A Syntactical and Grammatical Approach to Architectural Configuration, Analysis and Generation

Ju Hyun Lee, Michael J. Ostwald and Ning Gu

School of Architecture and Built Environment, The University of Newcastle, Newcastle, Australia

Abstract.

The combination of Space Syntax and Shape Grammar approaches to design potentially offers a rigorous way of first understanding a characteristic instance of architectural configuration (say, a set of designs by an architect, or a particular building type) and then producing variations of that instance. This paper presents a method which commences with a JPG-grammar-based analysis of an architectural design, and then uses a grammatical interpretation of the structure of this syntax. Using this graph as a starting point, it is then possible to develop a 3D shape grammar of the same building. In this way, both syntactical and formal information are captured as part of the sequential analysis. This combined analytical approach can support a better understanding of an architectural style and its design instances as well as the generative capability of these two computational theories – Space Syntax and Shape Grammar. The paper introduces this combined method and then demonstrates it using ten house designs by Glenn Murcutt.

Keywords:

Space Syntax; Shape Grammar; Justified Plan Graph (JPG); Architectural Configuration; Three Dimensional (3D) Shape; Glenn Murcutt

Introduction

The first technologically-enabled, computational approaches to architecture are conventionally traced to the 1960s and 1970s (Alexander 1964; March and Steadman 1971; Steadman 1983). One of the most famous of these approaches, Shape Grammar, views architecture primarily as a type of formal language, examining the logical relationships between elements in the two or three-dimensional shape of a building (Stiny and Gips 1972). By developing a series of rules, often derived from typological considerations, this particular theory can be used to describe a design style and then generate new designs that conform to the principles of this style. The other famous computational approach from this same era, Space Syntax, is more concerned with spatial topologies and social relations, regardless of their formal expressions (Hillier and Hanson 1984). Its purpose is largely analytical, and its conventional application is to question the epistemological properties of a design. Both of these computational approaches have a variety of techniques and offer differing perspectives on, respectively, form and space in design. However, despite their apparent complementarity, they have rarely been successfully combined to consider both the formal and spatial features of an architectural style or type. The present paper describes and demonstrates a new way of selectively combining aspects of the two approaches to allow for the analysis of design instances from an architectural style. In this sense, using the linguistic analogies that were common in the 1970s, this paper demonstrates a potential, syntactically-derived and grammatically-interpolated strategy for architectural configuration and analysis.

The idea of combining the strengths of both Space Syntax and Shape Grammar techniques is, in itself, not new. Several researchers have previously demonstrated that it is possible to develop a Shape Grammar for an architectural style and then, after defining all of the potential form-generating rules, use a Space Syntax technique to decide which combinations of these are most important (March 2002; Heitor et al. 2004; Eloy 2012). Those approaches start with a grammatical, form-based process and conclude with an analysis using graph theory of the way the form-based rules used in this process are applied. There are benefits to such an approach, but they are essentially an application of graph theory to understanding a set of rules and they do not aim to capture any of the important syntactical properties of a design which are amongst the most valuable products of Space Syntax. Therefore, instead of continuing to develop such a model, the present paper offers a new approach, which does the reverse. Firstly, it defines the spatial or syntactical properties of a design using a special type of Justified Plan Graph (called a "JPG grammar") and then it derives a set of rules to generate its corresponding three-dimensional form (a "3D shape grammar"). This sequence effectively starts with a consideration of functional issues and then derives forms to contain or house them. In this way the new method mirrors a common assumption in design theory that, within practical limits, form follows function or that, in a developed design proposal, the arrangement of major functional spaces precedes, or at least moderates, decisions about form.

The new method described in the present paper commences by translating the spatial qualities present in a set of architectural works into three different components. The first is the functional zones in each plan (*node*), the second is their relative adjacency (*link*) and the last, their formal expression (*shape*). Collectively, these three features of a building allow for the spatial and syntactic relations implicit in a design to be defined, before topological rules can be derived from this process for the purpose of determining typical or possible design variations. This approach is ideally intended, like many other applications of syntactical and grammatical research, to be applied to the analysis of characteristic sets of buildings. Be they historic French villas, Prairie Houses by Frank Lloyd Wright or modern offices, patterns in the way a set of works are spatially and formally configured present a fertile subject for both design analysis and generation. The advantage of working with such characteristic sets is both practical and statistical, because understanding and extrapolating any system to produce new data is more effective and useful if there is a larger body of information as a starting point.

This paper commences with an overview of the proposed method and its two configurational grammars: the JPG grammar (Authors 2013; 2014) and the 3D shape grammar. Thereafter, for demonstration purposes, the method is applied to analyse and reveal the syntactical and morphological characteristics of ten house designs by Australian architect Glenn Murcutt. The paper concludes with a discussion about methodological issues and opportunities along with potential further applications of the method to support conceptual thinking at the early stage in the architectural design process.

Node, link and shape

The new method starts with a design from which it extracts three different types of spatial and formal information, each of which can be understood as a stage in the overall process (Figure 1). This approach can then be repeated for multiple additional designs and statistics can then be used to analyse the characteristics of a set of works and reveal a dominant design in the set, or generate new designs that conform to these characteristics.

The first type of information extracted from a design, called a *Node*, is a functionally defined group of spaces. In traditional Space Syntax research a node represents either a convex space or a functionally-defined room, but the breadth of potential variations of such types are too extensive to be useful and so the method used in the present paper is drawn from Amorim's concept of dwelling 'sectors' (Amorim 1999). A dwelling sector is a zone made up of functionally or programmatically related rooms. By grouping such rooms, sector-based nodes are capable of designating initial functional requirements. The second type of information extracted from a design is called a *Link.* A link represents a clear connection or direct relationship between two nodes. When the complete set of nodes and their links are identified the functional relationships presented in an architectural plan are expressed as a graph. However, unlike a conventional Space Syntax graph, in this new process a sequence or order of link construction is derived from a combination of the type of nodes present in the set, and a prioritised sequence. In this way a special type of Justified Plan Graph (JPG) is created, which already has the basis for a grammatical sequence of actions embedded in its formation, thus the new title, a JPG grammar.

The final type of information required in the method is related to *Shape*. This last type, which corresponds to the last stage in the process, provides the basis for several shape-based configurational processes including the development of a 3D Shape Grammar. The third stage is broadly reminiscent of those processes described in past research in the field (Koning and Eizenberg 1981) although its rule basis is determined by the functional structure of the space (expressed in the JPG and its grammar). It is important to note that both the JPG grammar and the 3D shape grammar are not developed strictly in accordance with the standard Shape Grammar approach. Shape Grammar and its variations (including Color, Functional, Parallel, Parametric and Set Grammars) are production systems that can specify a corpus of designs (a design language) by identifying the transformations needed to generate the designs (Stiny and Gips 1980). In contrast, the JPG grammar and the 3D shape grammar are computational algorithms that selectively adopt certain characteristics from both Space Syntax and Shape Grammar theories. This new combined method initially privileges the functional properties of the design (through its syntactical *Node* and *Link* structure) and thereafter derives corresponding forms in the design (through formal and graphical means of *Shape*) that address these issues, and which have not been achieved in previous models that have combined these two theories.

In combination, across its three stages, the method proposed in the present paper allows syntactic knowledge and topological constraints to inform the computational analysis and generation processes. The syntactic knowledge and constraints are potentially distinct factors in an architectural style or a building type. When combined with the associated formal aspects of the style or type, the method is capable of both analysis and generation. This combination of features is unique to the 3D shape grammar presented in this paper in comparison to the other 3D shape studies undertaken in the past (Koning and Eizenberg 1981; Cui and Tang 2014). The processes of constructing a JPG grammar and a 3D shape grammar are described in more detail in the following two sections.

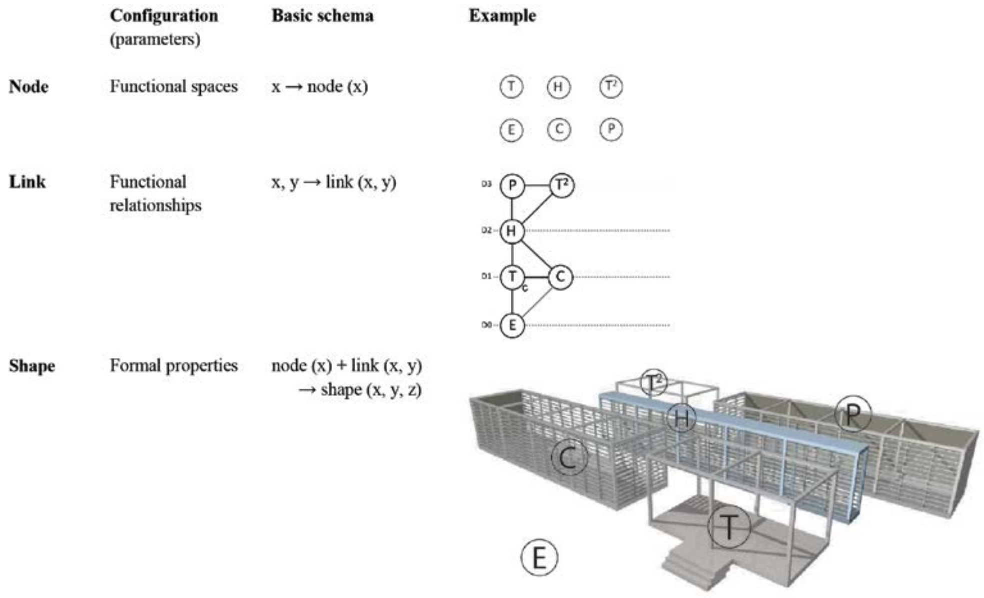


Figure 1. Architectural configuration and schema of node, link, and shape.

Node Key: Common (C), Private (P), Hall (H), Transit (T), Exterior (E)

Constructing the JPG grammar

A JPG consists of nodes and topological links that are formed by two basic schemas, $x \rightarrow$ node (x) and x, $y \rightarrow$ link (x, y) (Figure 1). The formation of the JPG can be defined using traditional graph grammars ((Freudenstein and Maki, 1979; Brandenburg 1995; Grzegorz 1997; Rekers and SchÜRr 1997; Schmidt and Cagan 1997; Li and Schmidt 2004; Kong, Zhang, and Zeng 2006) which consist of sets of non-terminal and terminal symbols. The structural and functional relations are represented by vertices and edges, which are nodes and links in this paper. A link is represented as (i, j) when i and j are the two nodes of the graph. In this paper each node and each link between two nodes are designated as node (x) and link (x, y) respectively.

A JPG can be given a grammatical structure by applying a simple logic to its sequence of construction. Such logic structures are common in Shape Grammar research where many researchers (Cagdas 1996; Hanson and Radford 1986a; Stiny and Mitchell 1978) have selectively adopted structures based on presumed steps in the design process. For example, Stiny and Mitchell (1978) use eight steps for several of their projects, whilst Hanson and Radford (1986a) use 12 steps to generate a Murcutt-style house. These steps, while largely following a reductive logic (that is, they move from a consideration of large scale issues to small ones) are regularly presented as a possible idealised design process for an architect. While all of these examples are for structuring the creation of shape or form, the same logic can also be applied to the construction of the functional zoning relationships in an architectural plan. That is, following a similar logic, it is possible to determine the order in which links are added to a plan graph. For this purpose, the new method requires seven steps, each of which are described hereafter (Figure 2).

Step 1: identify and annotate the functional nodes (spaces with similar programmatic groupings) in the plan and designate the exterior node as the carrier. This step defines the topological size of the JPG (the number of nodes) and the different types of nodes in the graph. The primary functional zonings suggested for this stage, although others are possible, are: exterior (E), hall (H), common area (C), private areas (P), transit (T), and garage or service areas (G). Multiple additional zones of the same type are numbered to differentiate them (for example, two private sectors which are not directly connected would be designated P^1 and P^2).

Step 2: define a core node. Much like the “core unit” in Koning and Eizenberg’s (1981) research, the grammar distinguishes a core node as an important sector which includes the main entrance so that it directly links to an exterior node. The core node plays a significant role in configuring spatial programs as well as the forms required for the future design stage (*Shape*).

Step 3: add the first set of links starting from a core node to its functionally adjacent nodes at the second depth.

Step 4: identify a second set of links starting from a node to functionally adjacent ones at the third depth. Steps 3 and 4 may be similar, however the division allows for identifying the different topological role of a core node as well as for sequentially forming the tree-like structures of a JPG. With the “check step” (4-1 in Figure 2), the application of the fourth step confirms that all the nodes that are developed at the first step are linked in the JPG.

Step 5: where required, generate a link to configure a sub-entrance into a node or a garage sector node at the first depth.

Step 6: add a link between any two nodes as required to replicate permeability and adjacency in the plan. The fifth and sixth steps collectively support the configuration of the JPG such that it can possess various important syntactic structures including being “tree-like” or “ring-like.

Step 7: terminate the generation process for the JPG grammar.

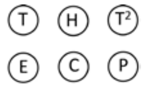
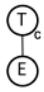
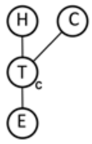
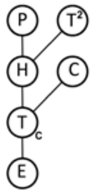
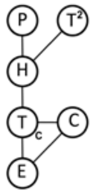
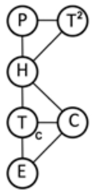
	Step	Rule	Example	
1	Create the required nodes	$x \rightarrow \text{node}(x)$	node(E) node(T) node(H) node(C) node(P) node(T^2)	
2	Define a core node * A core node includes a main entrance so that it directly links to an exterior sector node being regarded as a carrier.	$\text{node}(x) \rightarrow \text{node }(x)^C$ E, $\alpha \rightarrow \text{link(E}, \alpha)$, where α is a core node	node(T)^C link(E,T)	
3	Add a first set of links (starting from the core node to functionally adjacent nodes at the second depth)	$\alpha, x \rightarrow \text{link}(\alpha, x)$, where α is a core node	link(T,H) link(T,C)	
4	Add a second set of links (starting from a node to functionally adjacent nodes at the third depth)	$\beta, x \rightarrow \text{link}(\beta, x)$, where β is a node(s) linking to an adjacent node	link(H,T^2) link(H,P)	
4-1	Check if all nodes are linked	If not, apply $\beta, x \rightarrow \text{link}(\beta, x)$		
5	Add additional links between a node and the exterior node (i.e. inserting a sub-entrance into a node or a garage sector node)	E, $x \rightarrow \text{link(E}, x)$	link(E,C)	
6	Add additional links between two nodes	$x, y \rightarrow \text{link}(x, y)$	link(H,C) link(P,T^2)	
7	Termination			

Figure 2. Sequential steps and rules for the construction of a JPG grammar.

Constructing the 3D shape grammar

The JPG grammar for the spatial function of a plan is based on two basic schemas, '$x \rightarrow$ node (x)' and 'x, $y \rightarrow$ link (x, y)' and is constructed using a sequence of steps, the 3D shape grammar can be represented by the schema 'shape (x, y, z)'. The 3D shape grammar configures formal and graphical features that extend the JPG grammar. There are four shapes or surface related processes considered in this paper (Figure 3): module, block, composition and roof. Once again, additional processes could be used, but the paper adopts these four to analyse the conceptual design processes of a domestic design. Using these four processes, the 3D shape grammar schema is elaborated into the following: ‘shape (Module, Block, {Composition}, Roof)’. For example, ‘shape (C_{4R}, tr, {H, T}, CG)’ indicates a common sector shape consisting of four room-type modules (C_{4R}), with transparent walls (tr), linking to a hall sector ({H,T}), and with a curved gable roof (CG). The four processes are described as follows.

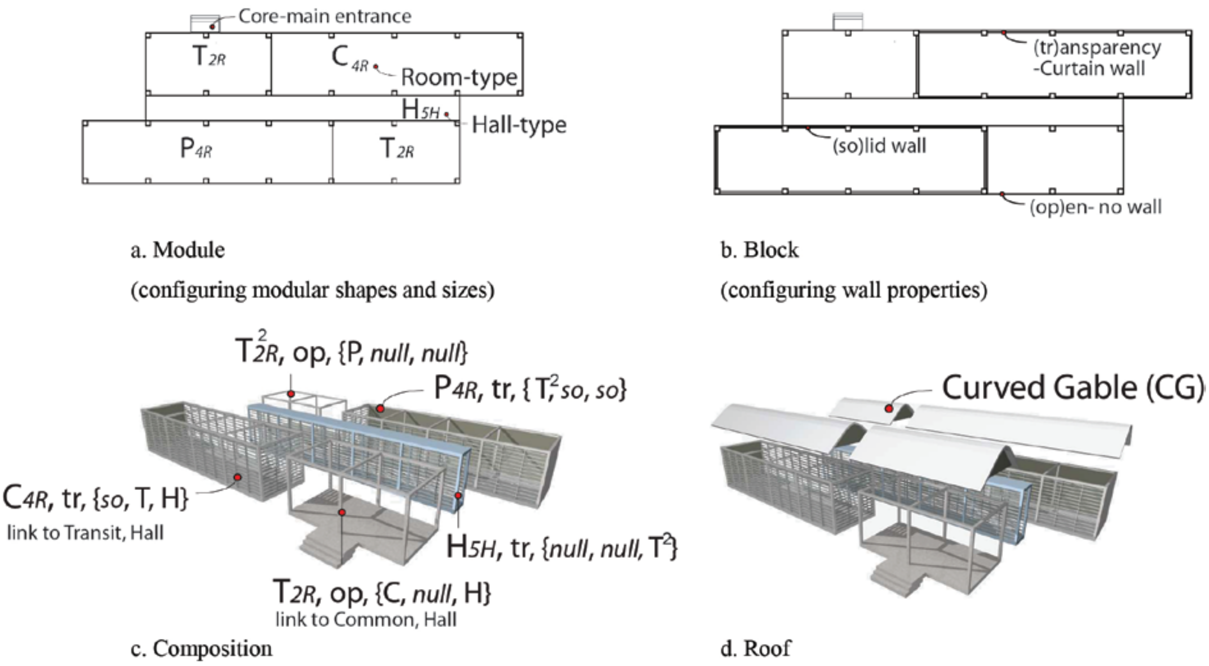


Figure 3. Four processes (a, b, c and d) of the 3D shape grammar for the Marie Short House.

Process 1 (defining modular shapes): the basic plan modules are defined. The conceptual foundation for configuring a 3D grammar typically involves both simplifying the features of the forms used in architecture and decomposing them using a modular system. In past research, Lego, Froebel and Cuisenaire blocks have all provided a useful analogical basis for such modulor systems. Thus, the 3D shape grammar starts by defining the basic modules (structural or functional bays) in a design, which are used to standardise and delineate enclosed spaces. The first shape rule follows the first step of the JPG grammar to transform sector nodes into shapes. Thus, the first rule (shape rule 1.1) is:

$$\text{node}(x) \rightarrow \text{shape}(x), \text{ where } x \text{ is a set of sectors.}$$

The bays can be simplified using a rectangular grid (Figure 3a) which, as demonstrated in the present paper, often conforms to the repetitive column layout or modular construction systems found in many designs. While the dimensions of the enclosed spaces may vary from one building to another, the module, regardless of its shape, is a major characteristic of most designs.

Furthermore, the exact dimensions of modules are not considered in the grammar because the need to simplify the number of variables and dimensions depends on a variety of design contexts. In this part of the process, the sectored modules are elaborated to have an appropriate form and size (modular shapes). For example, Glenn Murcutt's Marie Short House consists of two types of modules: a wide room-type (represented by 'R') which features habitable spaces and a narrow hall-type module (represented by 'H') for circulation (Figure 3). The common sector of this house consists of four room-type modules (4R), while the hall is composed of five hall-type modules (5H). The modular shapes are generated by the combination of two basic modules (room-type and hall-type modules). There are four possible combinations of room-type and hall-type modules. 'R' refers to using only a room-type module, while 'H' a hall-type module. 'RH' is the combination of a room-type and a hall-type module. 'RHH' is the combination of a room-type and two hall-type modules. Since the dimensions of modules are not considered in the grammar, the combinations, 'RH' and 'RHH', can be regarded as a similar modular shape, but the distinction is necessary for generating the different shape composition. 'RH' signifies a combined shape linking two different roof shapes (of one room and one hall), while 'RHH' generates a combined shape linking three different roof shapes (of one room and two halls). Thus, shape rule 1.2 is specified as

$$\text{shape}\ (x) \rightarrow \text{shape}\ (x_{\,ayb}),\ \text{where}\ y = \{\text{R, H, RH, RHH}\},$$

In this shape rule, *a* and *b* together define the size of the 3D shape – *a* (length) indicates how many basic modules the modular shape consists of, while *b* (width) shows the second dimension of the modular shape. For example, 'shape ($H_{3R2/3}$)' is a hall sector whose length equals the length of three basic modules and whose width is two thirds of the width of a basic module. To simplify the notion it can be assumed that a modular shape has the same width as its length, so that the notation system can omit the width (b). Such simplification is often practical and necessary during design conceptualisation and abstraction.

In order to identify the core shape that visualises the core node, the 3D shape grammar adds a set of steps to the main entrance indicating that it is a core shape (see Figure 4 b). Although the locations of the main entrance can vary and it may also have an entrance door, these details are not considered in the current development of the grammar. Thus, shape rule 1.3 configures a core shape distinguished by a main entrance:

$$\text{shape}\ (x_{\,ayb}) \rightarrow \text{shape}\ (x_{\,ayb})^{\,C},\ \text{where}\ x\ \text{is a core node.}$$

Process 2 (defining block shapes): the grammar configures the overall block shapes of the modules by adding walls and specifying their properties. Thus, it is not only the shape of the plan which determines how their forms will look and be used, but also the degree to which the boundaries of each block are transparent, opaque, or an intermediate mixture of these two. For example, in Murcutt's architecture transit modules have no wall and are usually roofed and semi-enclosed.

Thus, they act as intermediate zones between interior and exterior spaces. Common modules often have a transparent wall to the exterior, while private modules have a higher proportion of solid walls to the exterior (Figure 3b). All of these factors are encoded into the design grammar.

Block shapes are typically rectangular prisms having four faces for walls. The bottom face of the block is solid, like a slab-on-ground, while the top face is open. The four wall-faces may have different types, but those defined at this second stage have the same types. This is because the following stage (shape composition) will finalise each wall property according to the three growth directions (Figure 4b) for shape composition. Thus, shape rule 2.1 defines block shapes:

shape (x) → shape (x, Block), where Block = {op, tr, se, so}.

There are four types of blocks: open (op), transparency (tr), semi-transparency (se) and solid (so). Open and transparency blocks are often adopted in Murcutt's architecture, which famously accommodates uninterrupted visual connections to the surrounding landscape. A transparency block tends to employ a curtain wall (glass wall) with internal and/or external louvers. We may adopt sliding walls to provide an adjustable building skin, but the 3D shape grammar is only concerned with base wall types not transient ones. 'Solid' can include small windows, while semi-transparency occurs when over half of a wall is openable, but not including bay windows.

Process 3 (shape composition): the block shapes are connected to generate a 3D form of the design. The logic of shape composition is based on the structure of the JPG grammar which defines the first set of links starting from a core node/shape to functionally adjacent nodes/shapes. The core shape, as the first 3D shape defined in the 3D shape grammar, provides three possible growth directions (α, β and γ in Figure 4b) for composing the overall 3D form by linking to adjacent 3D shapes. The first process and this third process have a direct relationship to the JPG grammar because the first process relates to the nodes and the third process deals with the links in the JPG grammar. Although the detailed composition varies in ways that correspond to a variety of building designs or architectural styles, the basic schema of shape composition can be defined as:

shape (x, Block) → shape (x, Block, {α, β, γ}).

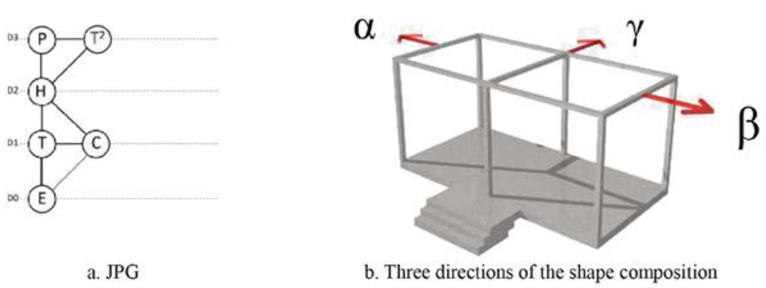


Figure 4. An example JPG (a) and its corresponding 3D form (b) specifying three growth directions for shape composition.

Based on this schema, the shape composition can then be developed using four shape rules corresponding to the generation steps specified in Table 1. Shape rule 3.1 creates the markers (→) of three growth directions on the three faces of the core shape. Shape rule 3.2 attaches block shapes to the core shape (or other block shapes) and the added block shapes also create the markers except for the face joining the other shape. Shape composition to the first two growth directions (α and β) of a block shape is uniquely defined when added shapes is joined. However, shape composition to the third growth direction (γ) can have two options (left or right). After all the shape compositions are defined, shape rule 3.3 finalises all block shapes by configuring individual walls. If there is no further configuration required, the final step deletes remaining markers by applying shape rule 3.4. In addition, if there is no configuration at this third process, i.e. {*null*, *null*, *null*}, the entire process is skipped and is represented by the symbol for the empty set, {}.

Table 1. Four shape rules (3.1 – 3.4) for shape composition.

Rule	Generation Step	Notation
3.1	Create marker	shape (x, Block) → shape (x, Block, {α, β, γ})
3.2	Add block shape	shape (x, Block, {α, β, γ}) → shape (x, Block, {y, β, γ}) + shape (y, Block, {α, x, γ}).
		shape (x, Block, {α, β, γ}) → shape (x, Block, {α, y, γ}) + shape (y, Block, {x, β, γ}).
		shape (x, Block, {α, β, γ}) → shape (x, Block, {α, β, y}) + shape (y, Block, {α, β, γ})
		; option, where option = {left, right}.
3.3	Define wall	shape (x, Block, {α, β, γ}) → shape (x, Block, {Wall, β, γ}).
		shape (x, Block, {α, β, γ}) → shape (x, Block, {α, Wall, γ}).
		shape (x, Block, {α, β, γ}) → shape (x, Block, {α, β, Wall}).
		,where Wall = {op, tr, se, so}
3.4	Delete marker	shape (x, Block, {α, β, γ}) → shape (x, Block, {*null*, β, γ}).
		shape (x, Block, {α, β, γ}) → shape (x, Block, {α, *null*, γ}).
		shape (x, Block, {α, β, γ}) → shape (x, Block, {α, β, *null*}).

When overlapping two different wall types in the compositions, the face follows the thicker types from open to solid. For example, merging ‘transparency’ with ‘solid’ results in a ‘solid’ wall. As observed in Murcutt’s domestic buildings, a hall block shape consisting of ‘H’ modular shapes has no shape composition to the first two growth directions (α and β). In addition, the only shape composition to the third growth direction (γ) of the hall block shape deletes its γ marker.

Process 4 (defining roof shapes): the roof shape is configured. There are many possible roof shapes including flat, shed, gable, hipped and butterfly roofs. Although the roof shapes depend on the actual design and its context, this final process provides some common 3D forms of roofs corresponding to the defined block shapes. They can be combined, expanded and transformed for different scenarios. Within Murcutt’s domestic buildings, we grouped the most common roof designs into six types (Table 2). Thus, this shape rule configures roof shapes based on these six types:

$$\text{shape } (x, \text{Block}, \{\text{Composition}\}) \rightarrow \text{shape } (x, \text{Block}, \{\text{Composition}\}, \text{Roof})$$

$$\text{, where Roof} = \{\text{SC, SL, FC, FL, GC, GL}\}.$$

Table 2. Common roof types in Murcutt's domestic buildings.

Shed		Gable	
Curved (SC)	Linear (SL)	Curved (GC)	Linear (GL)
			
Flat		Other variation	
Curved (FC)	Linear (FL)		
			

A syntactically-derived grammar for Murcutt's domestic architecture

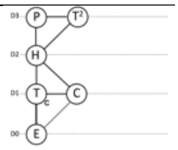
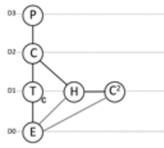
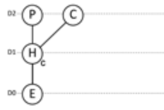
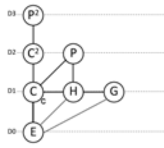
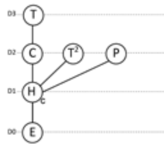
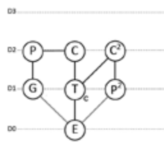
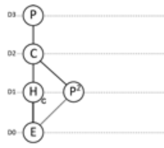
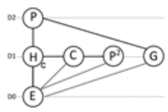
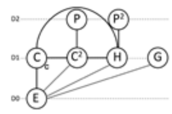
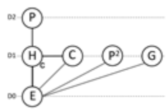
Glenn Murcutt's architecture is characterised by a high degree of consistency and clarity in his approach to creating both space and shape. As such, it has been the subject of both Space Syntax (Ostwald 2011a; 2011b) and Shape Grammar research (Hanson and Radford 1986a; 1986b). In order to establish and demonstrate a syntactic, grammatical approach, the present paper illustrates the method using ten domestic buildings on rural sites, which were constructed between 1975 and 2005.

An application of the JPG grammar and the JPG-grammar-based analysis

The six sectors used as nodes for the ten examples of Murcutt's domestic designs are exterior (E), hall (H), common (C), private (P), transit (T), and garage (G). There are four 'enclosed' or internal sectors; (C, P, H, G). The first of these (C), includes living rooms, dining rooms, foyers and kitchens while the second (P), contains bedrooms and bathrooms. The third (H) includes corridors, hallways and linking spaces and the fourth (G) is focussed on the storage of cars, but also includes workshops, laundries and service areas. In addition to this, there are two types of 'open' sectors, transit (T) and exterior (E). Transit spaces are intermediate zones between interior and exterior, or occasionally between two interior spaces. Transit spaces are usually roofed, and semi-enclosed, but are still open to weather conditions. The exterior is simply the outside world, and it is most closely associated with ingress and egress relationships.

Each functional sector can include multiple small spaces (alcoves, bathrooms, toilets, utility cupboards) within its larger grouping. Thus, a grouping of entry foyer, living room and gallery might be a common space (C), while elsewhere in the same plan a second grouping of dining room, kitchen and music room, might be a second common space (C^2). Table 3 shows the production of each step through the JPG grammar application towards the final JPG outcomes generated for the ten cases.

Table 3. The production of each step through the JPG grammar application and the final JPG outcomes generated for the ten cases.

Case	Step 1	Step 2	Step 3	Step 4	Step 5	Step 6	JPG
1: Marie Short House (1975)	node(E), node(T), node(H), node(C), node(P), node(T^2),	node(T)^C, link(E,T)	link(T,H), link(T,C)	link(H,T^2), link(H,P)	link(E,C)	link(H,C), link(P,T^2)	
2: Nicholas House (1980)	node(E), node(T), node(C), node(P), node(H), node(C^2),	node(T)^C, link(E,T)	link(T,C)	link(C,P), link(C,H), link(H,C^2)	link(E,H), link(E,C^2)	-	
3: Carruthers House (1980)	node(E), node(H), node(P), node(C),	node(H)^C, link(E,H)	link(H,P), link(H,C)	-	-	-	
4: Fredericks house (1982)	node(E), node(C), node(C^2), node(P), node(H), node(P^2), node(G)	node(C)^C, link(E,C)	link(C,C^2), link(C,P)	link(C^2,P^2), link(P,H), link(H,G)	link(E,H), link (E,G)	link(H,C)	
5: Ball-Eastaway House (1982)	node(E), node(H), node(C), node(T), node(T^2), node(P),	node(H)^C, link(E,H)	link(H,C), link(H,T^2), link(H,P)	link(C,T)	-	-	
6: Magney House (1984)	node(E), node(T), node(C), node(P), node(G), node(P^2), node(C^2)	node(T)^C, link(E,T)	link(T,C), link(T, C^2)	link(C,P), link(C^2,P^2), link(P,G)	link(E,P^2), link(E,G)	-	
7: Simpson-Lee House (1994)	node(E), node(H), node(C), node(P), node(P^2)	node(H)^C, link(E,H)	link(H,C)	link(C,P), link(C,P^2)	link(E,P^2)	-	
8: Fletcher-Page House (1998)	node(E), node(H), node(P), node(C), node(P^2), node(G),	node(H)^C, link(E,H)	link(H,P), link(H,C)	link(C,P^2), link(P,G)	link(E,C), link(E,P^2), link(E,G)	-	
9: Southern Highlands (2001)	node(E), node(C), node(C^2), node(P), node(H), node(P^2), node(G)	node(C)^C, link(E,C)	link(C,C^2), link(C,H)	link(C^2,P), link(H,P^2)	link(E,C^2), link(E,H), link(E,G)	link(C^2,H)	
10: Walsh House (2005)	node(E), node(H), node(P), node(C), node(P^2), node(G)	node(H)^C, link(E,H)	link(H,P), link(H,C)	-	link(E,C), link(E,P^2), link(E,G)	-	

The first step in the JPG grammar application configures the topological size of each house. All example cases consist of six sectors, but no case has all of the six functional sectors. Thus, as previously stated, although two adjacent rooms that have a similar function are regarded as a sector, there may be a second or even third sector of the same functional type. There are four common nodes in Murcutt's houses: (E, H, C, P). The topological size of each JPG in the ten cases ranges from four to seven sectors. Five cases consist of six sectors (50%) and three cases, seven sectors. That is, six or seven sectors can be regarded as the dominant topological size among the ten cases. These nodes are E, C, P, H, P^2 and G. Interestingly, the more recent buildings (cases 8 and 10) actually include these dominant nodes.

Next, in order to configure links, the JPG grammar generates a core node link. In the first case a transit sector (T) is the core node, while in the last case it is a hall sector. An examination of the frequency of core nodes indicates that a hall sector (H) dominates at the second step (50%). A transit sector (T) is located as a core in three cases and a common sector in two cases. The third and fourth steps of the JPG grammar application consider the next adjacent functional space after the configured nodes at the previous step. The rules at the third step add a first set of links starting from a core node. Two links are commonly generated in the third step so that they sequentially form a tree-like structure. For example, the pair of links, link (T, H) and link (T, C), and link (H, P) and link (H, C), allow a core node to be adopted as a circulation zone. The particular production recorded in Table 3 can also be related to its syntactic values. A pair of links, link (H, P) and link (H, C), dominates this step. This is because half of the cases at the previous step use a hall as a core node. However, the core nodes in two cases (2, 7) are linked to only a common sector at this step. Thus, the common sector provides the circulation function. Case 5 adopts rules linking to three sectors at this step, so the core node has the highest integration (*i*) and control (*CV*) value.

The fourth step defines a second set of links starting from a node to functionally adjacent nodes. The third and tenth cases skip this step as they are exceptions, while the total of 15 links are generated in the other eight cases. Similar to the third step, many links (10 links) generated here also start from a common sector. The common and hall sectors may be generally used as a circulation zone that naturally links to the other spaces and often forms a chain or loop in the graph.

The fifth step of the grammar application defines a link between a node and the exterior node by configuring a sub-entrance node or a garage sector node. In eight of the selected Murcutt's houses, five sectors (G, P^2, C^2, C, H) are linked to the exterior at this step. These rules are applied to form 17 such links in eight of the ten cases. The last three cases each add three such links at this step. That is, more spaces in the later houses tend to connect directly to the exterior sectors so that the exterior sector would become the most integrated space. After this step, many cases also show particular ring-type (c-d-type) nodes (Hiller 1999). The results also indicate that the common sector and garage sector often link to the exterior sector.

The sixth step of the grammar application adds a link between any two remaining nodes. Only three cases (1, 4, 9) apply the sixth step. The particular links generated are between H, C or C^2. Thus, a hall sector and a common sector are often linked together. The dominant rule at the sixth step however is skipping to the final step (not illustrated in Table 3) to terminate the generation of the JPG.

By investigating the rules applied at each of the above steps, we are able to explore the syntactic structure of each JPG for Murcutt houses. Once the graph of sectors (nodes) and functional adjacency (links) is prepared and justified, then mathematical analysis that determines total depth (*TD*), mean depth (*MD*), Relative Asymmetry (*RA*), integration (*i*) and control (*CV*) (Hillier and Hanson 1984; Osman and Suliman 1994; Hanson 1998; Ostwald 2011c) can be considered as syntactic properties. These mathematical values are also used to assist to identify a genotype (Ostwald 2011b), or to support an understanding of the design topologies present (Heitor et al. 2004). The syntactic analysis results can also be used to direct the rule application of the JPG grammar for the generation of design instances, which best capture a particular design style as measured through these syntactic values.

An application of the 3D shape grammar and the 3D-shape-grammar-based analysis

The 3D shape grammar configures a set of shape-based extensions from the JPG grammar to provide an overall form of the design. This descriptive grammar involves four processes with relevant configurations for defining (a) modular shapes, (b) block shapes, (c) shape composition and (d) roof shapes.

The first of these, *modular shapes,* are configured using two sets of rules. The first set in the 3D shape grammar groups similar programmatic areas into functionally-sectored modules in the plan. This parallels the first step of the JPG grammar application but uses a basic schema, node (x) → shape (x). That is, the sector nodes of the JPG are transformed into shapes. In this way, the 3D shape grammar produces modular shapes and then configures each sectored shape with an appropriate size. Except for two cases (cases 1 and 4 in Table 4), all modular shapes adopt the basic module length, that is, a=1, and the notation system omits the width (b) for simplification. The length of common and private sectors in the ten cases ranges from 1 to 4 modules while the length of hall sectors vary more between 1/2 and 6 modules. The length of the remaining transit and garage sectors range between 1 and 2 modules.

The length is determined by design requirements. For example, two cases (2 and 4) require more common spaces (including a second common sector) and because of this their lengths reach up to 5 modules. Three cases (7, 8 and 10) appear to have more private spaces (over 3.5 modules) than common spaces (under 2.5 modules). In the other three cases (1, 6 and 9) the length of their common sectors is similar to one of their private sectors. Even though the current grammar disregards the exact dimensions of each module to develop more simplified rules, they can still be used as a means to examine and generate various volumes of the 3D modular shapes for the design. In future, they can be augmented to consider detailed dimensions if needed.

The room-type module (‘R’) is a dominant shape in the ten houses. The combination of a room-type and a hall-type module ('RH') is found in the Simpson-Lee House (Case 7 in Table 4). The combination of a room-type and two hall-type modules ('RHH') is seen in the Magney House (Case 6 in Table 4). Both cases 6 and 7 could be regarded as complex (or unusual) compositions in terms of modular shape generation. For all ten cases, altogether there are 39 'R', 5 'H', 6 'RHH' and 1 'RH' modular shapes being generated (see the production of step 1 in Table 4 for details).

In the four cases featuring a pavilion (cases 3, 5, 8 and 10), their architectural forms adopt only 'R' modular shapes, while the Simpson-Lee House (case 7) consists of three types of modular shapes, being 'R', 'H' and 'RH'. By adding a hall-type module (an internal corridor), the 3D shape grammar can form two pavilions in domestic architecture. In cases 1, 2 and 4 the 'R' modular shapes are connected by the 'H' module, while the form of the Southern Highlands (case 9) is expanded by adding a long 'H' modular shape. Figure 5 illustrates the sectored plans and the generation of modular shapes using two cases (1 and 10) as examples.

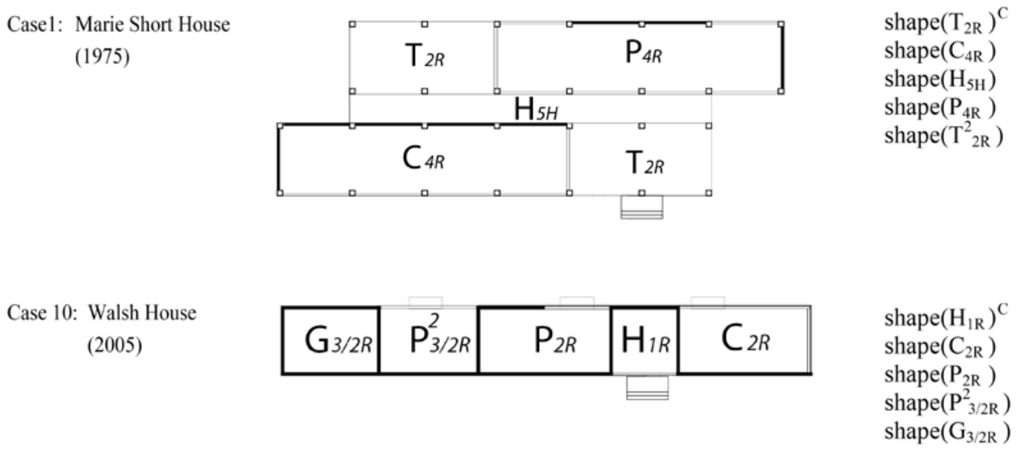


Figure 5. Two examples of the generation of modular shapes.

Block shapes are defined by wall types and the designation of the main entrance (as for a core shape). Figure 6 illustrates the generation of different block shapes using two cases (1 and 10) as examples.

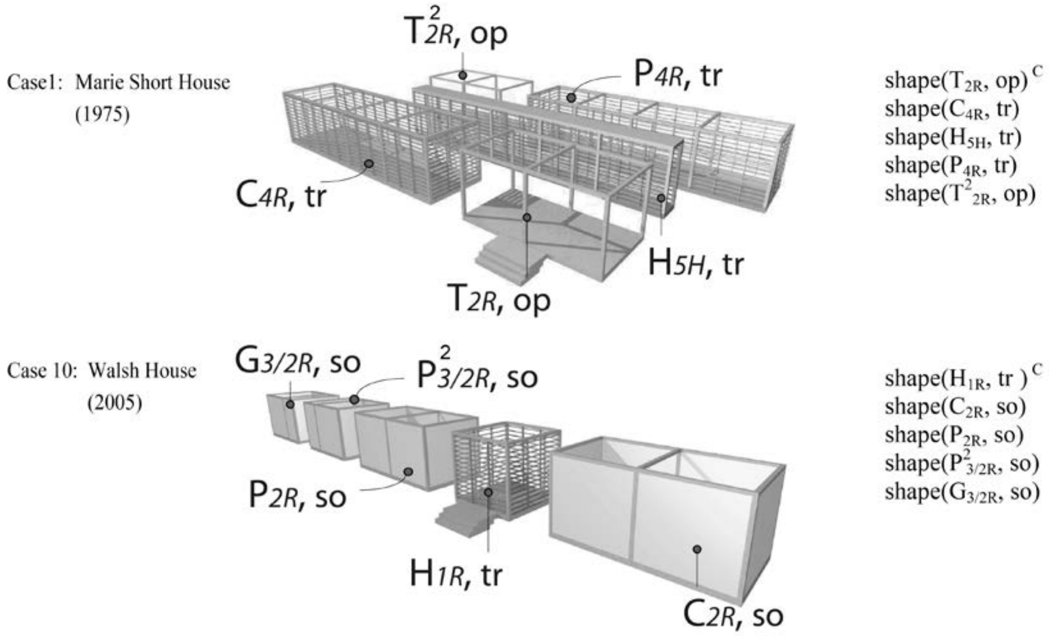


Figure 6. Examples of the generation of block shapes (Open (op), Transparency (tr), Solid (so))

Shape Composition is the third process of the 3D shape grammar application for composing the overall form of the design. A core shape is the centre of the design and is used as the starting point for composition. The 3D shape grammar uses the JPG grammar as a reference (Table 3) providing links to adjacent shapes. This process also highlights three possible growth directions (α, β and γ) for composing the overall 3D form by joining adjacent shapes. Figure 7 illustrates the shape compositions for the Marie Short House (case 1). The links (semantics) specified between different nodes in the JPGs (Table 3) not only guide shape composition here step by step, but they also allow us to avoid unnecessary or unconventional form generation. The process of shape composition is finalised by examining and selecting from the generated alternatives according to the syntactic structure of the architecture. For example, the shape composition for the Marie Short House (Figure 7) follows the steps from the first depth to the third depth of the JPG and generates a corpus of design alternatives. Due to the final examination and selection, two alternatives (each marked with a cross symbol in Figure 7) are removed from the corpus because they do not match the links specified in the JPG. In this way, the 3D shape grammar is not only able to generate the original Marie Short House but also alternatives that share the compositional characteristics of the Marie Short House, by altering different growth directions in shape rule 3.2 and different wall configurations in shape rule 3.3.

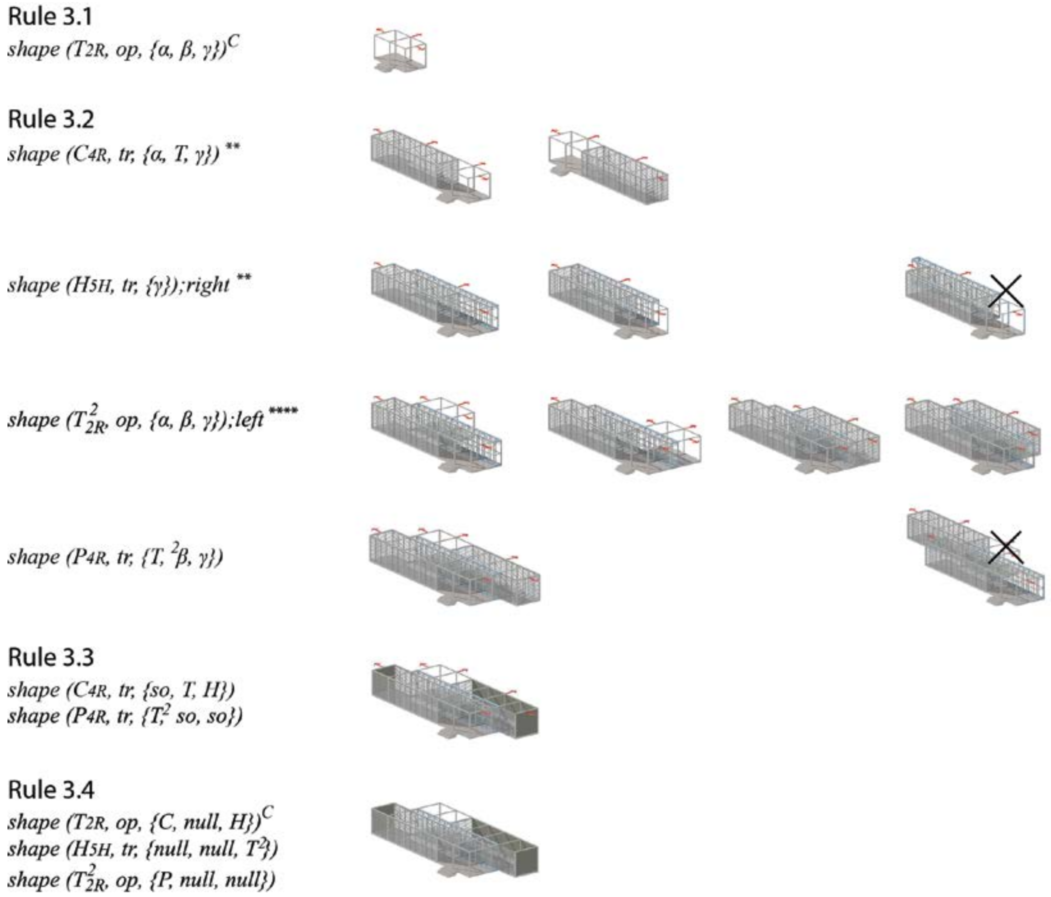


Figure 7. Shape composition of the Marie Short House (** two alternatives generated, **** four alternatives generated)

Table 4. The production of each process through the 3D shape grammar application and the final 3D forms generated for the ten cases.

Case	Process 1	Process 2	Process 3	Process 4	3D Form
1	shape(T_{2R})^{C} shape(C_{4R}) shape(H_{5H}) shape(P_{4R}) shape(T^2_{2R})	shape(T_{2R}, op)^{C} shape(C_{4R}, tr) shape(H_{5H}, tr) shape(P_{4R}, tr) shape(T^2_{2R}, op)	shape (T_{2R}, op, {C, *null*, H})^{C} shape (C_{4R}, tr, {so, T, H}) shape (H_{5H}, tr, {*null*, *null*, T^2}) shape (T^2_{2R}, op, {P, *null*, *null*}) shape (P_{4R}, tr, {T^2, so, so})	shape (T_{2R}, op, {C, *null*, H}, GC)^{C} shape (C_{4R}, tr, {so, T, H}, GC) shape (H_{5H}, tr, {*null*, *null*, T^2}, FL) shape (T^2_{2R}, op, {P, *null*, *null*}, GC) shape (P_{4R}, tr, {T^2, so, so}, GC)	
2	shape(T_{2R}) shape(C_{2R}) shape(P_{1R}) shape(H_{3H}) shape(C^2_{3R})	shape(T_{2R}, op)^{C} shape(C_{2R}, tr) shape(P_{1R} , so) shape(H_{3H}, tr) shape($C^2_{3R/2}$, so)	shape(T_{2R}, op, {*null*, C, *null*})^{C} shape(C_{2R}, tr, {T, P, H}) shape(P_{1R} , so, {C, *null*, H}) shape(H_{3H}, tr, {*null*, *null*, C^2}) shape($C^2_{3R/2}$, so, {})	shape(T_{2R}, op, {*null*, C, *null*}, GL)^{C} shape(C_{2R}, tr, {T, P, H}, GL) shape(P_{1R} , so, {C, *null*, H}, GL) shape(H_{3H}, tr, {*null*, *null*, C^2}, FL) shape($C^2_{3R/2}$, so, {}, GL)	
3	shape($H_{3R/2}$) shape($P_{3R/2}$) shape(C_{4R})	shape($H_{3R/2}$, tr)^{C} shape($P_{3R/2}$, so) shape(C_{4R}, tr)	shape($H_{3R/2}$, tr, {C, *null*, P})^{C} shape($P_{3R/2}$, so, {}) shape(C_{4R}, tr, {*null*, H, so})	shape($H_{3R/2}$, tr, {C, *null*, P}, GL)^{C} shape($P_{3R/2}$, so, {}, GL) shape(C_{4R}, tr {*null*, H, so}, GL)	
4	shape(C_{2R}) shape(P_{2R}) shape(C^2_{3R}) shape(P^2_{2R}) shape(H_{1H}) shape(G_{2R})	shape(C_{2R}, tr)^{C} shape(P_{2R}, tr) shape(C^2_{3R}, tr) shape(P^2_{2R}, tr) shape(H_{1H}, tr) shape(G_{2R}, so)	shape(C_{2R}, tr, {P, C^2, so})^{C} shape(P_{2R}, tr, {so, C, so}) shape(C^2_{3R}, tr, {C, P^2, so}) shape(P^2_{2R}, tr, {C^2, so, so}) shape(H_{1H}, tr, {*null*, *null*, G}) shape(G_{2R}, so, {})	shape(C_{2R}, tr, {P, C^2, so}, GC)^{C} shape(P_{2R}, tr, {so, C, so}, GC) shape(C^2_{3R}, tr, {C, P^2, so}, GC) shape(P^2_{2R}, tr, {C^2, so, so}, GC) shape(H_{1H}, tr, {*null*, *null*, G}, FL) shape(G_{2R}, so, {}, GC)	
5	shape($H_{3R2/3}$) shape(P_{1R}) shape(C_{2R}) shape(T_{1R}) shape($T^2_{2R/3}$)	shape($H_{3R2/3}$, so)^{C} shape(P_{1R}, so) shape(C_{2R}, so) shape(T_{1R}, op) shape($T^2_{2R/3}$, op)	shape($H_{3R2/3}$, so, {P, C, T^2})^{C} shape(P_{1R}, so, {tr, H, *null*}) shape(C_{2R}, so, {H, T, *null*}) shape(T_{1R}, op, {C, *null*, *null*}) shape($T^2_{2R/3}$, op, {P, C, *null*})	shape($H_{3R2/3}$, so, {P, C, T^2}, GC)^{C} shape(P_{1R}, so, {tr, H, *null*}, GC) shape(C_{2R}, so, {H, T, *null*}, GC) shape(T_{1R}, op, {C, *null*, *null*}, GC) shape($T^2_{2R/3}$, op, {P, C, *null*}, GC)	
6	shape(T_{1RHH}) shape(C_{1RHH}) shape(P_{1RHH}) shape(G_{1RHH}) shape(C^2_{1RHH}) shape(P^2_{1RHH})	shape(T_{1RHH}, op)^{C} shape(C_{1RHH}, tr) shape(P_{1RHH}, tr) shape(G_{1RHH}, tr) shape(C^2_{1RHH}, tr) shape(P^2_{1RHH}, tr)	shape(T_{1RHH}, op, {C^2, C, so})^{C} shape(C_{1RHH}, tr, {T, P, so}) shape(P_{1RHH}, tr, {C, G, so}) shape(G_{1RHH}, tr, {P, *null*, so}) shape(C^2_{1RHH}, tr, {P^2, T, so}) shape(P^2_{1RHH}, tr, {*null*, C^2, so})	shape(T_{1RHH}, op, {C^2, C, so}, {SC,FL,SC})^{C} shape(C_{1RHH}, tr, {T, P, so}, {SC,FL,SC}) shape(P_{1RHH}, tr, {C, G, so}, {SC,FL,SC}) shape(G_{1RHH}, tr, {P, *null*, so}, {SC,FL,SC}) shape(C^2_{1RHH}, tr, {P^2, T, so}, {SC,FL,SC}) shape(P^2_{1RHH}, tr, {*null*, C^2, so}, {SC,FL,SC})	
7	shape(H_{1H1}) shape($C_{5/2RH}$) shape(P_{2R}) shape($P^2_{3/2R}$)	shape(H_{1H1}, tr)^{C} shape($C_{5/2RH}$, tr) shape(P_{2R}, so) shape($P^2_{3/2R}$, so)	shape(H_{1H1}, tr, {C, *null*, P})^{C} shape($C_{5/2RH}$, tr, {P^2, H, so}) shape(P_{2R}, so, {C, tr, *null*}) shape($P^2_{3/2R}$, so, {tr, C, *null*})	shape(H_{1H1}, tr, {C, *null*, P}, SL)^{C} shape($C_{5/2RH}$, tr, {P^2, H, so}, SL) shape(P_{2R}, so, {C, tr, *null*}, SL) shape($P^2_{3/2R}$, so, {tr, C, *null*}, SL)	
8	shape($H_{1/2R}$) shape(C_{2R}) shape($P_{5/2R}$) shape(P^2_{2R}) shape($G_{3/2R}$)	shape($H_{1/2R}$, tr)^{C} shape(C_{2R}, se) shape($P_{5/2R}$, so) shape(P^2_{2R}, so) shape($G_{3/2R}$, so)	shape($H_{1/2R}$, tr, {P, C, *null*})^{C} shape(C_{2R}, se, {H, P^2, tr}) shape($P_{5/2R}$, so, {G, H, se}) shape(P^2_{2R}, so, {C, *null*, *null*}) shape($G_{3/2R}$, so, {*null*, P, *null*})	shape($H_{1/2R}$, tr, {P, C, *null*}, SL)^{C} shape(C_{2R}, se, {H, P2, tr}, SL) shape($P_{5/2R}$, so, {G, H, se}, SL) shape(P^2_{2R}, so, {C, *null*, *null*}, SL) shape($G_{3/2R}$, so, {*null*, P, *null*}, SL)	
9	shape($C_{8/3R}$) shape(C^2_{1R}) shape(P_{1R}) shape($P^2_{8/3R}$) shape(G_{1R}) shape(H_{6H})	shape($C_{8/3R}$, se)^{C} shape(C^2_{1R}, tr) shape(P_{1R}, se) shape($P^2_{8/3R}$, tr) shape(G_{1R}, so) shape(H_{6H}, so)	shape($C_{8/3R}$, se, { P^2, C^2, H})^{C} shape(C^2_{1R}, tr, { C, P, H}) shape(P_{1R}, se, {C, *null*, *null*}) shape($P^2_{8/3R}$, tr, {G, C, H}) shape(G_{1R}, so, {*null*, P^2, *null*}) shape(H_{6H}, so, {tr, *null*,tr})	shape($C_{8/3R}$, se, { P^2, C^2, H}, GC)^{C} shape(C^2_{1R}, tr, { C, P, H}, GC) shape(P_{1R}, se, {C, *null*, *null*}, GC) shape($P^2_{8/3R}$, tr, {G, C, H}, GC) shape(G_{1R}, so, {*null*, P^2, *null*}, GC) shape(H_{6H}, so, {tr, *null*,tr}, SC)	
10	shape(H_{1R}) shape(C_{2R}) shape(P_{2R}) shape($P^2_{3/2R}$) shape($G_{3/2R}$)	shape(H_{1R}, tr)^{C} shape(C_{2R}, so) shape(P_{2R}, so) shape($P^2_{3/2R}$, so) shape($G_{3/2R}$, so)	shape(H_{1R}, tr, {P, C, so})^{C} shape(C_{2R}, so, {H, tr, tr}) shape(P_{2R}, so, {P^2, H, se}) shape($P^2_{3/2R}$, so, {G, P, tr}) shape($G_{3/2R}$, so,{*null*, P^2, *null*})	shape(H_{1R}, so, {P, C, so}, SL)^{C} shape(C_{2R}, so, {H, tr, tr}, SL) shape(P_{2R}, so, {P^2, H, se}, SL) shape($P^2_{3/2R}$, so, {G, P, tr}, SL) shape($G_{3/2R}$, so, {*null*, P^2, *null*}, SL)	

Roof shapes in Murcutt's architecture actually vary in response to climatic and contextual issues including sunlight, winds and views (Hanson and Radford 1986b). In addition, the corrugated metal sheets Murcutt uses for roofs allow for a wide variety of shapes. However, our 3D shape grammar focuses on the combination of the possible roof sub-shapes that make up the overall roof shape. Under this constraint, a pavilion employing only a 'R' or 'H' block shape has a roof sub-shape, while a pavilion adopting a 'RHH' block shape such as the Magney House (case 6), is registered by three different roof sub-shapes. The buildings featuring two pavilions (cases 1, 2 and 4) that consist of three block shapes (‘R’, ‘H’ and ‘R’) also have three roof sub-shapes.

The more recent cases (7, 8 and 10) adopt simple linear mono-pitch pavilions and have a ‘SL’ (shed linear) roof shape, while the large and complex houses such as the Southern Highlands (case 9) has three roof sub-shapes of ‘SL’, ‘GC’ (gable curved) combing with another ‘SL’. Twin pavilion buildings (cases 1, 2 and 4) also have three roof sub-shapes consisting of two pitched roofs (‘GL’ or ‘GC’) for ‘R’ block shapes and a flat roof (‘FL’) for ‘H’ block shapes. The orientation of shed roofs in Murcutt’s houses follows a common design strategy with ‘open fronts’ and ‘protective backs’. By following these four processes, the application of the 3D shape grammar generates the final 3D forms of the ten selected Murcutt domestic buildings (Table 4).

Conclusion

This paper describes and demonstrates a new method for recording and extrapolating design instances from an architectural style using both syntactical and grammatical processes. The method uses nodes, links and shapes developed through two grammars: JPG grammar and 3D shape grammar. With the final forms produced by the 3D shape grammar adhering to the syntactic relationships described in the JPG grammar, the method captures both syntactic (space-based) and stylistic (form-based) characteristics of a design. When applied to a set of designs, the combined spatial and formal characteristics can be generalised and then used for generating new alternatives that share these characteristics. Such knowledge will allow for an appropriate abstraction for understanding a design language and for the search for an optimal design (Gero and Coyne 1985) as well as its alternatives. For example, Tables 3 and 4 can provide the frequencies of the rules being applied in each step and process, which can serve as potential guides for analysing the dominant design characteristics of the set and for reproducing these characteristics.

Fundamentally, the JPG grammar is derived by uncovering a topological design structure that allows for the analysis of a possible design process which has been used to develop these functional relationships. The results of the syntactical analysis of the JPG grammar are also useful for determining some of the characteristics of an architectural design and for supporting the evolution of designs by way of the topological relationships generated by the JPG rules. The JPG grammar is significant because it is capable of both rule-based and syntax-based analysis and it could facilitate the exploration of the design instances and inequality genotypes of domestic designs as demonstrated in the selected cases (Ostwald 2011a; 2011b; Authors 2013). Despite this, the JPG-grammar-based framework presented here relies on programmatic adjacencies and so may not provide the same level of insight into syntactic relations as it does for spatial topologies. Furthermore, even though the modular plan facilitates the generation of the basic 3D shapes, the outcome may be over simplified and allow for only a basic massing study to be developed. The use of convex spaces, instead of functional sectors, would allow for this method to be usedf to address larger epistemological questions raised by design, although such a change may make the grammar too complicated to be practical. Nonetheless, this line of inquiry is one which the authors propose to investigate.

For the 3D shape grammar, its four configurational processes allow for the generation and semantic presentation of a conceptual 3D form that conforms to the syntactic and functional relations specified in the JPG. Furthermore, it contributes to the generation of new alternatives by altering the rule application. Admittedly, as a prototype example, the current grammar disregards many subtle architectural details, for example, each face of the block shapes currently has only one wall property. This limitation could be addressed, to accommodate a much higher level of detail mapping more closely to the particular design cases and styles being studied, but as such factors are often shaped by complex external issues, this too may be impractical to implement.

Ultimately, this paper combines selected aspects of two accepted computational methods to provide a new critical knowledge base about architecture in terms of forms, styles and spatial configurations, complementing and challenging our current understandings, which have often been exclusively constructed, debated and maintained by architects and architectural historians.

References

Alexander, C. 1964. *Notes on the Synthesis of Form*. Cambridge: Harvard University Press.

Amorim, L. M. D. E. 1999. *The sectors' paradigm: a study of the spatial and functional nature of modernist housing in Northeast Brazil*. London: University of London.

Brandenburg, F. 1995. Designing graph drawings by layout graph grammars. *Lecture Notes in Computer Science* **894**: 416-427.

Cagdas, G. 1996. A shape grammar: the language of traditional Turkish houses. *Environment and Planning B: Planning and Design* **23**, 4:443-64.

Cui, J., and M.-X. Tang. 2014. Representing 3D Shape Grammars in a Generative Product Design System. In *Design Computing and Cognition '12*, edited by John S. Gero, 377-392. New York; London: Springer.

Eloy, S. 2012. *A transformation grammar-based methodology for housing rehabilitation: meeting contemporary functional and ICT requirements*. Lisbon: Universidade Técnica de Lisboa.

Freudenstein, F., and E. R. Maki. 1979. The creation of mechanisms according to kinematic structure and function. *Environment and Planning B* **6**, 4:375-91.

Gero, J. S., and R. D. Coyne. 1985. Logic programming as a means of representing semantics in design languages. Review of. *Environment and Planning B: Planning and Design* **12**, 3:351-69.

Grzegorz, R. 1997. *Handbook of graph grammars and computing by graph transformation: volume I. foundations*. River Edge, NJ: World Scientific Publishing Co., Inc.

Hanson, J. 1998. *Decoding Homes and Houses*. Cambridge: Cambridge University Press.

Hanson, N. L. R., and A. D. Radford. 1986a. Living on the Edge : A Grammar For Some Country Houses by Glenn Murcutt. *Architecture Australia* **75**, 5:66-73.

Hanson, N. L. R., and T. Radford. 1986b. On Modelling the Work of the Architect Glenn Murcutt. *Design Computing* **1**, 3:189-203.

Heitor, T., J. Duarte, and R. Pinto. 2004. Combing Grammars and Space Syntax: Formulating, Generating and Evaluating Designs. *International Journal of Architectural Computing* **2**, 4:492-515.

Hillier, B. 1999. *Space is the Machine: A Configurational Theory of Architecture.* Cambridge: Cambridge University Press.

Hillier, B., and J. Hanson. 1984. *The social logic of space*. Cambridge: Cambridge University Press.

Kong, J., K. Zhang, and X. Zeng. 2006. Spatial graph grammars for graphical user interfaces. *ACM Trans. Comput.-Hum. Interact.* **13**, 2:268-307.

Koning, H., and J. Eizenberg. 1981. The language of the prairie: Frank Lloyd Wright's prairie houses. *Environment and Planning B* **8**, 3:295-323.

Lee, J. H., M. J. Ostwald, and N. Gu. 2013. “Combining Space Syntax and Shape Grammar to investigate architectural style: Considering Glenn Murcutt's domestic designs”. Paper presented at the 9th International Space Syntax Symposium, Seoul.

Lee, J. H., M. J. Ostwald, and N. Gu. 2014. “Using a JPG grammar to explore the syntax of a style: An application on Glenn Murcutt’s architecture”. Paper presented at the Design Computing and Cognition '14, University College London, London.

Li, X., and L. Schmidt. 2004. Grammar-based designer assistance tool for epicyclic gear trains. *Journal of Mechanical Design* **126**, 5:895-902.

March, L. 2002. Architecture and Mathematics Since 1960. In *Nexus IV: Architecture and Mathematics*, edited by Kim Williams and Jose Francisco Rodrigues, 7-33. Fucecchio (Florence): Kim Williams Books.

March, L., and P. Steadman. 1971. *The geometry of environment: an introduction to spatial organization in design*. London: RIBA Publications.

Osman, K., and M. Suliman. 1994. The Space Syntax Methodology: Fits and Misfits. *Architecture and Behaviour* **10**, 2:189-204.

Ostwald, M. J. 2011a. Examining the relationship between topology and geometry: A configurational analysis of the rural houses (1984-2005) of Glenn Murcutt. *Journal of Space Syntax* **2**, 2:223-46.

Ostwald, M. J. 2011b. A Justified Plan Graph Analysis of the Early Houses (1975-1982) of Glenn Murcutt. *Nexus Network Journal* **13**, 3:737-62.

Ostwald, M. J. 2011c. The Mathematics of Spatial Configuration: Revisiting, Revising and Critiquing Justified Plan Graph Theory. *Nexus Network Journal* **13**, 2:445-70.

Rekers, J., and A. SchÜRr. 1997. Defining and Parsing Visual Languages with Layered Graph Grammars. *Journal of Visual Languages & Computing* **8** (1):27-55.

Schmidt, L., and J. Cagan. 1997. GGREADA: A graph grammar-based machine design algorithm. *Research in Engineering Design* **9**, 4:195-213.

Steadman, P. 1983. *Architectural Morphology: An Introduction to the Geometry of Building Plans*. London: Pion.

Stiny, G., and J. Gips. 1972. Shape Grammars and the Generative Specification of Painting and Sculpture. In *Information Processing 71*, edited by C.V. Freiman, 1460-5. Amsterdam: North-Holland.

Stiny, G., and J. Gips. 1980. Production Systems and Grammars: A Uniform Characterization. *Environment and Planning B* **7**, 4:399-408.

Stiny, G., and W. J. Mitchell. 1978. The Palladian grammar. *Environment and Planning B* **5**, 1:5-18.