A prototype interactive simulated shape grammar

I-KANG LI Andrew

Department of Architecture, The Chinese University of Hong Kong

Existing shape grammar interpreters have implemented some of the distinguishing features of the formalism, notably emergence or rule application under multiple transformations. By contrast, I implement a grammar in which these capabilities either are not required or can be simulated. This frees a user from mechanical issues, allowing him to pay more attention to the process embodied in the grammar.

Introduction

Shape grammars are well developed in theory, but in practice many people find them difficult to use. Determining which rules can be applied to a given shape, under which transformations, and with what outcomes; applying a rule accurately; recording the derivation – all are time-consuming, error-prone tasks. Automating at least some of these tasks would allow the user to devote more attention to the development of the design.

Several interpreters have been developed to automate some of the more mechanical aspects of using grammars. These interpreters support emergence and identify all rule applications (Tapia 1999; Wang 1998). The user can formulate an initial shape and rules; see all the ways in which to apply the rules; select one and apply it in one of those ways; and transform the shape; all accurately and in real time. These interpreters do not simply allow users to formulate an initial shape and rules in real time; they require users to do so. This encourages users to create designs by using a few rules and applying them many times. A fractal, which is created by a single rule applied many times, is an extreme example of this kind of design.

But these are not the only types of designs that designers would like to create. Also of interest are, for example, products like architectural plans and sections (Chiou and Krishnamurti 1995; Duarte 1999). These are generated by grammars that are in some ways more complex than those mentioned above – they have many labels, have many rules applied few times, and are often parametric – and so cannot be handled by existing interpreters. On the other hand, in some important respects they are simpler: the initial shape and rules are predefined; the rules can be applied under fewer assignments and transformations; and emergence is rarely required.

These facts present an opportunity: forego emergence, forego user-defined initial shape and rules, and simulate the rest. Thus, instead of implementing the most characteristic features of the shape grammar formalism, I propose to minimize or simulate those features and instead to automate those points of interaction that are difficult for users. To test the feasibility of this approach, I am constructing with Macromedia Flash a prototype of an interactive simulated parametric shape grammar that creates sections of wood-frame buildings according to the twelfth-century Chinese building manual Yingzao fashi (Liang 1983; Li 2001). As this article is written, the grammar consists of 48 rules and generates in real time dozens of building sections that are 6 rafters deep.

The algorithm

Our knowledge of building sections in the Yingzao fashi comes from a corpus of 18 sections, varying in depth from 4 to 10 rafters. Each section is represented by a drawing and a terse written description, for example, a 6-rafter building, with a 2-rafter beam in front and in back, and with 4 columns (figure 1). Notice that the middle 2-rafter beam is not specified, but merely implied by the overall depth (6 rafters) and the two outer beams (each 2 rafters long). Clearly, twelfth-century Chinese builders did not necessarily see what we see.

Following Stiny's (1981) formalization of descriptions, shapes, and their generation, I develop a three-part algorithm. The first part consists of rules, each of which manipulates one salient feature (say, a 2-rafter beam in front) in both its symbolic representation (the description) and its spatial representation (the shape). The second part of the algorithm does grammatical housekeeping (cleaning up the labels), and the third part completes the section (by instantiating building components, such as rafters, not specified in the description). These parts of the derivation are deterministic.

Thus this example from the Yingzao fashi is interesting to implement, not only because it generates a language of designs, but also because the algorithm itself embodies a traditional logic which may in turn be appreciated by a modern user.

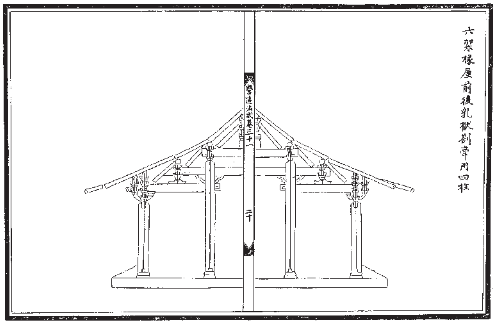

Figure 1. One of 18 sections shown in the Yingzao fashi (Liang 1983). The written description is a 6-rafter building, with a 2-rafter beam in front and in back, and with 4 columns.

The interface

The screen of the implementation has two halves (figure 2). The lower half shows the (predefined) rules. There are 48 in all, but there is room on the screen for only 25, so they are spread over 3 pages. The three tasks correspond to the three parts mentioned above: set the salient features, clean up the labels, and complete the section. The upper half of the screen shows the current section, either alone (large size) or together with the preceding derivation (small size).

How does the simulation make the grammar easier to use? It is helpful to put it in terms of the formalization for a parametric grammar: if t (g (A)) ≤ C, then C¢ = [C – t (g (A))] + t (g (B)). Our simulation helps the user in the following ways.

How does the user know which rules can be applied to the current section? A rule (or, to be more precise, a schema) A Æ B is applicable to the current section C if its left shape A is, under some assignment g and some transformation t, a part of C. Those rules which have no such values of g and t – which cannot be applied – are dimmed; those which can be applied are not dimmed.

How does the user know under which assignments and transformations the applicable rules can be applied? For any applicable rule in this grammar, there are – by design – always only one assignment g and usually only one transformation t (occasionally two), so there is usually only one way, and at most two ways, for a rule to be applied. Rules that can be applied under two transformations (as it happens, reflections) occur on the third page. Here the user chooses the reflection by toggling a switch.

How does the user know the outcome of applying a rule? When he moves the cursor onto an applicable rule, four shapes are highlighted in red: the left shape of the rule, that is, A; the left shape of the rule as a part of the current section under the appropriate assignment and transfor-

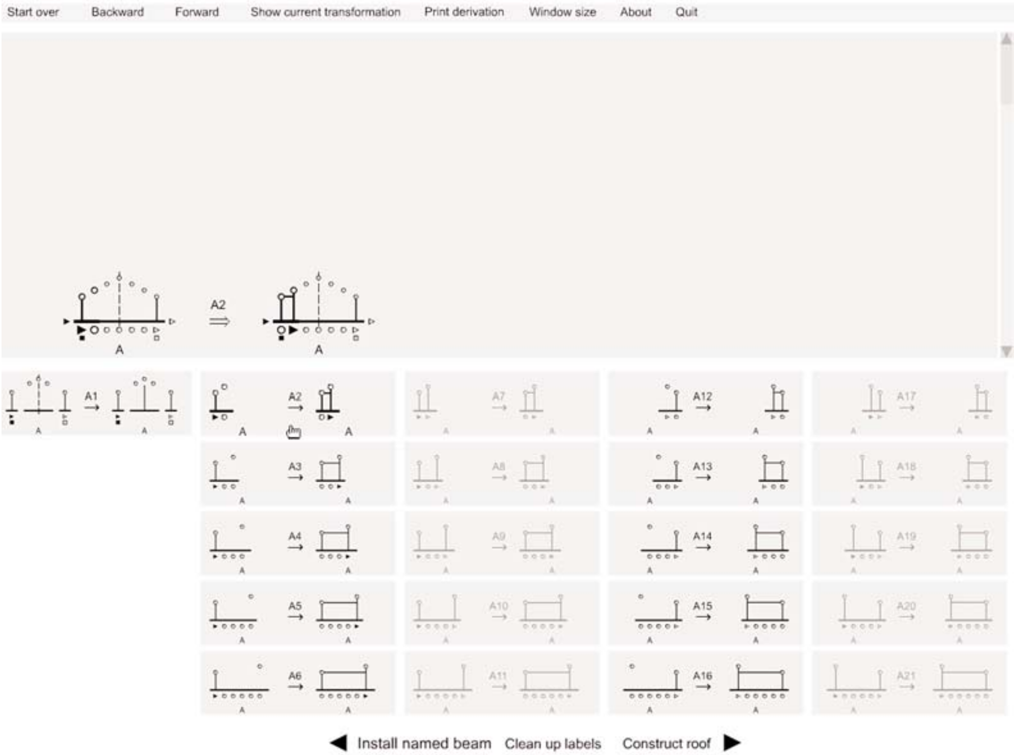

Figure 2. The screen shows the current section above and the rules below. The outcome of applying rule A2 is shown; the left and right shapes of the rule and, under the appropriate assignment and transformation, the corresponding shapes in the current and next sections are highlighted.

mation, that is, t (g (A)) ≤ C; the right shape of the rule, that is, B; and the right shape of the rule as a part of the next section under the same assignment and transformation, that is, t (g (B)) ≤ C¢. This test is simulated in real time by the implementation, based on an atomic representation of the shapes.

How does the user apply a rule accurately? He clicks on the rule, and the implementation applies it in real time, based on the same atomic representation.

How does the user record the derivation? The derivation is shown in the upper half of the screen, and can be printed out.

Discussion

In informal trials, users understood easily what the rules mean and how to use them. In particular, they appreciated how to use the first page of rules to set the salient features of the section. They also appreciated that the other two groups of rules define a deterministic process and in fact would have much preferred not to have to apply each rule manually, since it required no thought on their part.

This suggests that grammars can in fact be useful for describing or guiding a user's participation in a design process, especially if those parts of the process which do not require his participation can be hidden from him. Thus, in this case, the two deterministic sequences should be automated.

In general terms, the approach is born out. I intend not only to make the specific improvement mention above, but also to increase the scope of the grammar by allowing sections of other depths (4, 8, and 10 rafters) and including descriptions as part of both the rules and the sections.

One might object that this approach is fundamentally antithetical to shape grammar: after all, it involves atomic units and no emergence. But in fact it is antithetical to only one aspect of shape grammar – the definition of a line as containing infinitely many shorter lines – while it enables other aspects, such as supporting a designer's process. Right now, we have to choose between two incomplete approaches, but when complete interpreters – those that support all the theoretical features of grammars – are available, we will be able to have our cake and eat it too.

Acknowledgements

I acknowledge with thanks special support from the Department of Architecture, The Chinese University of Hong Kong (CUHK), and a Direct Grant for Research, CUHK. I also thank my assistant Ephes Lau Man Kuen for his fine work.

References

Chiou, S. C., and Krishnamurti, R.: 1995, The grammar of Taiwanese traditional vernacular dwellings, Environment and planning B: planning & design, 22, pp. 689–720.

Duarte, J. P.: 1999, Malaguiera: Alvara Siza Vieira – 1977/99, http://destec.mit.edu/malag/.

Li, A. I.: 2001, A shape grammar for teaching the architectural style of the Yingzao fashi, Ph. D. dissertation, Department of Architecture, Massachusetts Institute of Technology, Cambridge, Mass.

Liang, S. C.: 1983, Yingzao fashi zhushi [The annotated Yingzao fashi], Zhongguo jianzhu gongye, Beijing.

Stiny, G.: 1981, A note on the description of designs, Environment and planning B: planning & design, 8, pp. 257–267.

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

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