu-ch'eng [= Liang Sicheng]. 1984. *A pictorial history of Chinese architecture: a study of the development of its structural system and the evolution of its types.* Edited by Wilma Fairbank. Cambridge, Mass.: MIT Press.

March, Lionel, and George Stiny. 1985. Spatial systems in architecture and design: some history and logic. *Environment and planning B: planning & design* 12: 31–53.

Schön, Donald A. 1987. *Educating the reflective practitioner: toward a new design for teaching and learning in the professions.* San Francisco: Jossey-Bass.

Stiny, George. 1977. Ice-ray: a note on the generation of Chinese lattice designs. *Environment and planning B: planning & design* 4: 89–98.

———. 1981. A note on the description of designs. *Environment and planning B: planning & design* 8: 257–267.

———. 1990. What is a design? *Environment and planning B: planning & design* 17: 97–103.

———. 1992. Weights. *Environment and planning B: planning & design* 19: 413–430.

Stiny, George, and William J. Mitchell. 1978. The Palladian grammar. *Environment and planning B: planning & design* 5: 5–18.

———. 1980. The grammar of paradise: on the generation of Mughul gardens. *Environment and planning B: planning & design* 7: 209–226.

Tapia, Mark. 1999. A visual implementation of a shape grammar system. *Environment & planning B: planning & design* 26: 59–73.

Wang, Yufei. 1998. 3D architecture form synthesizer. S.M.Arch.S. thesis, Department of Architecture, Massachusetts Institute of Technology, Cambridge, Mass.