Shape rules for architectural design

William Fawcett

This paper was presented at the *EuropAI* conference held in Liège, Belgium, in March 1990.

william.fawcett@carltd.com

CAMBRIDGE
ARCHITECTURAL
RESEARCH
LIMITED

*Paper presented at **EuropIA 90** conference, Liège, Belgium, March 1990*

SHAPE RULES FOR ARCHITECTURAL DESIGN

W H FAWCETT
Cambridge Architectural Research Ltd
Cambridge - UK

ABSTRACT

The left-hand side and right-hand side of shape rules consist of shapes. Rule-based production systems using shape rules can generate complex shapes, or designs. A shape rule interpreter has been constructed in the AutoCAD computer-aided design environment, using AutoCAD for display and AutoLISP for reasoning. It has been used educationally in schools of architecture and commercially in architectural design offices.

1. SHAPE RULES

1.1 Architectural knowledge represented in rules

Artificial Intelligence has emphasised the idea of rule-based programming. The use of rules to encapsulate architectural knowledge pre-dates the AI boom, for example in the work of Christopher Alexander from the mid-1960's with his "atoms of environmental structure" [1] and "pattern languages" [eg.2]. He used sets of rules to represent many aspects of architectural knowledge, both geometrical and non-geometrical. A large pattern language contains over 200 rules.

1.2 Shape rules: structure and application

Shape rules were first discussed in the early 1970's and have been the subject of much academic research [3]. A shape rule is like other rules which have a left-hand side and a right-hand side:

IF {conditions to be satisfied ... } > THEN {actions to be performed ...}

Both the LHS and the RHS of a shape rule are shapes [fig.1]:

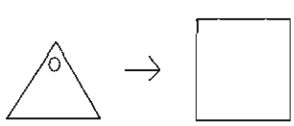


fig.1 *A shape rule*
The left hand "IF ... " side and the right hand "THEN ... " side are both shapes.

A shape rule can be applied to a shape if the LHS of the rule is a subshape of the shape; when the rule is applied the subshape corresponding to the LHS is deleted and one corresponding to the RHS is inserted. The scale and rotation of the deleted shape are preserved for the inserted shape [fig.2].

The Eden Centre 47 City Road Cambridge CB1 1DP United Kingdom Tel +44 (0)1223 460475 Fax +44 (0)1223 464142

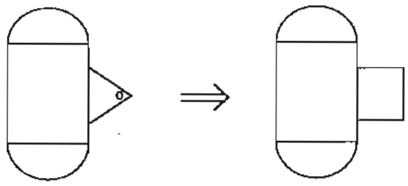


fig.2 *Applying a shape rule to a design*
The shape rule shown in fig.1 can be applied to a design which contains its LHS as a subshape. When the rule is applied the subshape is deleted and replaced with one corresponding to the RHS of the rule. The scale and rotation of the deleted shape are preserved for the inserted shape.

1.3 Shape grammars and languages of designs

A shape rule represents a simple geometrical transformation. To generate non-trivial shapes or designs a number of shapes rules are required, and a set of related shape rules is called a shape grammar. By applying the rules from a grammar in various ways a variety of designs can be produced from the same initial shape, called the language of the grammar [fig.3]. The language of even a simple grammar can be infinite, by endlessly repeating rule applications. The designs produced in the language of a given shape grammar often share a "family resemblance".

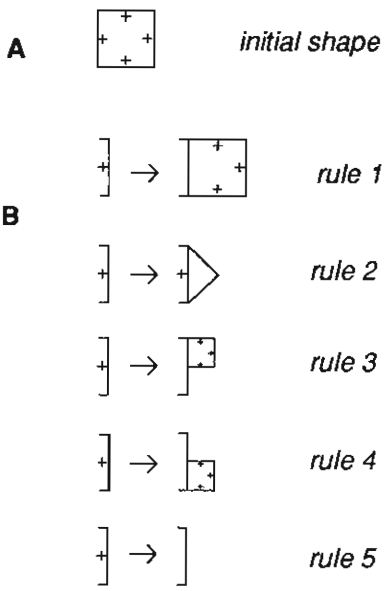
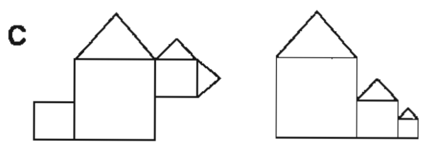


fig.3 *A simple shape grammar*
A shape grammar comprises an initial shape (A) and a number of shape rules (B). By applying the shape rules in different ways a variety of shapes (eg. C)can be generated in the language of the grammar



1.4 Design knowledge embedded in shape rules

Shape rules have been written for established architectural styles and other types of design [eg.4,5]. Using these rules it is possible to re-create historic examples of the styles and also to create new designs which posses the same characteristics. Arguably this is a good model of the design method of most architects and architectural students, who produce new designs but rely heavily on established architectural languages. New architectural designs are generated every day, but new architectural languages arises only rarely.

Using shape rules designers can create designs that they would not be able to do unaided. For example, a designer with only superficial knowledge of Frank Lloyd Wright's Prairie Houses could design a convincing new pastiche with a Prairie House shape grammar, because of the knowledge of the style embedded in the shape rules, not in he designer. Shape rules can be seen as an example of Artificial Intelligence in the domain of architectural design.

2. "EMERGENT " RULES OR "BLOCK" RULES?

2.1 Generating designs with shape rules

The generation of designs with shape rules is typical of rule-based production systems. It involves a repeated cycle of three tasks:

1 rule matching
2 rule selection
3 rule application

The first task involves matching the LHS's of all the active shape rules against the current design. Remember that the LHS of a rule matches if it exists as a subshape in the current design regardless of scale or rotation; so each rule may match in in more than one way. Rule matching quickly becomes non-trivial, although it is essentially mechanical.

The second task is often called "conflict resolution" in production systems. It is the critical stage in determining which of the designs in the language of the shape rules will result from a design session. In the applications reported here the task is carried out by human designers.

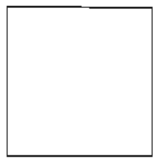
The third task, like the first, is mechanical. The shape corresponding to the LHS of the selected rule is deleted and the RHS shape inserted.

When the three tasks have been completed the design has been transformed and the cycle is repeated until a finished design is achieved.

The entire process of design with shape rules can be carried out by hand. But there is a good deal of mechanical work, which is tedious and error-prone when carried out by humans. It is therefore natural to use a computer graphics system for the mechanical tasks, with rule selection being carried out by a human operator. This is the scenario for shape rule based CAD.

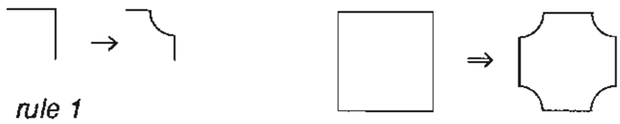
2.2 Rule matching: "emergent" shapes

Suppose that we are concerned with designs consisting of lines in 2D, that is, drawings. When matching the LHS of the active rules we search for lines with the same geometrical relations. There can be ambiguities due to conflicting interpretations of the constituent parts of the current design [fig.4].

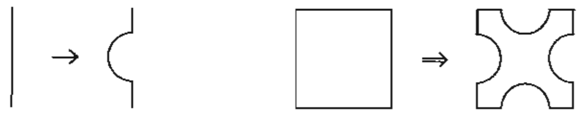


current shape

fig.4 *Ambiguous shapes*
The LHS's of both shape rules can match the current shape, based on different interpretations of the current shape.

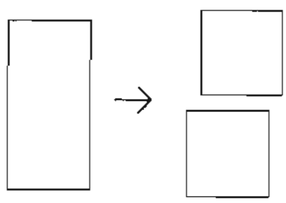


rule 1



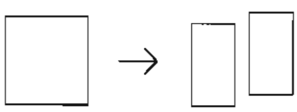
rule 2

The problem could be overcome by defining the current design in terms of fixed elements and matching shape rules accordingly. However, by restricting ourselves to pre-defined elements the interesting case of "emergent" shapes is excluded [fig.5].

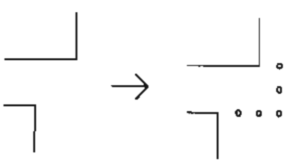


rule 1

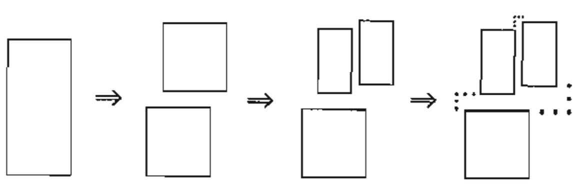
fig.5 *Emergent shapes*
The LHS of rule 3 is not a distinct shape, but an "emergent" shape formed by the juxtaposition of other shapes.



rule 2



rule 3



typical shape generation using the three rules

2.3 Rule matching: "blocks"

It is possible to define a vocabulary of elements for use in the LHS and RHS of shape rules; the designs generated by the rules will be composed from the same vocabulary of elements. In CAD systems a part of a drawing can be defined as a "group" or "block" and named; an insertion point is also specified. When a new instance of the block is inserted in a drawing it is called by name, and its new insertion point, new scale and new rotation are defined.

Shape rule can be defined with a block on the LHS and one on the RHS. This makes the rule matching task much easier because the database of a drawing has two parts: first a list of blocks giving their names, scales, rotations and insertion points; and secondly definitions of the blocks in terms of their constituent lines. To match shape rules whose LHS's are blocks it is only necessary to scan the first part of the database for the relevant blocks. Every instance of the relevant blocks, and therefore every match of the LHS's of the rules, can be identified together with scales, rotations and insertion points.

Rule application is also simple, consisting of a block insertion using the scale, rotation and insertion point from the database

Furthermore there is no penalty in the graphical complexity of the blocks used in shape rules, since they are matched by name not by their constituent lines.

3. IMPLEMENTATION OF A "BLOCK" RULE INTERPRETER

3.1 AutoCAD-based shape rules

Although design with shape rules is very different from conventional CAD, Cambridge Architectural Research has implemented a shape rule interpreter as an enhancement of the well-established AutoCAD software. In principle the interpreter could be implemented as an enhancement to other CAD systems. The advantages of the incremental introduction of shape rules, rather than attempting a global redefinition of CAD, are obvious. The whole project for implementing shape rules has been given the name *The Design Engine*.

The first version of the interpreter relied on AutoCAD macro commands to perform the rule application task (shape deletion and insertion) [6]. The user had to perform rule matching and rule selection. Human error in these tasks often resulted in shapes that violated the the rules.

The interpreter was then re-implemented using the AutoLISP dialect of LISP that is linked to AutoCAD with full access to the drawings database and drafting commands. This dramatically increases the reasoning capability that can be built in. Rule matching becomes possible, as does checking for the legality of rule selections made by the user, thus making it impossible to generate illegal designs.

The Design Engine implementation adopts the decisive simplification described above (§2.3), by operating on named AutoCAD blocks rather than graphic entities in the drawing. The LHS and RHS of each shape rule are AutoCAD blocks; rule matching is performed by scanning the list of blocks in the AutoCAD drawing database.

The Design Engine generates standard AutoCAD drawings. It is thus possible to mix shape rule generated elements and conventional drafting, and to move freely between drafting and shape rules. For example, a basic design can be generated with shapes rules and then elaborated by using drafting commands; or designs can be generated with shape rules on an AutoCAD site plan.

3.2 Educational use

The first application of *The Design Engine* was for teaching in schools of architecture in the UK, where AutoCAD is widely used. There are two reasons for the use of shape rules in CAD teaching.

The first is to simplify the initial exposure of students to CAD and AutoCAD, as an alternative to the rather laborious learning of drafting commands. With shape rules students can generate finished, if simple, designs in their first CAD session, rewarding their enthusiasm and encouraging further use of CAD. Without shape rules the learning curve with complex CAD systems has been too steep, resulting in frustration and loss of interest on the part of students.

The second educational objective is to teach the principles of shape rules and shape grammars. This topic is appropriate to a course on architectural theory, and specifically on rational methodologies [eg.7].

3.3 Commercial use

The first commercial applications of shape rules by Cambridge Architectural Research have been for clients who construct buildings with a high degree of modularity and repetition. The building types are low-rise suburban office developments and budget-price hotels. CAR developed shape rules specific to the requirements of the building types, for use with the AutoCAD/AutoLISP shape rule interpreter.

The systems are used in the early design stage - for feasibility studies or sketch design exercises. Although the building types are modular and repetitive there are hundreds of possible building configurations, and many alternatives are routinely evaluated for particular development sites. Before the use of shape rules all sketch designs were done by hand, and CAD was not used until basic configurational decisions had been made.

Both applications generate building plans. There are three advantages in using shape rules:

- plans can be generated quickly
- plans can be generated on CAD by designers with no expertise in CAD drafting
- the plans are more reliable and accurate than non-CAD sketches or CAD-drafted plans

Even at this early stage in the commercial use of shape rules there are tangible benefits.

4. EXTENSIONS

4.1 Rule-writing and modification

We can identify some outstanding tasks in the development of general-purpose CAD using shape rules.

The first is the question of rule writing and modification. The shape rules implemented so far in *The Design Engine* have not been accessible to users: there has been a complete separation between the writing and use of shape rules. Whilst the importance of rule selection in designing with shape rules has been discussed (§2.1), the question of who writes the shape rules is also critical.

Many shape grammars have been written for established architectural styles: the rules are contrived to lead to predetermined conclusions. Users of these shape grammars are restricted to designing within the languages defined by the shape rules. Although most designers do work within established styles, they often introduce some personal features. This corresponds to customising a given set of shape rules, and a developed shape rule interpreter for CAD would have to allow users to modify the shape rules. Ultimately the modifications might lead to radically new types of design. The evolution of shape rules has been studied, for example, in the career of Frank Lloyd Wright [8].

Another common design strategy is to adopt ideas from different sources, and again this has its equivalent in hybrid shape grammars. To permit this approach the user must be able to select shape rules from a variety of grammars to form a new eclectic shape grammar.

An interesting situation arises if shape rules are written without a preconception about the designs to be generated. The user would experiment with the rules to discover what they generate. This could be an extremely fertile stimulus to creative design.

4.2 Goal-directed use of shape rules

When using shape rules the user controls design development by the selection of rules. But how does the user decide which rules to apply and where to apply them? When designing with complex shape grammars there is a risk that the consequences of rule applications at an early stage may not be self-evident; and the user may be reduced to trial-and-error. How can the user even be sure that the shape rules are actually capable of generating a solution to the problem in hand?

This problem is familiar in large rule-based expert systems where complex interaction between rules can lead to unpredictable behaviour. It is probable that the way that shape rules are written and structured in shape grammars will be vital for the usability of shape rule based CAD.

If there were a way of stating design objectives such that an appropriate sequence of shape rule applications could be automatically determined, by a structure of meta-rules, then we would have an automatic design system. This does not seem to be a realistic possibility and the user's intelligent selection of rules will remain critical: and therefore the user must be able to understand shape rules and their behaviour.

4.3 Diverse sets of shape rules

To respond to the variety of design contexts that occur in architectural practice there must be a multitude of shape grammars available to users. We envisage that shape rules will be written by many people in many places, for specific styles, building types, or design sub-tasks. Some rules will be for private use, by many will be published and distributed, rather as CAD drawings of building components are already distributed.

4.4 Integration with non-shape design knowledge

As we noted above (§1.1), the use of rules to represent architectural design knowledge pre-dates the idea of shape rules, and the range of design knowledge that could be represented in rule-based systems extends far beyond shapes or the visual aspects of architectural style. This is the territory of expert systems.

Most rule-based expert systems have a narrow domain of expertise, and a large number of systems will be required to model a significant part of the knowledge used everyday by practising architects, involving, for example, regulations, technical performance, functional performance, costs, etc. We suggest that the architectural CAD workstation and software will be the host environment for both geometrical and non-geometrical rule-based systems. To some extent, therefore, we see *The Design Engine* as prototypical in that it is built on AutoCAD, an established CAD environment.

An extended CAD environment with multiple expert systems will have to be an open system. The database describing the current design will have to include both geometrical and non-geometrical data, which rule-based systems will access for information of whatever type. The left hand sides of rules may have a mixture of geometrical and non-geometrical conditions, and similarly the actions may be hybrid.

Everyone would benefit if agreed standards could be established, but few things are harder to achieve than ageed standards.

5. CONCLUSIONS

Shape rules have received considerable attention amongst academic researchers, but are only just beginning to emerge as part of the CAD environment. They offer the prospect of a more fundamental approach to CAD, in which architects work with the structures that generate designs rather than simply manipulating the pieces of a design.

Shape rules can be seen as a representation of certain kinds design knowledge including, for example, knowledge about architectural style. Unlike traditional ways of representing this kind of knowledge in architectural theory or criticism, shape rules are operational and can be used to create new designs.

The representation of geometrical design knowledge with shape rules is only part of a more ambitious idea. It looks forward to a CAD environment in which all kinds of design knowledge will be represented using rules and other techniques pioneered in Artificial Intelligence.

6. BIBLIOGRAPHY

(1) C A Alexander, B Poyner
"The Atoms of environmental structure"
G T Moore (ed) *Emerging Methods in Environmental Design*
MIT Press, Cambridge, Mass., 1970

(2) C A Alexander, I Fiksdahl-King, S Ishikawa, M Silverstein
A Pattern Language
Oxford University Press, New York, 1977

(3) G Stiny
"An Introduction to shape and shape grammars"
Environment & Planning B, vol.7, pp.343-351, 1980

(4) G Stiny, W Mitchell
"The Palladian Grammar"
Environment & Planning B, vol.5, pp.5-18, 1978

(5) H Konig, J Eizenberg
"The Language of the Prairie"
Environment & Planning B, vol.8, pp.195-224, 1981

(6) W Fawcett
"An Elementary shape rule interpreter for architectural CAD"
A Smith (ed) *Knowledge Engineering and Computer Modelling in CAD*
Butterworths, London, 1986

(7) J Wojtowicz, W Fawcett
Architecture: Formal Approach
Academy Editions, London, 1986

(8) T W Knight
"Transformations of languages of designs"
Environment & Planning B, vol.10, pp.125-178, 1983