Integrating Computation in Foundation Design Instruction.

Sotirios D. Kotsopoulos
Boston Architectural College
skots@alum.mit.edu

The paper outlines notions suitable for the introduction of computation at the first stages of design education. An exercise, based on a housing competition sponsored by the Habitat For Humanity in Boston, Massachusetts, is presented. The educational objective of the paper is to offer an elementary case of how computation can be used in designing from scratch, in the architectural studio. Shape grammar formalism, analogue, and CAD tools are combined in this effort.

Introduction

In the past two decades architects have been introduced into new computing tools that had profound impact on the productivity of architectural firms. Computation greatly influenced the way we understand the relationship between design and the various forms of modeling and representation of design information. The rapid expansion of the new technological tools caused a disconnection between the existing design practices and the emerging computational ones.

This disconnection becomes apparent in the context of design education. Students use computers for drafting and modeling, or they experiment in specialized computational courses isolated from the architectural curriculum. At the end, design and computation remain segregated. Beginning from this problem, the paper outlines some elementary notions appropriate for the introduction of computation at the first stages of design education. I discuss how the notion of computational procedure can be employed in common goal-driven studio tasks, and how analogue and digital means can coexist as complementary parts of the educational experience.

The hope is to promote the integration of computation in foundation design instruction, while triggering the creative potential that lies in the new technologies. As example, the paper describes a studio exercise in formal composition, combining the use of shape grammar formalism, analogue, and CAD tools.

Objectives - Method

Housing projects occupy a part of the architectural studio instruction. Developing an exercise on housing was deemed ideal for the introduction of computation to architecture students. The exercise was based on a competition sponsored by the Habitat For Humanity (HFH) the summer of 2002 in Boston, Massachusetts. The program of the competition called for adaptable 2, 3 and 4-bedroom low cost houses including a primary covered entrance, circulation areas, dining area, living area, at least one full bathroom, kitchen, and bedrooms.

A minimum space limit for all house types was set: 900 s. f. for 2-bedroom, 1050 s. f. for 3-bedroom, and 1150 s. f. for 4-bedroom apartments. The organizers did not designate specific sites. The exercise emphasized three factors: a) the building program, b) the provision for low construction cost, and c) the absence of specific sites. Further, the exercise encompassed some of the principles of the domino house concept, suggested by Le Corbusier in 1926-29. Starting from an initial number of rooms the design objective is to develop compositional principles able to produce house arrangements of variable size and morphology. The systematization of the ground plan is the method employed for the attainment of this objective. In order to achieve certain room adjacencies spatial relationships are first set out.

Then, the students determine the productive actions leading to the generation of the desirable relationships and they express them as computational rules. Analogue and digital tools are jointly used in the design process. Through an iterative process the spatial vocabulary and the rules are tested and refined according to their compliance to programmatic, intuitive, and other criteria.

The overall approach is supported by the computational framework defined in Stiny 1980; 1991. The framework allows a vocabulary of shapes of spatial dimension i, to be composed in spatial dimension j ($i < j$) with the aid of rule schemata and rules. Rule schemata are computational devices able to describe the interaction of spatial entities in a general manner. A rule schema of the form: $g(x) - g(y)$, determines rules each time the variables x, y are substituted by specific instances. A predicate g is used to specify the attributes of x and y. A rule is a rule schema that contains no free variables.

The action of shape rule schema on some instance C of a shape to produce a new shape C' happens according to the relation: $C' = [C - t(g(x))] + t(g(y))$. First, a transformation t matches some part of the shape C geometrically similar to the shape g(x), which appears on the left side of the rule. Second, the same transformation t is used to subtract g(x) from C and to add g(y), which appears on the right side of the rule, in its place. Further, shapes and symbols can be combined to capture the generation of design sets with specific properties. For example, Stiny and Mitchell (1978) describe the production of Palladian villa plans in a computational process of eight productive stages. Relevant examples include rule systems for the generation of Frank Lloyd Wright's prairie houses (Koning and Eizenberg 1981), Japanese tea-room designs (Knight 1981), Queen Ann houses (Flemming 1987), Taiwanese houses (Chiou and Krishnamurti 1995), Yingzao fashi houses (Li 2000), and Alvaro Siza's houses (Duarte 2005).

Common aspect in the course of the above efforts is to show how rule systems encode architectural styles. Understanding the compositional principles of a design corpus by a grand master like Palladio, Frank Lloyd Wright, or Alvaro Siza is of great educational value for the experienced students. But, novices are usually overwhelmed by the attempt to put into use such principles. The novel aspect of the proposed exercise is that it focuses on how rule schemata can capture the exploratory effort of designing from scratch. Students are propped to develop their own elementary vocabulary and compositional principles and to test them.

Development

In the exercise, a house is approached as an arrangement of rooms. First task of the students is to identify the possible rooms. A room may accommodate more than one functional area, (i.e. the living room can include a dinning area, etc.). Rooms are initially represented minimally as parametric parti rectangles made out of lines. After some candidate vocabulary of rooms is set, the students form possible adjacencies.

Spatial relationships are set to represent room adjacencies. Two rooms can form four spatial relations: (1) they can share a common boundary, (2) they can be placed so that they do not touch, (3) they can meet at a corner point, or (4) they can share some area, or be placed one inside the other (Table 1, Fig. I). The spatial relation 1 can be used to depict two adjacent rooms having a common boundary.

The relation A (Table 1, Fig. II) is an instance of the relation 1. Four more parametric spatial relations are set. The five parametric relationships A, B, C, D, E can be produced by two rule schemata (Table 1, Fig. III). On the left, the rule schema R1AB produces the relations A and B, where a parametric rectangle is added on the short side of an initial parametric rectangle. On the right, the rule schema R1CDE produces the relations C, D and E, where a parametric rectangle is added on the long side of an initial parametric rectangle. Labels can be used to restrict the two rule schemata to produce only certain adjacencies.

A derivation of a two-bedroom parti, beginning from the living room and continuing with the addition of the kitchen, the auxiliary spaces, and the bedrooms appears next in plan and axonometric (Table 1, Fig. IV).

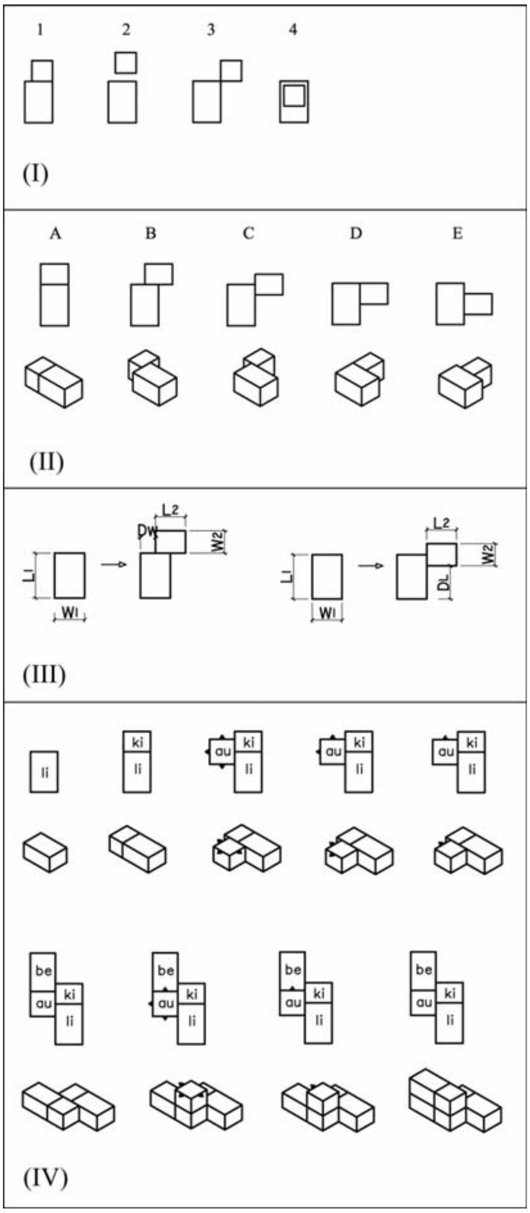

Table 1

Table 1.

Initially, candidate sets of spatial elements and rule schemata are described with pencil and paper. Then, they are converted into LISP scripting format. A parametric interpreter (Liew 2003) is used for the digital part. The digital interpreter allows fast in breadth exploration of the produced results. A vector description format (Nagakura 1995) is used to describe the geometry and variables of a shape rule schema.

A transformation mapping determines the transformation changes between the left-hand schema and the right-hand schema. Variable mappings define a relationship between the parameters of both schemata. A rule schema is composed of two parts: the geometry, and the constraints on the geometry variables.
The geometry of a parametric rule schema is described using a series of vector displacements. Each vector has 3 components: action, vector and label. The action component determines if the shape is a line or a point. The vector component describes the x and y displacement of the shape. The label component determines the name of the label.

A shape is described as a series of vector displacements that are connected from end to end. To describe a parametric schema, the values in the vector displacement description are substituted with variables. To apply a rule schema x → x + y, like the ones used in the exercise, the interpreter recursively searches the input shape for all instances of the left-hand schema x and presents the options through an interactive menu that highlights the embedded schemata. Once one selects an embedded schema, the rule application is completed by subtracting the selected schema from the input shape and adding the right-hand x+y schema of the rule. The next code describes the additive rule between two rectangles that appears in Table 1, Fig. III, right.

The left-hand schema describes in symbols the left-hand shape of the rule. The right-hand schema describes the right-hand shape of the rule. There is a part where the transformation and the parameter relationships between left and right-hand shapes are set and, a fourth part where all the previous three parts are linked.

First, the left-hand schema,

```
(setq schema-left-rule
  ‘((geometry
   ((action “line”) (vector W1  0) (label “parti”))
   ((action “line”) (vector 0  L1) (label “parti”))
   ((action “line”) (vector (- W1)  0) (label “parti”))
   ((action “line”) (vector 0  (- L1)) (label “parti”))
    )
    (parameter-constraints
    (W1  (> W1  0))
    (L1  (> L1  W1))
    )
   )
  )
```

Second, the right-hand schema of the rule,

```
(setq schema-right-rule
 '((geometry
  ((action "line") (vector W1  0) (label "parti"))
  ((action "line") (vector 0  L1) (label "parti"))
  ((action "line") (vector (- W1 ) 0) (label "parti"))
  ((action "line") (vector 0  (- L1)) (label "parti"))
  ((action "move") (vector W1  (- L1  (* 0.375  W1))))
  ((action "line") (vector L2  0) (label "parti"))
  ((action "line") (vector 0  W2) (label "parti"))
  ((action "line") (vector (- L2)  0) (label "parti"))
  ((action "line") (vector 0  (- W2)) (label "parti"))
  )
  (parameter-constraints
  (W1  (> W1  0))
  (L1  (> L1  W1))
  (L2  (> L2  0))
  (W2  (> W2  0))
  )
 )
)
```

Definition of the transformation and parameter mapping,

```
(setq tmap-rule
  '((delta-xo . 0)
    (delta-yo . 0)
    (delta-ro . 0)
    (delta-za . 0))
)
(setq pmap-rule
  '((W1  W1)
    (L1  L1)
    (L2  W1)
    (W2  (* 0.75  W1))
   )
)
```

Finally, an expression connecting the left and the right-hand schemata of the rule,

```
(setq housing-rule
  '((left . schema-left-rule)
    (right . schema-right-rule)
    (tmap . tmap-rule)
    (pmap . pmap-rule)
    (success . nil)
    (failure . nil)
    (applymode . "single")
    (rulename . "housing-rule")
    )
)
```

The rule schemata used in the overall process can be arranged in three productive levels, guiding the students to achieve some general objective. At the top level, formation rule schemata produce parti diagrams. At the middle level, a chosen parti is transformed into a boundary-layout. At the third level, refinement rule schemata apply on chosen boundary-layouts to determine tectonic details (windows, doors, etc.). Formation initiates the design process, while transformation and refinement are dedicated to the development of designs. The outcome of any level may influence the preceding and the subsequent level.
The framework is outlined as follows,

Σ: { finite set of rooms }

$$R: \left\{ \begin{array}{ccccc} \text{Formation} & & \text{Transformation} & & \text{Refinement} \\ A_1 \rightarrow F_1 & & G_1 \rightarrow M_1 & & N_1 \rightarrow W_1 \\ \vdots & \leftrightarrow & \vdots & \leftrightarrow & \vdots \\ A_n \rightarrow F_n & & G_k \rightarrow M_k & & N_r \rightarrow W_r \end{array} \right\}$$

where $A_1,\ldots, A_n$, are elements in Σ.

The Table 2 presents examples of produced designs. Three design subsets are shown: A_1-A_4, B_1-B_4, C_1-C_4. In each subset, the top two rows present partis in plan and axonometric, corresponding to the products of formation. In the third row, views corresponding to refinement are shown in axonometric. A 3d model is provided for selected designs, in the fourth row.

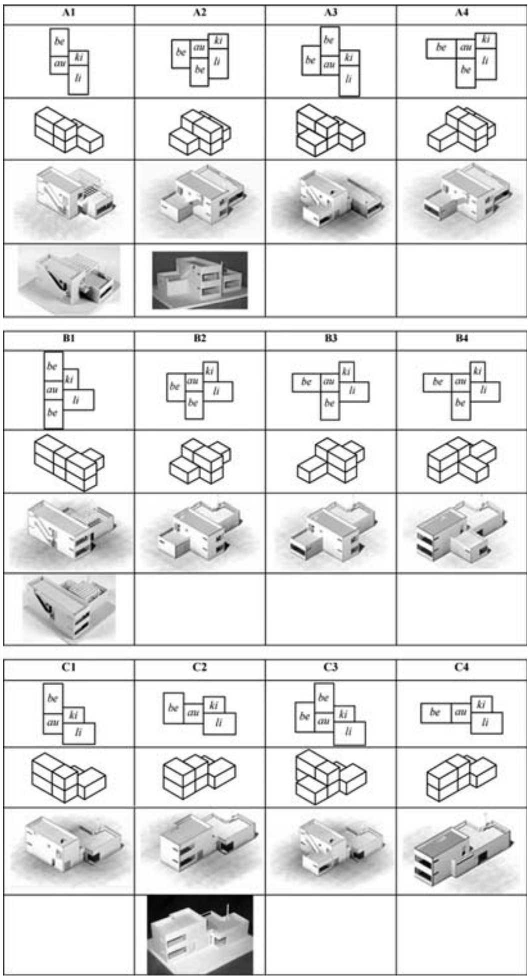

Table 2.

Conclusions

The paper outlines some elementary notions suitable for the introduction of computation at the first stages of design education. It shows how the notion of computational procedure can be employed in goal-driven studio tasks, and how analogue and digital means can coexist as parts of the same educational experience. As example, the paper presents an exercise in formal composition based on a low cost housing competition. The design objective of the exercise is to produce houses of variable size and morphology. The educational objective is to offer an elementary case of how computational rule schemata can be used in designing from scratch, in the studio.

The systematization of the ground plan is the method employed for the attainment of these objectives. Shape grammar formalism, analogue, and CAD tools are the means used. The students are propped to develop their own elementary spatial vocabulary and compositional principles, and to test them. Students begin from abstract arrangements and rule schemata to gradually conclude to specific house designs and compositional rules.

The heuristic of the design process is arranged in three levels: Formation rule schemata produce parti diagrams. Transformation rule schemata transform a chosen parti is into a boundary-layout. Refinement rule schemata determine tectonic details. Interconnected formats of description, analogue and digital, are used throughout the design process. Through an iterative process of formation, transformation and refinement, the spatial vocabulary and the rule schemata are gradually refined according to their compliance to programmatic, intuitive, and other criteria.

References

Chiou S and Krishnamurti R: 1995, 'The grammar of the Taiwanese traditional vernacular dwellings', Environment and Planning B: Planning and Design 22 689-720

Duarte J P: 2005, Towards the mass customization of housing: the grammar of Siza's houses at Malagueira, Environment and Planning B: Planning and Design 32 347-380

Flemming U: 1967, 'More than the sum of parts: the grammar of Queen Anne houses' Environment and Planning B: Planning and Design 14 323-350

Knight K: 1981, 'The Forty-one Steps: the languages of Japanese tea-room designs', Environment and Planning B: Planning and Design 8 97-114

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

LeCorbusier: 1954 (2000), The modulor, Birkhauser, Basel, Switzerland

Li A: 2000, A teaching grammar of the Yingzao fashi, Ph.D. Dissertation, Department of Architecture, Massachusetts Institute of Technology, Cambridge, Massachusetts.

Liew H: 2003, SGML: A Meta-Language for Shape Grammars PhD Dissertation, Department of Architecture, Massachusetts Institute of Technology, Cambridge, Massachusetts.

Nagakura T: 1995, Form-processing: A system for architectural design, Ph.D. Dissertation, Harvard University, Massachusetts

Stiny G: 1980, 'Introduction to shape and shape grammars', Environment and Planning B, volume 7, pp. 343-351

Stiny G: 1991, 'The algebras of design', Research in Engineering Design, 2, pp. 171-181

Stiny G and Mitchell W J: 1978, 'The Palladian grammar', Environment and Planning B: Planning and Design, volume 5, pp. 5-18

Keywords:
Rule Schema, Formation, Transformation, Refinement.