Open Research Online

The Open University's repository of research publications and other research outputs

Shapes, structures and shape grammar implementation

Journal Item

How to cite:

Jowers, Iestyn; Earl, Chris and Stiny, George (2019). Shapes, structures and shape grammar implementation. Computer-Aided Design, 111 pp. 80–92.

For guidance on citations see [FAQs](#).

© [not recorded]

<https://creativecommons.org/licenses/by-nc-nd/4.0/>

Version: Accepted Manuscript

Link(s) to article on publisher's website:

<http://dx.doi.org/doi:10.1016/j.cad.2019.02.001>

Copyright and Moral Rights for the articles on this site are retained by the individual authors and/or other copyright owners. For more information on Open Research Online's data [policy](#) on reuse of materials please consult the policies page.

oro.open.ac.uk

Shapes, structures and shape grammar implementation

Iestyn Jowers^{a}, Chris Earl^b, George Stiny^c*

a. The Open University, Milton Keynes, UK. e: iestyn.jowers@open.ac.uk

b. The Open University, Milton Keynes, UK. e: c.f.earl@open.ac.uk

c. MIT, Cambridge MA, USA. e: stiny@mit.edu

* corresponding author

Abstract. Shape grammars are a generative formalism in which dynamic changes to shape structure plays a vital role. Such changes support ambiguity and emergence, and as a result shape grammars are often used as the basis for proposed developments in supporting shape exploration in computer-aided design. However, the general implementation of shape grammars remains an unsolved problem, and a common solution is to adopt a fixed structure. This paper explores the consequences of assuming a fixed shape structure, via analysis of a simple shape grammar, often used as a benchmark problem to illustrate advances in shape grammar implementation. With reference to the combinatorics of words, it is proved that adopting a finite fixed structure limits the capability of a shape grammar. The paper concludes with a discussion exploring the implications of this result for shape grammar implementation and for design descriptions in CAD.

Keywords: shape structure, shape grammar implementation, combinatorics, computer-aided design

1. Introduction

Shape grammars [1] are a generative formalism, often used as the basis for proposed developments in supporting shape exploration in computer-aided design (CAD) and computer-aided conceptual design (CACD), e.g. [2-4]. Composed of shapes and shape rules, they are production systems where digital representations are manipulated as visual structures, and designs are generated via shape computations involving repeated application of rules. Research into interpreters that facilitate automated implementation of shape grammars spans many design disciplines, including engineering, architecture and product design; see [5] for a recent summary. But, despite some promising results, these implementations have not yet satisfactorily met the potential suggested by theoretical developments of the shape grammar formalism [1]. The visual nature of shape grammars means they are particularly suited to the early stages of design, where ambiguity and emergence play an important role [5, 6], but implementing the visual shape computations that result from applying shape rules has proven to be difficult within the logics of symbolic computation that underlie CAD and CACD systems [7, 8].

A key challenge lies in the specification of part structures which in an explorative activity, such as creative sketching, can change dynamically as a design concept evolves [6]. Part structures arise naturally as a result of interpreting a design representation, with different ways of seeing and understanding a concept giving rise to new avenues of exploration, and recognition of alternative embedded parts [9, 10]. Recreating this mechanism computationally in CACD can be difficult to achieve because the structures of CAD models have fixed topology, which can limit shape exploration [11, 12]. In CAD the underlying structures of shapes and their parts are expressed as

point sets which have regularity conditions to ensure that topological properties are preserved during shape operations and manipulations [13]. As a consequence, part structures are maintained as shapes change. But, parts also have a topological structure as embedded ‘individuals’ [14, 15]. In shape grammars, dynamic changes in structure are supported by a changing topology of shape which reflects the visual freedom apparent in sketching, with embedded parts recognised via rule application [16]. These differences in approach lead to an undesirable situation where implementation of shape grammars in CACD is necessarily limited by the logic of part structure supported by CAD, and can lead to restrictions, for example with respect to rule format and rule ordering [17].

In unrestricted shape grammars, as defined by Knight [17], a change in structure is realised via application of shape rules in which shape recognition is followed by application of shape operations to recognise and replace embedded parts, including parts that are emergent. It has previously been identified, e.g. [2, 18-20], that implementations of such shape grammars, e.g. via a general interpreter, should be able to accommodate the changes in structure that occur during shape computations. However, the problem of shape recognition is difficult, and for parametric shapes is NP-hard [7]. Instead, shape rule application is often made tractable by assuming a fixed part structure which focusses recognition on known parts [5, 17], and this paper investigates the consequences of such an assumption.

By focussing on the implementation of simple shape grammars the paper provides a formal proof that adopting a finite fixed structure limits the capability of a shape grammar, and can make it impossible to fully implement shape computations. Here, full implementation is in the sense that all possible rule applications are supported. The shape grammars investigated have previously been used to support arguments that in shape computations, shape structures should not be fixed, e.g. [21, 22, 1], and proving this conjecture strengthens the argument that computational shape interpretation and reinterpretation is essential for supporting creative exploration in CACD and CAD. Building on results presented in [23], it is shown that all but the simplest shape grammars become impossible to fully implement, when a fixed part structure is adopted, even if the rule applications can be finitely enumerated.

The paper proceeds in the next section by introducing a simple shape grammar composed of a single shape rule and an initial shape. All possible applications of the rule are enumerated and a part structure is identified to accommodate the resulting shape computation. In Section 3 this result is formalised and generalised with reference to the combinatorial structure of words [24]. In Section 4 two variations of the shape grammar are considered, in which different spatial arrangements and scales are explored. Based on these explorations, it is proved that, except for a few well-defined cases, no structure can be found that will fully accommodate the resulting shape computations. Finally, in Section 5 these results, and their implication for shape grammar implementation and for design descriptions in CAD are discussed.

2. Implementing a simple shape grammar

Figure 1 shows an example of a simple shape grammar that is often used as a benchmark problem to illustrate different aspects of shape grammar research, including developments in implementation [2, 19], user-interface [3, 20], and alternative formalisms [4, 25]. The grammar is composed of a single shape rule (Figure 1a), and an initial shape (Figure 1b). In shape grammars, application of a rule proceeds by recognising the shape on the left-hand side of the rule embedded as part of a

shape. The recognised part is then removed and replaced by the shape on the right-hand side of the rule. In this example, the shape on the left-hand side is a square, and the shape on the right-hand side is the same square rotated through an angle of 45° about its central point.

Figure 1. A simple shape grammar

Repeatedly applying rules gives rise to a lattice of shapes which represents a visual shape computation, as illustrated in Figure 2. For this shape grammar, the resulting shape computation is finite, including seven distinct shapes, each of which is generated from the initial shape via repeated application of the shape rule. Arrows indicate application of rule, with a double-headed arrow indicating that the rule is reversible. Despite its simplicity, implementing this computation in CACD software via application of the shape rule is difficult due to the necessity to recognise all squares (as left-hand side shapes) including any which are emergent, not having been identified as parts of the initial shape. It is for this reason that this simple shape grammar serves as a useful benchmark in shape grammar research.

Figure 2. Application of the shape grammar in Figure 1

The role emergence plays in the computation in Figure 2 can be identified by considering the different part structures required to execute rule applications. Applying the rule to the small square requires a different structure than applying it to the large squares, and a third structure is required after rotating both large squares so that line segments overlap. This is a problem previously considered in [21, 22], and in [1] a conjecture was proposed that the shape computation in Figure 2 gives rise to a part structure where the maximal lines of the shape in Figure 1b, i.e. the edges of the large square, are decomposed into parts according to a periodic palindrome **ABABAABABA**, where **A** and **B** are line segments, and their lengths are such that $B = A\sqrt{2}$. Here, line segments are denoted by bold upper-case letters, while the length of a line is denoted by the corresponding non-bold letter.

To derive all seven possible shapes and accommodate all possible rule applications, the shape computation, illustrated in Figure 2 requires three distinct decompositions of the initial shape. These are illustrated in Figure 3, where separation between the lines that compose the shapes are exaggerated. Figure 3a illustrates the structure that results from recognising the small square. The

maximal lines of the shape have been cut to accommodate the edges of the small square which are composed of lines labelled **X**. Figure 3b illustrates the structure that results from recognising the large squares, which have edges composed of lines labelled **Y**; these are the maximal lines of the shape. Figure 3c illustrates the structure that results from recognising the two large squares after they have been rotated. The squares overlap and share line segments which are labelled **P**. The lengths of **X** and **P** can be expressed relative to the length of **Y** as follows: $X = \frac{1}{2}Y$ and $P = (1 - \frac{1}{2})Y$.

Figure 3. Part structures necessary to accommodate the shape computation in Figure 2

Accommodating all three of these decompositions in a single shape representation would make it possible to implement the shape grammar in Figure 1, since all the parts recognised during the shape computation would be included. However, this is not only a matter of decomposing the edge of the large square so that it accommodates the smaller embedded lines. Multiple possible rule applications under symmetries on the squares (reflection and rotation) requires that decompositions of the edges of both large and small squares are symmetrical. Also, decompositions of the lines **P** in Figure 3c need to be symmetrical in order to accommodate their embedding as parts of the edges of the large squares. Taking these constraints into consideration gives rise to a part structure which accommodates all applications of the rule in Figure 1a. Derivation of this structure is illustrated in Figure 4, which for clarity focusses on one edge of a large square. Also for clarity, triangles are added to this edge to illustrate part structures while highlighting their symmetry. Each triangle corresponds to a line segment embedded in the edge, and these are subdivided into finer structures, representing lines embedded in lines. Embedded lines associated with triangles are symmetrical, and their subdivision into embedded parts is symmetrical; in this sense, the triangles represent the structure of the edge as visual palindromes. To accommodate the symmetry of the squares, all of the maximal lines in the shape should be decomposed according to the same structure.

Figure 4. Deriving a part structure to accommodate the shape grammar in Figure 1

Figure 4a illustrates the high-level structure of the edges of the large squares, which are the maximal lines of the shape. To account for the symmetric properties of the squares, a line **Y** is identified as a

visual palindrome, represented by a triangle. Figure 4b illustrates the embedding of the edge of the small square in the edge of the large square; the structure of the line **Y** now incorporates an embedded line **X** which is the length of the edge of the small square, represented by the triangle highlighted in grey. Figure 4c illustrates the embedding of the overlap in the edges of the squares; the structure of the lines **X** and **Y** now incorporate an embedded line **P** which is the length of the overlap, represented by the triangle highlighted in grey. This new structure breaks the symmetry of the line **X**, which is addressed in Figure 4d by reflecting the smaller triangle in the illustrated axis of symmetry of the larger triangle. A new triangle is defined by the overlap, and this represents further subdivision of the lines **Y**, **X** and **P**. Figure 4e resolves the symmetry of the line **P** by reflecting the emergent triangle in the illustrated axes of symmetry. Similarly, Figure 4f resolves the symmetry of the line **X** according to the illustrated axes of symmetry. Finally, in Figure 4g, the symmetry of the line **Y** is resolved to give a subdivision of the maximal lines of the shape that accommodates embedding **P** in **X** and **X** in **Y**, whilst also taking into account the symmetries of the large and small squares. The result is a periodic palindromic structure where the line **P** is decomposed into three parts of two different lengths, say **ABA**, the line **X** is decomposed into five parts of two different lengths, **ABABA**, and the line **Y** is decomposed into ten parts of different lengths, **ABABAABABA**. This periodic palindromic structure is the same as the structure identified in [1], where it was noted that a consequence of this decomposition is that the edges of the large squares and the small square have different part structures. This can be taken into consideration in the shape grammar by including two versions of the rule in Figure 1a: one which recognises squares with edges decomposed according to the periodic palindromic structure **ABABA** (i.e. the small square), and another which recognises squares with edges decomposed according to the periodic palindromic structure **ABABAABABA** (i.e. the large squares). So, even though the shape computation in Figure 2 is simple, including only seven distinct shapes, implementing it according to a fixed part structure has resulted in a decomposition of the initial shape into increasingly finer constituents [1].

3. Formalising the structure of a simple shape grammar

The periodic palindromic structures identified in Section 2 provide a symbolic representation for the edges of the shapes, and these can be analysed by referring to the combinatorial structure of words [24]. In combinatorics, a finite set of symbols is said to be an alphabet, denoted Σ , and words are sequences (either finite or infinite) of symbols from Σ . Adopting a standard notation, words are denoted by bold upper-case letters, for example **A** and **B**, and the empty word is denoted ϵ . The length of a word, which is the number of symbols in a word, is denoted by non-bold upper-case letters, for example $|A| = A$ and $|B| = B$, so $|\epsilon| = \epsilon = 0$. The constituent symbols of words are denoted by bold lower-case letters, for example $A = a_1 \dots a_A$ and $B = b_1 \dots b_B$ are words over Σ if the symbols a_i and b_j are members of Σ , with $1 \leq i \leq A$ and $1 \leq j \leq B$, for some integers A and B . Two words, **A** and **B**, are equal if they are composed of the same symbols, so $A = B$, if $A = B$ and $a_1 = b_1, a_2 = b_2, \dots, a_A = b_B$.

The set of all finite words, denoted Σ^* , is generated by Σ under an associative operation defined by concatenation. For example, if $A = a_1 \dots a_A$ and $B = b_1 \dots b_B$, are words over Σ , then they are members of Σ^* , and the word $A \cdot B = AB = a_1 \dots a_A b_1 \dots b_B$ is also a member of Σ^* . Σ^* is a free monoid with ϵ as the identity element under concatenation, i.e. $A \cdot \epsilon = A = \epsilon \cdot A$. Repeated self-concatenation is denoted by superscript, for example $A^3 = A \cdot A \cdot A = AAA = a_1 \dots a_A a_1 \dots a_A a_1 \dots a_A$. The length of a word is recursively defined under concatenation as follows: given that ϵ has length 0, and any symbol a in Σ has length 1, for any word **A** in Σ^* and any letter a in Σ , $|Aa| = |A| + 1 = A + 1$. Using these

definitions it is possible to formalise the part structure of shapes, with reference to analogous parts of words.

3.1 The combinatorial structure of embedded words

A word that is embedded as part of a second word is either a factor, a prefix or a suffix. For two words A and B , these are defined as follows. A is a factor of B if the words X and Y exist in Σ^* such that $B = XAY$. A is a proper factor of B if $X \neq \epsilon$ and $Y \neq \epsilon$. A is a suffix of B if there exists a word X in Σ^* such that $B = XA$ and A is a proper suffix of B if $X \neq \epsilon$. Similarly, A is a prefix of B if there exists a word Y in Σ^* such that $B = AY$ and A is a proper prefix of B if $Y \neq \epsilon$.

The periodic structure derived in Figure 4 arises readily in combinatorics, when words with identified parts are equated with each other, as summarised in the following lemmas, proved in [26], and illustrated in Figures 5 and 6. The first of these is concerned with the structure that arises when two words with distinct parts are equated with each other, and the second is concerned with the structure that arises when two such words share a common part, identified as the prefix of one word and the suffix of the other.

Lemma 1. If A , B , C and D are words in Σ^* , such that $AB = CD$ and $A \leq C$, then there exists a word V in Σ^* such that $C = AV$ and $B = VD$.

This is illustrated in Figure 5, where words are represented by rectangles.

Lemma 2. If A , B and C are words in Σ^* , such that $AB = BC$ and $A \neq \epsilon$, then $A = UV$, $B = (UV)^k U$ and $C = VU$ for some words U , V in Σ^* , and integer $k = \lceil B/A \rceil - 1$.

Here, $\lceil \cdot \rceil$ represents a ceiling operator, so that $\lceil B/A \rceil$ is the smallest integer greater than B/A . This result is illustrated in Figure 6 where words are represented by rectangles.

Figure 5. Illustration of Lemma 1, with words represented by rectangles

The combinatorial structures that result from Lemma 2 are analogous to the part structures derived in Figure 4, which arise from the symmetric embedding of lines in lines, P in X and X in Y . This is because the symmetry of the squares requires that edges with identified parts be equal to their mirror images, which themselves contain the same parts. But, this condition of symmetry is stronger than the conditions under which Lemma 2 holds, and the structures of embedded lines can be further explored by considering words under similar symmetry constraints. As one-dimensional strings, the symmetry group of words is defined by one-dimensional translation and reflection, where translation produces periodic words and reflection produces palindromic words, as follows.

An infinite sequence $S = s_1 s_2 \dots$ is called periodic if there exists an integer $p \geq 1$, called a period, such that for each $n \geq 0$, $s_{n+p} = s_n$, so a finite word A in Σ^* is called periodic if there exists an integer $p \geq 1$, such that A is a prefix of an infinite sequence with period p . The period of A is the smallest such integer. For example, the word **abaaba**, has periods of 3, 5, and 6, and its period is 3. Lemma 2 shows that equal words with common parts are intrinsically periodic in nature, with a period of $U + V = A (= C)$, and the same is true for lines embedded in lines, as illustrated in Figure 4.

If $\mathbf{A} = \mathbf{a}_1 \dots \mathbf{a}_A$ is a word over Σ^* , then $\overline{\mathbf{A}} = \mathbf{a}_A \dots \mathbf{a}_1$ is the word obtained by reflecting $\mathbf{A}$, i.e. reading $\mathbf{A}$ backwards. A palindrome is a word that is equal to its reflection, so word $\mathbf{A}$ is a palindrome if $\mathbf{A} = \overline{\mathbf{A}}$ with $\mathbf{a}_1 = \mathbf{a}_A$, $\mathbf{a}_2 = \mathbf{a}_{A-1}$, For example, **racecar** and **rotavator** are examples of palindromes in the English language, and trivially, the empty word ϵ , all words of length 1, and all words of the form $\mathbf{a}^i$ (for some integer i), are also palindromes.

a) $A \geq B$

b) $A < B$, with inductive loop highlighted

Figure 6. Illustration of Lemma 2, with words represented by rectangles

In the shape computation in Figure 2, the symmetry of the squares requires that decompositions of edges are palindromic, which in Lemma 2, equates to the requirement that $\mathbf{B}$ is palindromic and that $\mathbf{AB} = \mathbf{BC} = \mathbf{W}$ is also palindromic. This requirement imposes further constraints on the structure of embedded parts. In particular, $\mathbf{A}$ and $\mathbf{C}$ are reflections of each other because if $\mathbf{W} = \overline{\mathbf{W}}$ then $\mathbf{AB} = \overline{\mathbf{CB}}$ and $\mathbf{B} = \overline{\mathbf{B}}$ gives $\mathbf{A} = \overline{\mathbf{C}}$. Also, $\mathbf{U}$ and $\mathbf{V}$ are palindromic because if $\mathbf{B} = \overline{\mathbf{B}}$ then $(\mathbf{UV})^k \mathbf{U} = \overline{\mathbf{U}(\mathbf{VU})^k} = (\overline{\mathbf{UV}})^k \overline{\mathbf{U}}$ so that $\mathbf{U} = \overline{\mathbf{U}}$ and $\mathbf{V} = \overline{\mathbf{V}}$.

3.2 The combinatorial structure of embedded lines

In light of these conditions, Lemma 2 accounts for the periodic palindromic structure derived in Figure 4, with words and their parts analogous to lines and their parts. To make this explicit, let $\mathbf{W}$ represent a line of length W , $\mathbf{A}$ represent a line of length A , and $\mathbf{B}$ represent a line of length B . Embedding $\mathbf{B}$ in $\mathbf{W}$, such that $\mathbf{B}$ and $\mathbf{W}$ share an end point, $\mathbf{W} = \overline{\mathbf{W}}$ and $\mathbf{B} = \overline{\mathbf{B}}$, gives rise to a palindromic periodic structure,

$$\mathbf{W} = \mathbf{AB} = \mathbf{UV}(\mathbf{UV})^k \mathbf{U} = (\mathbf{UV})^{k+1} \mathbf{U} \quad (\text{Equation 1})$$

where $\mathbf{U}$ is a line of length U , $\mathbf{V}$ is a line of length V and k is an integer given by $\lceil B/A \rceil - 1$. The period of this structure is $U + V = A$, and since $\mathbf{W} = \mathbf{AB} = (\mathbf{UV})^{k+1} \mathbf{U}$ and $\mathbf{B} = (\mathbf{UV})^k \mathbf{U}$, the lengths W and B can be written as

$$W = (k+2)U + (k+1)V$$

$$B = (k+1)U + kV$$

$$k/(k+1) \leq B/W \leq (k+1)/(k+2)$$

and it follows that

$$U = (k+1)B - kW \quad (\text{Equation 2})$$

$$V = (k+1)W - (k+2)B \quad (\text{Equation 3})$$

Examples of the structures arising from embedding a line **B** in a line **W**, with **B** and **W** sharing an endpoint, are illustrated in Figure 7, where different configurations of two lines are explored. The lines under consideration are edges of squares and the configurations are constrained such that in each one an edge of a small square is embedded in an edge of a large square, with both sharing an end point. Edge lengths are labelled according to Equation 1, and W , the edge length of the large squares, is kept constant while B , the edge lengths of the small squares increases (with corresponding increase in k as B/W increases) from Figure 7a to 7h. Again, triangles are included to illustrate the resulting part structures of the shared edges, whilst highlighting their symmetry.

In Figure 7a, $B < \frac{1}{2}W$ so that $k = 0$, and embedding the short edge in the long edge results in a part structure that might be intuitively expected: the structure of the short edge remains unchanged and the structure of the long edge includes the short edge as an embedded part. As a result, the structure of the short edge can be described by the word **U**, which represents a line of length of $U = B$, and the structure of the long edge can be described by the word **UVU** where **V** represents a line of length $V = W - 2A$. Increasing the edge length of the small square results in an increase in the length U , and a decrease in the length V . Specifically, as $B \rightarrow \frac{1}{2}W$, $U \rightarrow \frac{1}{2}W$ and $V \rightarrow 0$, and, in Figure 7b, when $B = \frac{1}{2}W$, $k = 0$, $V = 0$ and the long edge can be described by the word **UU**.

In Figures 7c-h, $B > \frac{1}{2}W$ and the embedded short edges overlap resulting in the emergence of more complicated structures, as illustrated in Figure 4d. When $B > \frac{1}{2}W$, $k > 0$ and embedding the short edge in the long edge results in a decomposition of both edges and, as B and k increase, the symbolic descriptions of the resulting part structures can be categorised according to the following cases:

- In Figure 7c, $\frac{1}{2}W < B < \frac{3}{4}W$, so that $k = 1$, and the short edge can be described by **UVU** and the long edge by **UVUVU**. As $B \rightarrow \frac{3}{4}W$, $U \rightarrow \frac{3}{4}W$ and $V \rightarrow 0$
- In Figure 7d, $B = \frac{3}{4}W$, $k = 1$, $U = \frac{3}{4}W$ and $V = 0$, the short edge can be described by **UU** and the long edge by **UUU**
- In Figure 7e, $\frac{3}{4}W < B < \frac{5}{8}W$, so that $k = 2$, and the short edge can be described by **UVUVU** and the long edge by **UVUVUVU**. As $B \rightarrow \frac{5}{8}W$, $U \rightarrow \frac{5}{8}W$ and $V \rightarrow 0$
- In Figure 7f, $B = \frac{5}{8}W$, $k = 2$, $U = \frac{5}{8}W$ and $V = 0$, the short edge can be described by **UUU** and the long edge by **UUUU**
- In Figure 7g, $\frac{5}{8}W < B < \frac{3}{4}W$, so that $k = 3$, and the short edge can be described by **UVUVUVU** and the long edge by **UVUVUVUVU**, and as $B \rightarrow \frac{3}{4}W$, $U \rightarrow \frac{3}{4}W$ and $V \rightarrow 0$
- In Figure 7h, $B = \frac{3}{4}W$, $k = 3$, $U = \frac{3}{4}W$ and $V = 0$, the short edge can be described by **UUUU** and the long edge by **UUUUU**

Figure 7. Part structures resulting from two squares sharing a common edge

The pattern identified here continues, tending towards the limiting case where $B = W$ and the two squares are equal, with the edges represented by a single line. But, as $B \rightarrow W$, $k \rightarrow \infty$ and $U \rightarrow 0$, and the part structure of the edges gets finer and finer with the number of line segments increasing. This structure is defined according to line segments of two alternating lengths, and it can always be described as a periodic palindrome over the two words, $\mathbf{U}$ and $\mathbf{V}$. In general, the structure of the shorter edge can be described by the word $(\mathbf{UV})^k \mathbf{U}$, and the structure of the longer edge can be described by the word $(\mathbf{UV})^{k+1} \mathbf{U}$, where $\mathbf{U}$ and $\mathbf{V}$ represent lines of length $U = (k+1)B - kW$ and $V = (k+1)W - (k+2)B$, respectively, and $k = \lceil B/A \rceil - 1$, with $A = W - B$.

3.3 The combinatorial structure of embedded symmetric lines

These results can be applied to the shape grammar in Figure 1, by considering the pairwise embedding of lines in lines, $\mathbf{X}$ in $\mathbf{Y}$, $\mathbf{P}$ in $\mathbf{X}$ and $\mathbf{P}$ in $\mathbf{Y}$, where $\mathbf{X}$, $\mathbf{Y}$ and $\mathbf{P}$ denote the line segments identified in Figure 3. The results are illustrated in Figure 8, where for the sake of legibility, Figure 8b is not drawn to the same scale as Figures 8a and c.

Figure 8. Pairwise combinations of the part structures identified in Figure 3

Firstly, when X is embedded in Y , from Equation 1 the structure of Y is given by $(U_{YX}V_{YX})^{i+1}U_{YX}$, and the structure of X is given by $(U_{YX}V_{YX})^iU_{YX}$. Here, U_{YX} and V_{YX} are line segments with the subscript YX denoting that the structure results from X embedded in Y , and i is an integer given by $i = \lceil X/(Y-X) \rceil - 1$. In Figure 3, the relative lengths of X and Y are given by $Y = 2X$, and so $i = 0$, and the structure is analogous to those in Figures 7a, with $X = U_{YX}$ and $Y = U_{YX}V_{YX}U_{YX}$. From Equations 2 and 3, the lengths of U_{YX} and V_{YX} are

$$U_{YX} = (i+1)X - iY = X$$

$$V_{YX} = (i+1)Y - (i+2)X = Y - 2X = 0$$

This gives, $X = U_{YX}$ and $Y = U_{YX}U_{YX} = XX$, as illustrated in Figure 8a.

Similarly, in Figure 3 the relative lengths of X and P are given by $P = (2 - \sqrt{2})X$, so from Equation 1, P embedded in X , gives $P = (U_{XP}V_{XP})^jU_{XP}$, and $X = (U_{XP}V_{XP})^{j+1}U_{XP}$ for $j = \lceil P/(X-P) \rceil - 1 = 1$. Therefore, the structure is analogous to those in Figure 7c, with $P = U_{XP}V_{XP}U_{XP}$ and $X = U_{XP}V_{XP}U_{XP}V_{XP}U_{XP}$, as illustrated in Figure 8b. From Equations 2 and 3, the lengths of U_{XP} and V_{XP} are

$$U_{XP} = (j+1)P - jX = (3 - 2\sqrt{2})X$$

$$V_{XP} = (j+1)X - (j+2)P = (3\sqrt{2} - 4)X$$

Finally, the relative lengths of P and Y are given by $2P = (2 - \sqrt{2})Y$, so from Equation 1, P embedded in Y , gives $P = (U_{YP}V_{YP})^kU_{YP}$ and $Y = (U_{YP}V_{YP})^{k+1}U_{YP}$, for $k = \lceil P/(Y-P) \rceil - 1 = 0$. Therefore, the structure is analogous to those in Figures 7a, with $P = U_{YP}$ and $Y = U_{YP}V_{YP}U_{YP} = PV_{YP}P$, as illustrated in Figure 8c. From Equations 2 and 3, the lengths of U_{YP} and V_{YP} are

$$U_{YP} = (k+1)P - kY = P = (2 - \sqrt{2})X$$

$$V_{YP} = (k+1)Y - (k+2)P = Y - 2P = (\sqrt{2} - 1)Y = 2(\sqrt{2} - 1)X$$

The three structures in Figure 8 result from pairwise combinations of the part structures in Figure 3, which are necessary to implement the shape grammar in Figure 1. In order to accommodate the full computation in Figure 2, these three structures need to be combined to derive a decomposition of Y in which both X and P are embedded, while retaining the symmetric properties of the squares. The resulting structure can be identified by considering the symbolic representations of the structure of Y and recognising that $Y = XX$ while $X = U_{XP}V_{XP}U_{XP}V_{XP}U_{XP}$, so that $Y = U_{XP}V_{XP}U_{XP}V_{XP}U_{XP}U_{XP}V_{XP}U_{XP}V_{XP}U_{XP}$. This is shown visually in Figure 9, where Y , X , P , V_{YP} , U_{XP} and V_{XP} are represented by rectangles. The resulting structure is equivalent to the structure identified in Section 2, where Y was decomposed according to $ABABAABABA$, with $B = A\sqrt{2}$.

Y :	X				X			
Y :	P		V_{YP}				P	
Y :	U_{XP}	V_{XP}	U_{XP}	V_{XP}	U_{XP}	V_{XP}	U_{XP}	V_{XP}

Figure 9. Combinatorial structure of Y , with lines represented by rectangles

This formal derivation has proven the conjecture presented in [1], and the part structure necessary to implement the shape grammar in Figure 1, is illustrated in Figure 10, where Figure 10a shows how the structure applies to the three lines X , Y and P , while Figure 10b shows how it accommodates the shape computation in Figure 2. In Figure 10b, the separation between the line segments are exaggerated for illustrative purposes, but this also highlights how the initial shape has been decomposed into shape into finer and finer constituents [1].

Figure 10. The part structure of two overlapping squares, with X , Y and P identified

4. Formalising the structures of modified shape grammars

Variations of the simple shape grammar in Figure 1 can be explored by changing the initial shape. In this section two such variations are considered, according to the alternative initial shapes illustrated in Figure 11, both of which are straightforward modifications of the original initial shape. In Figure 11a the spatial arrangement between the two large squares has been changed; and in Figure 11b an additional square on a different scale has been included. In both cases, the resulting shape computation is finite but, as will be shown, in both cases these changes to the initial shape result in shape computations that cannot be fully implemented using a fixed part structure.

Figure 11. Alternative initial shapes for the shape grammar in Figure 1

4.1 Consequences of changing the spatial arrangement

In the initial shape in Figure 11a, the spatial arrangement between the two large squares has been changed in such a way that the distance between their centres is shorter than in Figure 1b, but their centres continue to lie on a diagonal at an angle of 45° . This arrangement ensures that the small square is defined in the overlap. The size of this square is dependent on the distance between the centres of the large squares and, under this spatial arrangement, the lengths of the lines X and P , relative to the length of Y are measured according to the distance d between the centres of the two large squares, as illustrated in Figure 12.

Figure 12. Relative lengths of X , Y and P defined according to d

The parametric shapes in Figure 12 describe a family of shape grammars, in which the initial shape is analogous to the shape in Figure 1b, but with the distance between the centres of the large squares given by the parameter d . In general, it can be shown that $X = Y - \sqrt{2}d$ and $P = Y - d$, so that as d decreases X and P increase. The different values for d gives rise to seven different classes of shape grammar, differentiated by the fixed structure of shape necessary to implement the resulting shape grammar, as follows for decreasing values of d :

1. When $d \geq \sqrt{2}Y$, the two large squares do not overlap, so that $X = 0$, $P = 0$ and X and P are not defined. Y is not subdivided, so $Y = Y$, and the shape computation consists of four distinct shapes which result from applying the shape rule in Figure 1a to the large squares and there is no decomposition of the maximal lines of the shape.
2. When $Y \leq d < \sqrt{2}Y$, the two large squares overlap, but the rotated squares do not, so that $0 < X \leq (1 - \sqrt{2})Y$, $P = 0$, and X is defined but P is not. As a result X is not subdivided, so $X = X$, and since $X < \frac{1}{2}Y$ the structure of Y is analogous to those illustrated in Figure 7a, and $Y = XV_{YX}X$. When $Y < d < \sqrt{2}Y$ the shape computation consists of five distinct shapes, but when $d = Y$ it consist of seven distinct shapes.

In the remaining cases $d < Y$, and the rotated squares overlap so that X and P are both defined, and the resulting shape computations are analogous to the computation illustrated in Figure 2. They each consist of seven distinct shapes, as illustrated in Figure 13, where the rule in Figure 1a is applied repeatedly to the shape in Figure 11a.

Figure 13. Application of shape rule in Figure 1a to shape in Figure 11a

3. When $0 < P < \frac{1}{2}X$ then $\frac{1}{2}(4 + \sqrt{2})Y \leq d < Y$ and $(1 - \sqrt{2})Y < X < \frac{1}{2}(6 - 2\sqrt{2})Y$. The structure of P embedded in X is analogous to the structures illustrated in Figure 7a, P is not subdivided, so $P = P$ and $X = PV_{XP}P$. Also, since $X < \frac{1}{2}Y$ the structure of X embedded in Y is analogous to the structures illustrated in Figure 7a, so $Y = XV_{YX}X = PV_{XP}PV_{YX}PV_{XP}P$.

4. Similarly, when $P = \frac{1}{2}X$ then $d = \frac{1}{2}(4 + \sqrt{2})Y$, $X = \frac{1}{2}(6 - 2\sqrt{2})Y$ and $P = \frac{1}{2}X$. The structure of P embedded in X is analogous to the structure illustrated in Figure 7b, P is not subdivided, so $P = P$ and $X = PP$. Also, since $X < \frac{1}{2}Y$ the structure of X embedded in Y is analogous to the structures illustrated in Figure 7a, so $Y = XV_{YX}X = PPV_{YX}PP$.
5. When $\frac{1}{2}(6 - 2\sqrt{2})Y < X < \frac{1}{2}Y$ then $\sqrt{2}Y < d < \frac{1}{2}(4 + \sqrt{2})Y$ and $\frac{1}{2}(3 - \sqrt{2})Y < P < (1 - \sqrt{2})Y$. Since $\frac{1}{2}X < P < \frac{2}{3}X$ the structure of P embedded in X is analogous to the structures illustrated in Figure 7c, so $P = U_{XP}V_{XP}U_{XP}$, and $X = U_{XP}V_{XP}U_{XP}V_{XP}U_{XP}$. Also, since $X < \frac{1}{2}Y$ the structure of X embedded in Y is analogous to the structures illustrated in Figure 7a, so $Y = XV_{YX}X = U_{XP}V_{XP}U_{XP}V_{XP}U_{XP}V_{YX}U_{XP}V_{YX}U_{XP}V_{XP}U_{XP}$.
6. Similarly, when $X = \frac{1}{2}Y$ then $d = \sqrt{2}Y$ and $P = (1 - \sqrt{2})Y$. This is the initial shape illustrated in Figure 1b, and as shown in Section 3, the structure of P embedded in X is analogous to the structures illustrated in Figure 7b, so $P = U_{XP}V_{XP}U_{XP}$, and $X = U_{XP}V_{XP}U_{XP}V_{XP}U_{XP}$, and the structure of X embedded in Y is analogous to the structure illustrated in Figure 7b, so $Y = XX = U_{XP}V_{XP}U_{XP}V_{XP}U_{XP}U_{XP}V_{XP}U_{XP}V_{XP}U_{XP}$.
7. Finally, when $d < \sqrt{2}Y$, $X > \frac{1}{2}Y$, and $P > (1 - \sqrt{2})Y$. Since $P > \frac{1}{2}X$, the structure of P embedded in X is analogous to the structures illustrated in Figure 7c-h and beyond, for larger B , so $P = (U_{XP}V_{XP})^i U_{XP}$ and $X = (U_{XP}V_{XP})^{i+1} U_{XP}$. Also, since $X > \frac{1}{2}Y$ the structure of X embedded in Y is analogous to the structures illustrated in Figure 7c-h and beyond, for larger B , so $X = (U_{YX}V_{YX})^i U_{YX}$ and $Y = (U_{YX}V_{YX})^{i+1} U_{YX}$. In order to implement the full computation, it is necessary to combine these three structures to derive a decomposition of Y in which both X and P are embedded, while retaining the symmetric properties of the squares. This means that the two distinct decompositions for X , $X = (U_{XP}V_{XP})^{i+1} U_{XP}$ and $X = (U_{YX}V_{YX})^i U_{YX}$, (and by extension the distinct decompositions of Y and P) need to be resolved in a single periodic palindromic structure. However, such a structure does not exist.

For example, class-7 includes the initial shape illustrated in Figure 11a, and the resulting shape computation is illustrated in Figure 13. Figure 14 illustrates pairwise combinations of the part structures required to implement this shape computation, and is analogous to Figure 8. For the sake of legibility, Figure 14b is not drawn to the same scale as Figures 14a and c, and subscripts of U and V have not been included. Resolving these part structures into a single structure that will accommodate the full shape computation is not possible.

Figure 14. Pairwise combinations of part structures necessary for the computation in Figure 13

The reason why a structure does not exist for the class-7 grammars results from the continuous nature of lines, a consequence of which is that, although it is useful to use combinatorics of words to identify the structures of embedded shapes, the analogy between shapes and words is limited. In the structures under consideration, the analogy breaks down with respect to the nature of the length of periods. From Equation 1, the structure $W = (UV)^{k+1}U$ has a period of $U + V$, and from

Equations 2 and 3, $U + V = W - B$. So, the structure that results from embedding X in Y has a period of $U_{YX} + V_{YX} = Y - X = \sqrt{1/2}d$, and the structure that results from embedding P in X has a period of $U_{XP} + V_{XP} = X - P = (1 - \sqrt{1/2})d$. However, irrational periods are not possible for combinatorial words U_{XP} , V_{XP} , U_{YX} and V_{YX} which are composed of discrete symbols instead of continuous lines. For X as a continuous line, these two incommensurate periods mean that X (and by extension Y and P) cannot be decomposed into a finite set of line segments. Establishing this is straightforward: if a line A has two periodic decompositions such that $A = B^i = C^j$, for line segments B and C and integers i and j , then $A = iB = jC$, and $j/i = B/C$, which establishes that B and C , the periods of A must be commensurate. That there is no finite decomposition of X is supported by a general theorem [27], which establishes that a continuous periodic function with two incommensurate periods is constant with effectively zero period, as stated in Lemma 3.

Lemma 3. If f is a continuous function with two incommensurate periods, then f is constant.

Lemma 3 implies that X has no finite decomposition, and its period can be regarded as infinitesimally small. In the context of implementing these shape grammars, incommensurate periods place similar conditions on the periodicity of the structures of Y (that result from embedding X in Y) and P (that result from embedding P in X). This means that, for all the (infinitely) many grammars included in class 7, there is no fixed structure that will accommodate the embedding of both X and P , whilst retaining the symmetric properties of the squares. Consequently, the class-7 shape grammars cannot be fully implemented if a fixed structure is assumed.

4.2 Consequences of changing scale

For the modified initial shape in Figure 11b, an additional square has been added which retains the spatial arrangement of the original but introduces a new square at a smaller scale. Applying the shape rule in Figure 1b to the shape in Figure 11b gives rise to the shape computation illustrated in Figure 15. The shapes produced are stylistically very similar to those in Figure 2, but with more squares embedded in the initial shape there are more opportunities to apply the rule.

The resulting computation is again finite, including only twenty-four distinct shapes, and to derive all of these shapes and accommodate all possible rule applications requires five distinct decompositions of the initial shape. These are illustrated in Figure 16, where separation between the lines that compose the shapes are exaggerated. Figure 16a illustrates the structure that results from recognising the small square. The maximal lines of the shape have been cut to accommodate the edges of the small square which are composed of lines labelled X . Figure 16b illustrates the structure that results from recognising the medium squares, and the maximal lines have been cut to accommodate the edges composed of lines labelled Y . Figure 16c illustrates the structure that results from recognising the large squares, which have edges composed of lines labelled Z ; these are the maximal lines of the shape. Figure 16d illustrates the structure that results from recognising the two medium squares after they have been rotated. The squares overlap and share line segments which are labelled P . Similarly, in Figure 16e, the rotated large squares overlap and share line segments labelled Q . The lengths of X , Y , P and Q can be expressed according to the length of Z as follows: $X = \frac{1}{4}Z$, $Y = \frac{1}{2}Z$, $P = \frac{1}{2}(1 - \sqrt{1/2})Z$ and $Q = (1 - \sqrt{1/2})Z$.

Figure 15. Application of shape rule in Figure 1a to shape in Figure 11b

Accommodating all five of these decompositions in a single shape representation would make it possible to implement the shape computation illustrated in Figure 15. As in Section 3, the resulting part structure can be identified by considering the decompositions that arise from pair-wise combinations of the line segments, **X**, **Y**, **Z**, **P**, and **Q**, with lines embedded as parts of other lines. There are ten different line-in-line combinations to consider, and the resulting part-structures are illustrated in Figure 17. For the sake of legibility, these are not all drawn to the same scale.

Figure 16. Part structures necessary to accommodate the shape computation in Figure 15

From arguments supported by Lemma 3, if these different part structures are to be combined in a single structure to accommodate the full shape computation then their periods must be commensurate. However, they are not, and for simplicity this can be shown by considering only the structures that result from embedding the lines Q and X in the line Y , i.e. from combining the structures in Figure 17e and f. Since $\frac{1}{2}Y < Q < \frac{3}{2}Y$, the structure in Figure 17e is analogous to the structures in Figure 7c, so that $Y = U_{YQ}V_{YQ}U_{YQ}V_{YQ}U_{YQ}$ and $Q = U_{YQ}V_{YQ}U_{YQ}$, and the period of this structure is $U_{YQ} + V_{YQ} = Y - Q = \frac{1}{2}(\sqrt{2} - 1)Z$. Since $X = \frac{1}{2}Y$, the structure in Figure 17f is analogous to the structures in Figure 7b, so that $Y = U_{YX}U_{YX} = XX$, and the period of this structure is $X = \frac{1}{2}Y$. Hence the periods of the two structures are $\frac{1}{2}Y$ and $(\sqrt{2} - 1)Y$, which are incommensurable, so that Y is ‘constant’, and its period is infinitesimally small. By extension, since a finite period cannot be found for Y that will accommodate the embedding of X and Q , a finite period cannot be found for Z that will accommodate the embedding of Y, Z, P and Q .

Figure 17. Pairwise combinations of part structures identified in Figure 16 (subscripts omitted in h)

As further illustration, this result can also be shown from consideration of the combinatorial structure of Y , with X and Q as embedded parts. It is required that $Y = XX = U_{YQ}V_{YQ}U_{YQ}V_{YQ}U_{YQ}$, and this is illustrated visually in Figure 18a where X , U_{YQ} and V_{YQ} are represented by rectangles. The structure XX bisects the line Y into two equal parts, and so the structure that results from embedding Q in X is also bisected by separating U_{YQ} into front and back halves, giving $U_{YQ} = U_f U_b$, as illustrated in Figure 18b.

Figure 18. Combinatorial structure of Y , with lines represented by rectangles

This gives $X = U_f U_b V_{YQ} U_f = U_b V_{YQ} U_f U_b$, where U_f and U_b are the same length. Equating the prefixes and suffixes gives $U_f = U_b = U'$, so that

$$U' V_{YQ} = V_{YQ} U'$$

Here, two structures are equated which share the same parts, and applying Lemma 2, the structures can be decomposed into a finer structure, to give

$$U' = A_1 B_1$$

$$V_{YQ} = (A_1 B_1)^i A_1$$

$$U' = B_1 A_1$$

for some integer i and line segments A_1 and B_1 . But now $U' = A_1 B_1 = B_1 A_1$, and applying Lemma 2 again, A_1 and B_1 are decomposed into a finer structure, to give

$$A_1 = A_2 B_2$$

$$B_1 = (A_2 B_2)^j A_2$$

$$A_1 = B_2 A_2$$

for some integer j and line segments A_2 and B_2 . But now $A_2 B_2 = A_1 = B_2 A_2$, so that Lemma 2 can be applied again. The same structure is derived over and over, as U' and V_{YQ} are decomposed into finer and finer parts in a process that tends towards infinitesimals. The process ends only when $A_\omega = \epsilon$ or $B_\omega = \epsilon$, for some integer ω . But, from Equations 2 and 3, this cannot occur since the lengths of V_{YQ} and U' are incommensurate. This means that there is no part structure for Y which accommodates the embedding of X in Q , and consequently, the shape computation cannot be fully implemented if a fixed structure is assumed.

5. Discussion

This paper has presented an analysis of examples of simple shape grammars. Each grammar includes an initial shape and a single shape rule, and the paper has focussed on identifying fixed part structures necessary to accommodate repeated application of the rule. Reducing shape representations to a set of fixed parts is a common strategy which can potentially simplify the difficulties inherent in implementing shape grammars [5, 17]. However, this paper has shown that fixing the parts of shapes can make it impossible to fully implement shape computations, which

includes recognition of emergent shapes. Each example grammar considered produced finite shape computations, including a small number of distinct shapes. Despite this, it was shown that for the majority of the grammars there is no fixed structure of shape decompositions that will accommodate all applications of the shape rule.

5.1 Combining grammars

The different shape grammars considered in the paper are individually limited in terms of their generative capabilities, but they can be combined into a single grammar, as illustrated in Figure 19. This grammar includes a square as an initial shape (Figure 19a) and three shape rules (Figure 19b). R1 formalises the overlapping relation between squares that is apparent in the initial shapes in Figure 1b and Figure 11b; R2 formalises the changing spatial relation between squares, as explored in Section 4.1; and R3 rule is the same rule used in all the grammar examples considered in the paper and rotates a square by 45° about its centre point.

Figure 19. A generative shape grammar, reproduced from [1]

When combined in a single grammar, the generative potential is increased infinitely, as identified in [1] and [8], and it is not possible to represent the resulting shape computation as a finite network of shapes. Some examples of generated shapes are illustrated in Figure 19c, reproduced from [1]. As embedded squares are recognised and transformed they combine with other parts of the shapes to produce a wide range of, sometimes unexpected, forms. The parts of the shapes are not fixed, but are identified and transformed as rules are applied. The shapes in Figure 19c are indicative of the forms that can arise when shape grammars are used as a generative mechanism for creative design. However, if a fixed structure of shapes is assumed then the number of shapes that can be generated is drastically reduced, as shown in Section 4. In Section 4.1, a limit was identified on the spatial arrangement between squares, while in Section 4.2, a limit was identified on the relative scale of squares. In combination, these two limits restrict the rule applications that can be accommodated, and the designs that can be generated.

5.2 Finite Decompositions

The result that a finite decomposition can limit the designs that can be generated by a grammar is arguably not new. Indeed, the simple shape grammars considered in the paper have previously received similar analysis, and the results derived have previously been identified as conjecture but not proved, [1]. Also, research into shape grammar implementation often recognises that relying on pre-defined parts can limit the usefulness of a shape grammar [5]. However, the argument often used to justify this conclusion recognises that in order to specify predefined parts it is first necessary

to make assumptions about the future shape explorations a designer may follow, and therefore restrict the designer's creative freedom. For CACD systems intended to support and enhance a designers' creativity such an outcome is not desirable. While this is a valid practical argument, the results presented in this paper provide a more fundamental reason why pre-defined parts limit the usefulness of a shape grammar implementation. All the shape computations considered in Section 4 are finite, and all shapes that can be generated from application of the shape rule can be enumerated, so that all possible shape explorations a designer may want to follow can be identified. Despite this, it is still not possible to identify part structures that will accommodate all the rule applications in the computation, and this is because of the nature of discrete structures. The different structures necessary to accommodate different rule applications are incommensurate and therefore cannot be combined without decomposing the parts without limit and reducing them to (equal) infinitesimals of Newtonian calculus.

The roots of this argument can be traced back to ancient Greece, and the discovery of irrational measures [28]. The recognition that some numbers are incommensurable with each other destroyed the Pythagoreans' belief that everything could be expressed according to integers, or ratios of integers. This was perhaps the first recognition of an unavoidable tension between two world views: the continuous and the discrete. Research into shape grammar implementation unavoidably straddles the chasm between these. When implementing shape grammars, it is inevitable that shapes are represented symbolically as sets of parts, but as this paper has reaffirmed, shapes are continuous and are inherently different from symbols. Using words to formalise the structure of shapes is a useful device for resolving combinatorial structures, but words and shapes differ fundamentally. The former have an explicit and well defined composition of vocabulary elements. The latter lack an explicit composition. Shapes are not without compositions, but these require construction; they are a consequence of rule application, and not a prerequisite. In practice, this issue has been resolved by recognising that the structure of shapes cannot be fixed. The parts of a shape change dynamically throughout a shape computation and in shape grammar implementation this is achieved by applying algorithms that recognise and transform embedded parts [18]. Using such methods, implementation is relatively straightforward for simple grammars such as those explored in this paper, since they are non-parametric [8]. But, implementing parametric shape grammars remains a challenge, and this paper has served to highlight that fixing the structure of shapes to address this challenge will limit their application.

5.3 Design descriptions

Since the 1960s the enterprise of CAD, e.g. [13, 29], has presented a widening repertoire of design descriptions, extending across lifecycle in Product Lifecycle Management (PLM) systems [30] and to the analysis and simulation of design in computer-aided engineering (CAE) systems [31]. These disparate descriptions have different features that depend on designers' decisions to decompose a design into parts. Parts may be geometric elements, assembly components, functional subsystems or the cell-based descriptions of finite-element analysis (FEA). They are identified for the computation ahead of time, and the computation manipulates just those parts, transforming descriptions via the activity of design. Maintaining consistency among descriptions is much prized, especially in PLM systems. However, designers and engineers may choose whichever parts they fancy as their attention is drawn to see new parts to manipulate and transform. These may be genuinely new parts, not in the original descriptions.

CAD systems do not in principle prejudice which parts can be defined by designers and engineers. An oft-quoted limitation that they can constrain users' creativity does not seem to be a limitation per se. Rather it can be cumbersome and expensive for a user or designer to select their own parts, and

especially difficult for PLM and CAD system developers to maintain consistency across multiple descriptions as those descriptions are transformed in design operations, since the transformations create new parts. If the new parts can be retrofitted to parts of the original then a consistent (topological) structure is maintained through the design transformation [16]. Such retrospective changes may appear counterintuitive when design is set out as a measured move towards an intended goal. But when design involves a speculative leap, then making the leap is followed by a post facto explanation, descriptions are adjusted and what was speculative appears measured and intentional in retrospect.

The straightforward geometrical shape descriptions, with parts as embedded shapes, analysed in this paper expose a problem that new parts created by a transformation in a design activity may decompose the shape into infinitesimal parts for which there is no way to make provision before the transformation. It might be argued that this problem is a consequence of the formal representation of the transformations in design activities as shape production rules [1]. Using rules to represent transformation is a core concept in computation but those rules generally apply to strings and symbols which form natural atomic parts and an associated hierarchy of aggregations. Inherently shapes may lack atoms, although they can easily be constructed for the purposes of computation. The underlying lack of atomic structure seems important to encompass the creativity of design. But this comes at a cost – the problem of infinitesimal parts – which dissolves the shape although there appears to be a residual structure in the repeating patterns of division presented here. Further work will examine this residual structure.

In the early CAD programs, for example Ivan Sutherland's SketchPad [29] sets of parts underlie shape descriptions including line segments and closed polygons. In a sense these 'hidden' shape elements are a characteristic feature of CAD, although always moderated by the ability, in principle, to choose any parts as design proceeds. Perhaps it is instructive to take a shape perspective on designing: choosing/constructing parts, transforming those parts, fusing them back together while revealing new parts. There is a priority for the visual if parts are constructed anew rather than selected from predefined ones. This paper provides some evidence to support this view. A rule (as formalisation of a design action) can create a myriad of parts but it is not always feasible to isolate these beforehand as demonstrated by the examples of embedded lines analysed in this paper. This is a generic issue rather than a pathological case, although, in practice, many rules for design work perfectly well on a defined set of parts with no surprise.

References

- [1] Stiny G, Shape: Talking about Seeing and Doing. MIT Press, 2006
- [2] Trescak T, Esteva M, Rodriguez I, A shape grammar interpreter for rectilinear forms. *Computer-Aided Design* 2012; 44(7): 657-670
- [3] Jowers I, Prats M, McKay A, Garner S, Evaluating an eye tracking interface for a two-dimensional sketch editor. *Computer-Aided Design* 2013; 45(5): 923-936
- [4] Ruiz-Montiel M, Belmonte M-V, Boned J, Mandow L, Millán E, Badillo AR, Pérez-de-la-Cruz J-L, Layered shape grammars. *Computer-Aided Design* 2014; 56: 104-119
- [5] Eloy S, Pauwels P, Economou A, AI EDAM special issue: advances in implemented shape grammars: solutions and applications. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing* 2018; 32(2): 131-137
- [6] Krstic D, From shape computations to shape decompositions. In: Gero JS (ed) *Design Computing and Cognition '16*. Springer, 2016, p. 249-266
- [7] Wortmann T, Stouffs R, Algorithmic complexity of shape grammar implementation. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing* 2018, 32(2), 138-146.

- [8] Yue K, Krishnamurti R, A Paradigm for Interpreting Tractable Shape Grammars. *Environment and Planning B: Planning and Design* 2014; 41(1): 110-137
- [9] Goel V, *Sketches of Thought*. MIT Press, 1995
- [10] Prats M, Lim S, Jowers I, Garner S, Chase S, Transforming shape in design: Observations from studies of sketching. *Design Studies* 2009; 30(5): 503-520
- [11] Krish S, A practical generative design method. *Computer-Aided Design* 2011; 43(1): 88–100
- [12] Lee S, Yan J, The impact of 3D CAD interfaces on user ideation: A comparative analysis using SketchUp and Silhouette Modeler. *Design Studies* 2016; 44: 52-73
- [13] Requicha AAG, Representations of rigid solid objects. In: Encarnacao J, editor. *Computer Aided Design Modelling, Systems Engineering, CAD-Systems. Lecture Notes in Computer Science*, vol 89. Springer, 1980
- [14] Leonard HS, Goodman N, The Calculus of Individuals and Its Uses. *The Journal of Symbolic Logic* 1940; 5(2): 45-55
- [15] Stiny G, Shapes are Individuals. *Environment and Planning B: Planning and Design* (1982) ; 9(3): 359-367
- [16] Stiny G, Shape rules: closure, continuity, and emergence. *Environment and Planning B: Planning and Design* 1994; 21(7): 49-78
- [17] Knight TW, Shape grammars: six types. *Environment and Planning B: Planning and Design* 1999; 26(1): 15-31
- [18] Krishnamurti R, The Construction of Shapes. *Environment and Planning B: Planning and Design* 1981; 8(1): 5-40
- [19] Grasl T, Economou A, From topologies to shapes: parametric shape grammars implemented by graphs. *Environment and Planning B: Planning and Design* 2013; 40(5): 905-922
- [20] Tapia M, A Visual Implementation of a shape grammar system. *Environment and Planning B: Planning and Design* 1999; 26(1): 59-73
- [21] Stiny G, A New Line on Drafting Systems. *Design Computing* 1986; 1: 5-19
- [22] Stiny G, Computational foundations of architectural design. *Environment and Planning B: Planning and Design* 1989; 13(2): 127-132
- [23] Jowers I, Earl C, Stiny G, Shape Computations Without Compositions. In: Çağdaş G, Özkar M, Gül LF, Güler E, editors. *Computer-Aided Architectural Design. Future Trajectories. 17th International Conference, CAAD Futures 2017, Istanbul, Turkey, July 12-14, 2017, Selected Papers*. Springer, 2017, p. 348-365
- [24] Lothaire M, *Combinatorics on Words (Second Edition)*. Cambridge University Press, 1997
- [25] Ruiz-Montiel M, Pérez-de-la-Cruz J L, Mandow L, López-Romero F, Fuzzy shape grammars. *Progress in Artificial Intelligence* 2016; 5(1): 47-53
- [26] Lyndon RC, Schützenberger MP, The equation $a^M = b^N c^P$ in a free group. *Michigan Mathematical Journal* 1962; 9: 289-298
- [27] Fine NJ, Wilfe HS, Uniqueness Theorems for Periodic Functions. *Proceedings of the American Mathematical Society* 1965; 16(1): 109-114
- [28] Euclid, *The Elements, Book X*. Translated by Heath TL. Dover, 1956
- [29] Sutherland IE, SketchPad: A man-machine graphical communication system. In: *AFIPS Conference Proceedings* 23 1963, 323-328
- [30] Stark J, *Product Lifecycle Management: 21st Century Paradigm for Product Realisation*. Springer-Verlag, 2006
- [31] Raphael B, Smith IFC, *Engineering Informatics: Fundamentals of Computer Aided Engineering*. John Wiley & Sons, 2013
- [32] Stiny G. *Pictorial and Formal Aspects of Shapes and Shape Grammars*. Birkhauser, 1975