Lost in Translation:

Towards an Automated Description of John Portman's Domestic Architecture

- Heather Ligler
  Georgia Institute of Technology, USA
  h.ligler@gatech.edu

- Thanos Economou
  Georgia Institute of Technology, USA
  economou@gatech.edu

Abstract

The prevalent mode of shape grammar output is a two-dimensional drawing grammar. For architectural applications, these two-dimensional shape rules can hold a variety of interpretations in three-dimensional space. This work translates an existing grammar from a manual two-dimensional drawing grammar to an automated three-dimensional building grammar to explore the challenges and opportunities that this translation suggests in the larger context of shape computation. The case study considered here is a grammar interpreting John Portman's architectural language as defined by the house Portman identifies as emblematic of his design principles, his 1964 personal residence Entelechy I.

Keywords: Shape Grammars, Shape Grammar Implementations, Formal Composition, Generative Systems, John Portman

Introduction

Shape grammars are primarily simulated manually and drawn as two-dimensional grammars, although for architectural applications in particular, shapes represent a variety of meanings for building three-dimensional grammars in space. The reality is that despite architecture's three-dimensionality, the clarity of the plan remains a primary convention of architectural representation. Simultaneously, the plan necessarily represents the coordination of more than it can convey alone. Ultimately a variety of architectural representations are needed to collectively convey formal intent and even more to convey built form. This is a familiar observation in shape computation, resulting in efforts to develop parallel and three-dimensional shape grammars to clarify intentions (Koning and Eizenberg, 1981; Duarte, 2005) as well as shape grammar interpreters to illustrate grammars as computer implementations (Tapia, 1999; Hoisl and Shea, 2011; Trescak et al, 2012; Grasl and Economou, 2013). Despite these efforts, grammars remain close to their two-dimensional origins without an adaptable framework that can lend itself to multiple custom endeavors. Here, a shape grammar for John Portman's Entelechy I (Ligler and Economou, 2015) is featured as a case study to illustrate a process for translating an architectural language from a two-dimensional grammar to an automated three-dimensional grammar.

The history of shape grammars, particularly as applied to domestic projects (Stiny and Mitchell, 1978; Koning and Eizenberg, 1981; Duarte, 2005), suggests their formalism as a productive method to parse the compositional system embedded in Entelechy I. The existing grammar features forty-four shape rules in four stages that generate the plans of the house. Additionally, playful variations of alternative designs serve as proof-of-concept for the expressiveness of the language. The shape rules of the grammar are structured as schemas that generate simple single plan outputs as well as layered plans for multiple level schemes utilizing a variety of conventions including layers, labels, and levels. However, these planar representations leave much formal information lost in translation.

John Portman's bewildering architecture spans over five decades of a hybrid practice uniquely involving both architecture and development. The architectural language shaped by these dual, and often conflicting, considerations attracts critical interest (Goldberger, 1990; Jameson 1991; Sorkin, 1994; Koolhaas, 1998), yet little conversation is directed towards his formal contributions in architectural theory. While known primarily for his hotels and large-scale commercial works, Entelechy I (Figure 1) is Portman's 1964 personal residence in Atlanta, Georgia that he identifies as the singular work embodying his architectural philosophy (Portman, 1976). The concept for the house was to create a domestic pavilion where his family could live privately as well as entertain publicly with a sense of openness, natural light, plants, and water features. Portman's concept of "space within space" is evident both functionally and formally in the house. Functionally, the house is separated into two spatial zones: private family and public entertaining. Formally,

the house is designed around a grid of exploded columns ordering the floor plans to articulate two types of space: major and minor. Promoting order and variety, this arrangement maintains a set rhythm while allowing for flexibility and variation over time.

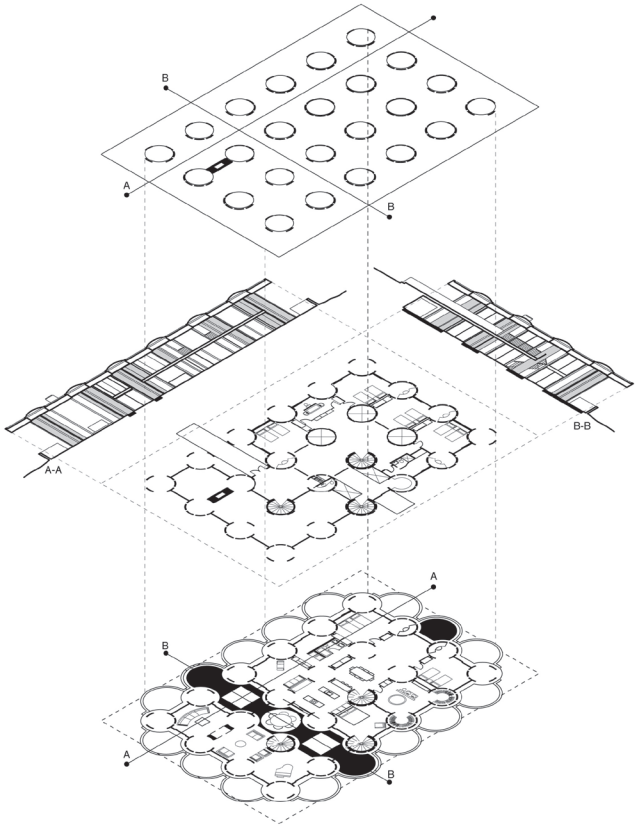

Figure 1: Entelechy I, 1964: composite illustration including plans and sections.

From 2D to 3D Grammars

The initial grammar for Entelechy I (Ligler and Economou, 2015) has a straightforward structure to capture intuitively the incremental process through design from basic concepts to developed articulation. In all, the grammar is comprised of four stages:

Stage 1: Framework
Stage 2: Configuration
Stage 3: Style
Stage 4: Termination

The first stage sets up the framework of the design and articulates its boundaries, underlying grid, and the two main spatial conditions of the plan(s): the delineation of the major and minor spaces. Significantly, this stage of the grammar defines the number of plans that will be generated in the process. These plan levels are treated as layers in the shape rules. The rules are always layered in the sense that if only one plan is generated the layered rules are null; if two plans are generated in a coordinated manner, layered rules apply; some rules only apply to an individual upper or lower layer; or all layers at once; and so forth. In this way, the layer grammar allows shape rules to function as overlays, while their output in production distinguishes the specificity of each layer as an individual plan level. This is analogous to the difference between the layers of a CAAD modeling space and the clean output of a coordinated design in paper space. Stage 2 develops the basic configurations within the system by concatenating spaces horizontally and vertically, opening up light wells, and adding staircases to connect levels. Stage 3 resolves interior and exterior details to fully articulate the style of the house. Finally, Stage 4 cleans up any remaining labels and finalizes the process. The rules within these stages are parametric too to provide both flexibility and precision in the design process. A basic representation of all shape rules in the four stages discussed above is shown in schemas in Figure 2. Most of the labels and the parameters of the variables of the rules are omitted for clarity of representation. The layered rules for the generative specification of additional plans are also omitted for clarity of representation.

The grammar outlined in Figure 2 is quite expressive; it can generate the original two plans of Entelechy I and a generous set of additional plans that are characterized by similar properties with the original plans of the house. These new plans come in a variety of scales and numbers ranging from the simplest plan depicting a single module in the language to plans and pairs of plans depicting more complex single or double-story mat building configurations to several n-tuples of plans depicting high-rise domestic configurations. A nice set of designs to illustrate the rising complexity of the language and its adaptation to more complex architectural programs and light considerations is given in terms of three designs generated for three distinct S, M, and L scales (Ligler and Economou, 2015). Significantly these three variations have been produced by a mixed modeling process involving a manual application of shape rules by drawing in a two-dimensional CAAD system as well as an automated application of planar rules by selecting possibilities among alternative rule applications. More specifically, the process involved drawing and deleting shapes in Rhino and drawing and applying rules in Grapeline, a web interface for a shape grammar in HTML 5 (Grasl and Economou, 2013, 2011). The final designs were produced using both techniques typically relying on some automated application of basic rules to fix the structure of the design and a manual application of several rules to build its detail. The exploded axonometric representation of Entelechy I in Figure 1 shows the two plans of the house generated by the grammar accompanied by manually drawn projections of the sections and the rooftop of the house. Solid lines have been inserted in the exploded axon to signify the planes of sections of the house and dotted lines respectively to foreground the correspondence of the functions between the first and second plan of the house.

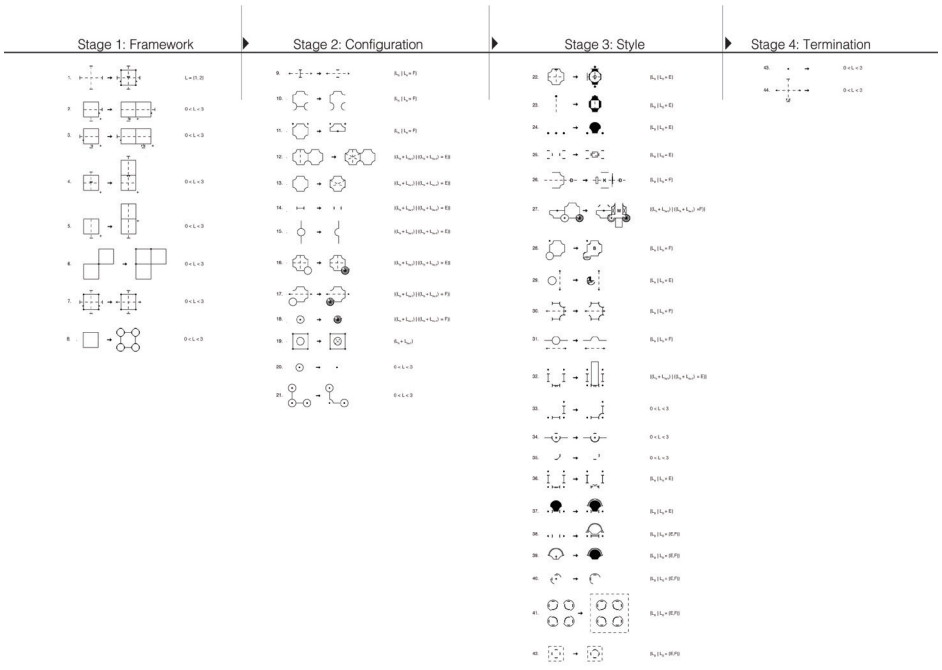

Figure 2: Shape rules in four stages to articulate Entelechy I.

Towards Building Grammars

The representational challenge of shape grammars for architectural applications, which with few exceptions are two-dimensional drawing grammars like the case study grammar for Entelechy I, is communicating three-dimensional building grammars that illustrate shape rules in space. One potential way to translate shape rules from two-dimensions to three is to simply extrude the planar rules. While this results in three-dimensional "spatial" relationships, it only adds one new type of information: height. While this may be all that is needed for some rules, ultimately this does not add much to clarify architectural relationships. Alternatively, the planar grammar can be redefined in the process of translation and incorporate formal information lost in the reduction to two-dimensional representation. This is the approach taken in this effort to rework the grammar for Entelechy I. The four stages ordering the grammar as well as the conceptual intent of the shape rules remain the same, but the shift in representation provides opportunities for modification.

The process to translate shape rules from two-dimensions to three is ideally facilitated by automated implementation. For the work outlined here, the GRAPE engine was utilized for translation, where the GRAPE (GRaph shAPE) plug-in for C# for Rhino 5 is paired with graph grammars implemented using GrGen.NET (Grasl and Economou, 2011). Within this computational framework (Figure 3), the two-dimensional drawing grammar of Entelechy I can be transformed into a three-dimensional building grammar. The major effort in the process is to rewrite two-dimensional shape rules as graph grammars coded following C# conventions. As each rule is written, it can be compiled through the graph rewriting system Gr.Gen.NET and updated in the Rhino 5 plug-in. Compiled rules can be procedurally applied in Rhino 5 to generate three-dimensional geometry in the familiar design space of existing software. After executing a rule set in the plug-in, the user is left with a model that can be further manipulated and developed by transformations using any existing Rhino tools. Once this framework is setup, the main effort to code the grammar can begin. Coding the rules is a trial-and-error process using nodes, edges, labels and a variety of attributes to define graph equivalents of shape rules. These rules are coded individually and modeled after shape rules with both a left-hand and right-hand-side. An example of this is shown in Figure 3. In this case, the two-dimensional shape rule fills in the corners of the underlying grid recursively until the rule no longer applies and a complete three-dimensional grid is produced. The code for the rule shown is constructed with alternatives so that the rule can complete the initial grid both in a single level and for any additional levels added vertically. To do this, there are two left hand sides to the rule: the first looks for nodes in a diagonal relationship and the second looks for L-shapes including two nodes stacked on top of each other. For each left hand side, a right hand side to fill with a subsequent node is written. Finally, the last part of the rule is written to recursively apply the rule as many times as a match is found. This allows the user to select the rule once and complete the grid framework for all planar and vertical growth defined before the rule is applied. In all, the process to code the shape rules involves reworking and reevaluating the initial grammar to add rules, subtract rules, combine rules, and transform them based on the combined requirements of graph grammars and resultant three-dimensional outputs.

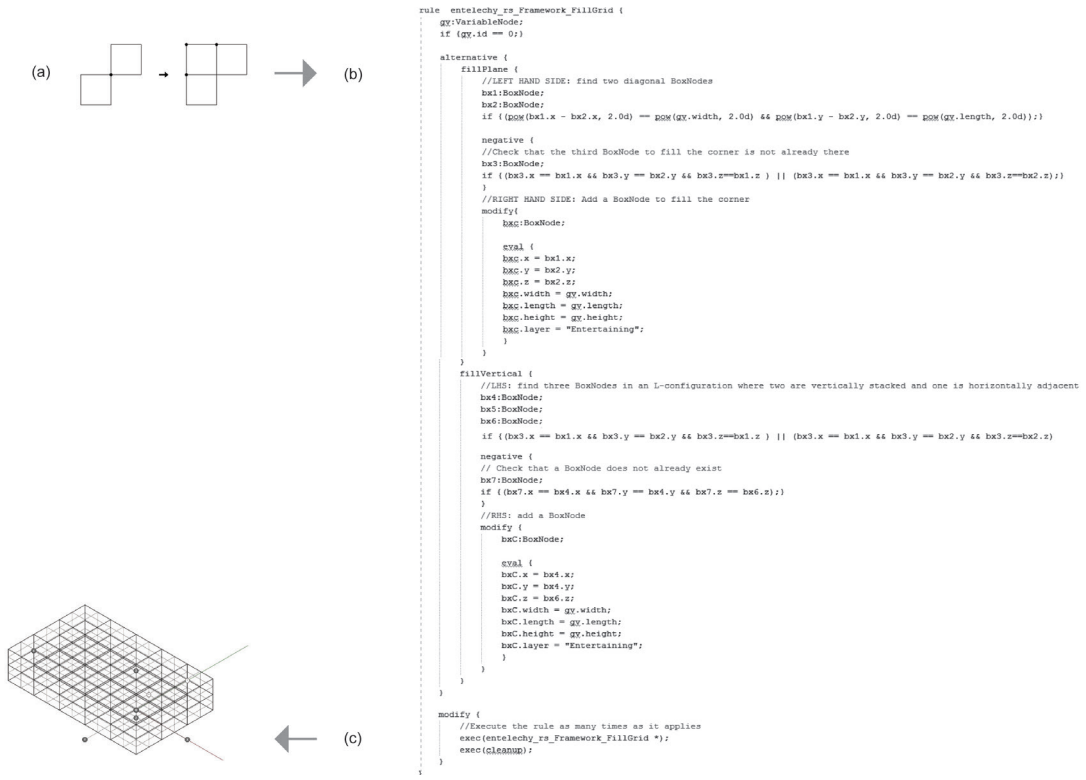

Figure 3: Diagram of the process: (a) two-dimensional shape rule; (b) C# coded rule; (c) three-dimensional implemented rule in Rhino 5.

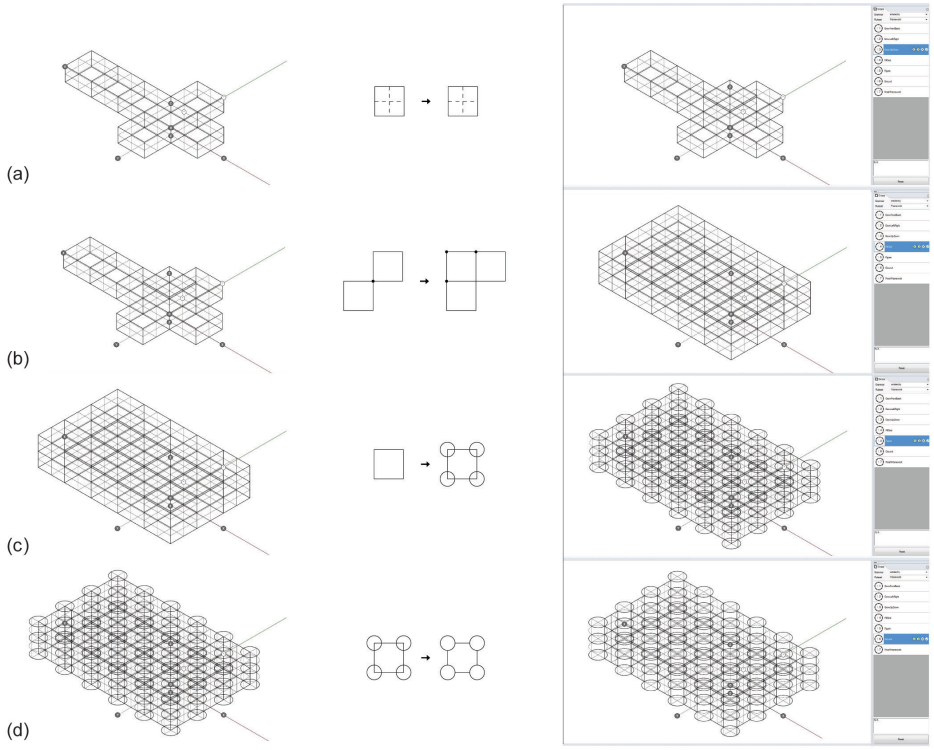

Figure 4: Composite production of three-dimensional rules implemented in GRAPE for Rhino with the equivalent two-dimensional shape rule illustrated in the center: (a) defines vertical growth; (b) completes an initial spatial lattice horizontally and vertically, (c) adds cylinders as figures at intersections of the grid, and (d) completes the Boolean operation to define major and minor spaces.

Translating the grammar as an automated description creates an interactive design tool for generating variations of Portman's domestic architecture that can be visualized and utilized for a variety of purposes. The work so far nicely illustrates this flexibility and its potential. Figure 4 shows the translation of a series of two-dimensional shape rules of Stage 1 of the grammar to their equivalent three-dimensional rules in the GRAPE for Rhino 5 plug-in. The freedom of the system allows for many interpretations and uses within a familiar CAAD environment where shape rules can be considered both for their original application in the grammar and for useful extension in other compositional pursuits. This work opens up a set of possibilities for active shape grammars that can be creatively applied, while also suggesting a process for testing and reworking shape rules in an automated environment.

Discussion

Architectural composition is an active process that involves working back and forth between representations that simulate the varied relations of a final construction. Abstract elements are utilized in this process of formation to map to requirements of the final form. An initial abstraction is helpful to foreground the important aspects of a particular design as well as to free the intuition to interpret them in multiple ways. Conventional representations in two and three-dimensions are abstract design tools utilized to calculate these acts of composition although they can only partially capture the implications of actual spatial construction. Instead, a layered process moving across both two and three-dimensional representations is necessary for the coordination required of an architectural design. These relations are a key architectural act and are hierarchically ordered in any design based on how they can resolve the design problem for a desired architectural performance. No matter what criteria they consider (function, light, circulation, systems, structure, etc.), these architectural relations are by definition spatial. Shape grammars represent these spatial relationships defined by architectural compositions as shape rules. Their simple computational means utilizing shapes and visual rules are familiar to architectural designers who use similar constructs to sketch and work through the multiple design relations of any given project.

Architectural relationships necessitate coordination in three-dimensional space. Where shape grammars aim to identify spatial relationships of a design, they are challenged by the limits of their traditional setup as two-dimensional drawing grammars in the context of architecture. Here we have proposed one process to translate from a two-dimensional drawing grammar to a three-dimensional building grammar to describe John Portman's architectural language as expressed in Entelechy I. This work begins to address a process of working from visual computation to symbolic computation and back again in a productive way that suggests an expansion of potential applications of shape computation. Shape grammar discourse in this context could involve flexibility to rework, reinterpret, and apply shape rules across varied contexts to foreground different design issues in an automated CAAD environment.

In the specific context of John Portman's work, this research aims to explore shape grammars as a method of visually calculating both new and existing designs by extracting the compositional information of a single architectural work representative of design principles applied to a larger, diverse corpus. Design logic across varied contexts can then be formally outlined and explored constructively. Recent recasting of shape rules and rule schemata (Economou and Kotsopoulos, 2014) emphasize the generous and often ambiguous formalism of shape grammars (Stiny, 2006) as illustrative of the expressive and productive possibilities that are currently underdeveloped within shape computation discourse as a whole. A plural approach to shape grammars is suggested as a method to explore their full potential as tools to understand Portman's formal contribution and suggest their larger application in the pursuit of creative architectural design.

Acknowledgments

GRAPE (GRaph shAPE) is a plug-in written for C# for Rhino 5. The graph grammars were implemented using GrGen.NET. We are indebted to Thomas Grasl for his collaboration, thoughtful advice, and continual support.

Reference

Duarte J, 2005. Towards the mass customization of housing: the grammar of Siza's houses at Malagueria. 30th eCAADe Conference Proceedings. 347-380.

Economou A and Kotsopoulos S, 2014. "From Shape Rules to Rule Schemata and Back." in Design Computing and Cognition DCC'14. J.S. Gero and S. Hanna (eds), Springer 2014, 419-438

Goldberger P, Portman J and Riani P, 1990, John Portman (L'Arca)

Grasl T and Economou A, 2011. "GRAPE: Using graph grammars to implement shape grammars" in SimAUD 2011.

Grasl T and Economou A, 2013. From topologies to shapes: parametric shape grammars implemented by graphs. Environment and Planning B: Planning and Design, 40(5), 905-922.

Hoisl F and Shea K, 2011. An interactive, visual approach to developing and applying parametric three-dimensional spatial grammars. Artificial Intelligence for Engineering Design, Analysis and Manufacturing, 25, 333-356.

Jameson F, 1991. Postmodernism or, the Cultural Logic of Late Capitalism (Duke University Press, Durham)

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

Koolhaas R, Mau B and Werlemann H, 1998. S,M,L,XL (Monacelli, New York)

Ligler H and Economou A, 2015. "Entelechy I: Towards a Formal Specification of John Portman's Domestic Architecture" in Proceedings of the 33rd eCAADe Conference, Martens B., Wurzer G., Grasl T., Lorenz W.E., and Schaffranek, R (eds), Volume 1, 445-452.

Portman J and Barnett J, 1976. The Architect as Developer (McGraw-Hill, New York)

Sorkin M, 1994. Exquisite Corpse: Writings on Buildings (Verso, New York)

Stiny G and Mitchell WJ, 1978. The Palladian Grammar. Environment and Planning B. 5(1). 5-18.

Stiny G, 2006. Shape: Talking and Seeing and Doing. MIT Press.

Trescak T, Esteva M, and Rodriguez I, 2012. A shape grammar interpreter for rectilinear forms. Computer-Aided Design, 44(7), 657-67