FORMALIZING GENERATION AND TRANSFORMATION IN DESIGN

A Studio Case-Study

OMER AKIN, HODA MOUSTAPHA
Carnegie Mellon University, Pittsburgh, PA, USA

Abstract. This paper is an integration of two substantial endeavours. One is a general purpose 3D modelling system, ICE that introduces a new notation and an entire family of graphic design functionalities based on generative structures and manipulation handles. The other is an exhaustively annotated design studio, in which the entire graphic output of students and the annotations of their faculty have been ethnographically recorded. In this paper, we are using the ICE notation to represent the key graphic products of a selected student and the transformations between these representations. Our goal is to demonstrate that, through ICE's formal notation (1) graphic entities in complex design sequence can be unambiguously represented, (2) transformations between graphic entities in complex design sequence can be unambiguously represented, and (3) the various design sequences can be formally captured for subsequent process or cognitive analysis

1. Motivation

This paper is motivated with the goal of codifying design (taken both as a noun and a verb) unambiguously and formally. We believe this will lead to quantifiable representations that can help analyse designs and design generation for cognition and intent capture in design. There are two motivating ingredients of this study. One is a general purpose 3D modelling system, ICE (interactive configuration exploration) that introduces a new notation and an entire family of graphic design functionalities based on generative structures and manipulation handles, (Moustapha and Krishnamurti 2001). The other is an exhaustively annotated design studio in which the entire graphic output of students and the annotations of their faculty have been ethnographically recorded. We will briefly introduce these ingredients below. In the following sections, we will elaborate each one and conclude with a discussion of the implications of our approach for design cognition and intent capture.

1.1. THE ANNOTATED STUDIO

A vertical design studio in the School of Architecture, at Carnegie Mellon University was offered during the summer of 2002, by Professor Omer Akin. There were six students taking the studio, one having completed the 2nd year, two the 3rd year and three the 4th year of their college education. The entire studio work was recorded through digital photographs of student work brought to each class session and the midterm and final reviews (Akin, 2002). These graphic records were accompanied by daily diary annotations kept by the instructor for each student's progress as well as the overall progress of the studio.

Students were invited to define their own design programs or continue with a previous design problem either they experienced or experienced by their peers. Three different problems emerged, international housing prototype, dormitory housing, and a toy manufacturer's headquarters building. The studio work was complemented by visits by external faculty on a weekly basis. They gave feedback to students on their work through critics and presentations of their own work. The same faculty served on the midterm and final reviews of the studio. A typical annotation for a studio day contains segment like the one provided below for each student and some directed to the general issues in the studio.

TABLE 1. Typical annotation for a studio segment for Thursday, May 23, 2002

Subject-W's has created a swirling shape that expresses the housing hierarchy: rooms, units, unit-clusters, wings, buildings, building-clusters. I tell her[1] to examine her ideas spatially *vis a vis* a model. I remind her of her own narrative about the new dorm lounge spaces that do not attract any social gatherings due to being out of the way. Making a conceptual diagram about social hierarchy does not always work in actual spatialization because connectivity must be solved in another medium, namely a physical model.	 Extract from *Design The Art and Science of the Synthetic* unpublished manuscript by Ömer Akin ©

1.2. THE ICE NOTATION AND GENERATION SYSTEM

The ICE system has two major components. (1) A formal notation for representing complex design configurations based on their underlying generative and relational structures. (2) An interactive-generative

[1] We used "she" or "her" to refer to all subjects--students and critics--of the annotated studio for the purpose of anonymity. No gender implications are intended.

computational system that supports the interactive exploration of design configurations by means of the constructs of the notation.

The formal notation summarizes any configuration into the set of minimal steps required for its generation, as well as the set of meaningful relationships required for its organization. These generative steps or relationships form the basic units of the configuration's structure, referred to as "regulators". Regulators are combined in various ways to represent the diverse types of structures observed in architectural configurations, for example symmetry, proportion, rhythm, and gradation among many others. In the ICE system, regulators control other design elements: regulators are used as handles to manipulate the design; these also control the changes happening in the design, when it is manipulated by the user. In other words, regulators are higher level entities that "regulate" the behavior of lower level design elements, hence the name regulator.

A regulator encapsulates a formula, by which it computes the positional (or other) attribute of the elements it regulates. A regulator can be a transformation, a constraint, an operation, or a variation. For instance: translations, rotations, reflections, are transformation regulators. These, along with the dilation (scale), and the curve transformations, constitute the primary regulators used for generating shapes and complex configurations. Alignments and containments are constraint regulators. These, among many others, constitute the relational regulators that further control the configuration elements. The section entitled "The ICE Notation Syntax" illustrates how shapes and designs can be generated using regulators, and their corresponding description using the ICE notation.

The ICE system is a 3D exploration tool, which uses regulators to construct design configurations, as well as to interactively manipulate those configurations. Regulators allow the structure (both generative and relational) of configurations to be redefined at any time during the exploration process.

1.3. THE INTEGRATION OF THE STUDIO AND THE NOTATION

We integrate these two endeavors to illustrate the potential of the ICE notation for formally describing a realistic design situation, in which the configuration is evolving such as the one presented in the annotated studio. This allows us to use the ICE notation to codify the design process in stages, which are defined by each drawing in the sequence of the design development. We also use the ICE notation to codify the transformations from each drawing in the sequence to the next one. We intend to show that significant design transformations, that would normally be quite labor intensive, can be described in a simple-straightforward way by using our ICE notation and consequently can be easily achieved using the ICE system.

2. The ICE Notation's Features and Syntax

The basic elements of the ICE notation are (1) the point, denoted by a lowercase letter, for instance p, and (2) the regulator, denoted by a bold uppercase letter, for instance **T**. Shapes are composite objects defined by points and regulators and are denoted as lowercase words. A prefix for the regulator, a Greek letter, indicates the type of regulator: Δ transformations, Φ constraints, Ξ variations, and Ω operations. Superscripts indicate the subtype for the regulator: for example $\Delta\mathbf{C}^{\mathbf{p}}$, $\Delta\mathbf{C}^{\mathbf{h}}$, indicates two types of curve regulators (i.e. two distinct formulae). Subscripts (for regulators, shapes, and points) are used for indexing to differentiate elements of the same type for instance $\Delta\mathbf{T_1}$, $shape_3$.

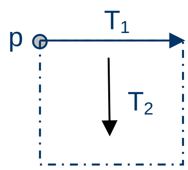
We developed two forms for the ICE notation, a short form, which captures the regulator and the regulated object/s, for instance $\Delta\mathbf{T}(\text{shape})$ and the expanded form, which also shows the parameters of the regulator enclosed in curly brackets with vectors depicted with an overline: $\Delta\mathbf{T}^{\mathbf{1}}[\ \{\bar{p}, \bar{t}, d, n\}\ (\text{shape})\ ]$. These include translation vectors and distances, rotation points and degrees, reflection and glide axes, etc. The long form is essential for system implementation. However, for the purpose of simplicity, we use the short form for all examples in this paper.

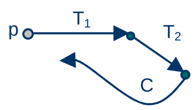
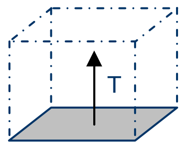
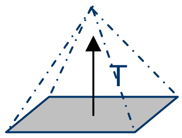
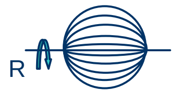
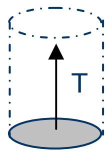
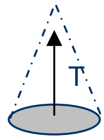
The conjunction $\wedge$ (and) is used to join two related clauses which share regulated objects. For instance: $\Delta\mathbf{T}(\text{shape}) \wedge \Delta\mathbf{A}(\text{shape})$.

The ICE notation has the following distributive property: $\Delta\mathbf{T}(\text{shape}_1, \text{shape}_2) = \Delta\mathbf{T}(\text{shape}_1) \wedge \Delta\mathbf{T}(\text{shape}_2)$.

Table 2 shows how the ICE notation is used to describe shapes.

TABLE 2. The ICE notation for generating simple shapes

Straight line: $\Delta\mathbf{T}(\text{p})$ The translation regulator $\Delta\mathbf{T}$ sweeps the starting point p and generates a straight line.	
Curved line: $\Delta\mathbf{C}(\text{p})$ The curve regulator $\Delta\mathbf{C}$ sweeps p to create a curved line. Subtypes include quadratic and cubic curves.	
Plane: $\Delta\mathbf{T_2}(\Delta\mathbf{T_1}(\text{p}))$ A plane is generated by the application of successive regulators. The second regulator takes as input all the generated points of the previous regulator. Note that the $\mathbf{T}_1$ regulator is applied first.	

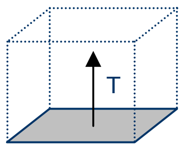
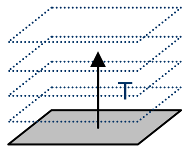
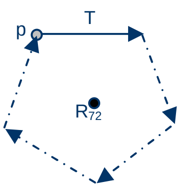
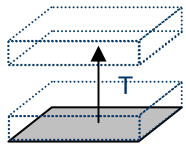
Polyline: $\Delta\mathbf{C}(\Delta\mathbf{T_1}(\Delta\mathbf{T_1}(\mathrm{p})_{\#n})_{\#n})$ A polyline is also generated by the application of successive regulators, each one taking, as its input, only the last generated point of the previous regulator. The subscript indicates the items taken as input for the next regulator.	
Prism: $\Delta\mathbf{T_3}(\mathrm{base}) \wedge \mathrm{base} = \Delta\mathbf{T_2}(\Delta\mathbf{T_1}(\mathrm{p}))$ A prism is generated by sweeping a square base along the translation regulator $\Delta\mathbf{T}$ **.**	
Pyramid: $\Delta\mathbf{TD}(\mathrm{base})$ A pyramid is generated by sweeping a square base along a straight line while incorporating a dilation regulator D. If the scale factor is decreased, the result is a frustum. When two regulators are applied simultaneously, they are denoted as two juxtaposed regulator symbols: $\Delta\mathbf{TD}$ **.**	
Sphere: $\Delta\mathbf{R_2}(\mathrm{circle}) \wedge \mathrm{circle} = \Delta\mathbf{R_1}(\Delta\mathbf{T}\ (\mathrm{p}))$ A sphere is generated by sweep-rotating a circle using the rotation regulator $\Delta\mathbf{R}$	
Cylinder: $\Delta\mathbf{T}(\mathrm{circle})$ A cylinder is generated by sweeping a circular base along the translation regulator $\Delta\mathbf{T}$.	
Cone: $\Delta\mathbf{TD}(\mathrm{circle})$ A cone is generated by sweeping a circular base along the translation regulator $\Delta\mathbf{T}$ composed with a dilation $\Delta\mathbf{D}$.	

The ICE notation supports several ways of generating shapes: continuous generation, discrete generation and in combination. These allows for the description of any sub part of a shape. This feature is indicated by superscripts as illustrated in table 3.

3. A Sample Generative Sequence from the Studio Data

In this section, we present a series of sketches and models created by Subject-W. She is one of the students working on the dormitory housing project. The sequence we present here starts about a quarter of the way into the studio and runs through to the end, highlighting all major formal solutions produced. The data consists of annotations for each day in which the sketch or the model was created. We also include the ICE notation for each graphic display (see table 4). This is a hypothetical description of each design stage after their completion; these designs have been generated by Subject-W independent of the ICE system.

TABLE 3. The ICE notation for generating sub-shape

Continuous : $\text{Cube} = \Delta\mathbf{T}_3(\text{square})^{<0-4>}$ The superscript indices enclosed in one set of brackets $^{<0-4>}$ indicate a continuous generation. The connecting line indicates that all indices between the given ranges are generated.	
Discrete: $\text{Cube} = \Delta\mathbf{T}_3(\text{square})^{<0>-<4>}$ The superscript indices, each enclosed in a separate set of brackets $^{<0>--<4>}$ indicate a discrete generation. The discrete generation is used to define complex configurations composed of disjoint, yet related, shapes.	
Regular polygon: $\Delta\mathbf{R}(\Delta\mathbf{T}((\text{p})^{<0-n>})^{<0>-<4>})$ An outline polygon is generated by translating a point to construct a line, then by rotating the line, discretely, to construct the remaining sides.	
Combination: Cube = $\Delta\mathbf{T}(\text{p})^{<0-1><3-4>}$ Here, the superscript items are grouped by brackets $^{<0\text{-}1><3\text{-}4>}$ indicating a combined generation. Each group enclosed in a bracket is a continuous group. Notice that in this example, the second index was not generated, therefore, the generated items form a subset of the possible generation of the given $^{<0\text{-}4>}$ range.	

We selected Subject-W's project mainly because she constantly changes directions in her development. It gives us the opportunity to illustrate how the ICE notation represents changes in design exploration paths. In Subject-W's project, the main design constructs that are repeated throughout the drawing sequence are the rooms, the dorm units, the entrances and common spaces. We will use this same relatively high level of abstraction as the main elements encoded in the ICE notation. We chose not to express further details because (i) details have not been resolved in most of Subject-W's drawings and (ii) this level of abstraction is adequate to describe the fairly complex ICE notation.

Let's begin by reviewing Subject-W drawing in table 1. She describes her housing hierarchy of rooms, units, unit-clusters, wings, buildings, building-clusters by a seemingly unstructured swirling drawing. Nevertheless, we identified relationships between the subparts of this drawing and we describe it using the following ICE regulators: The rotation $\Delta\mathbf{R}$, the curve $\Delta\mathbf{C}$ and the dilation (scale) $\Delta\mathbf{D}$. Assuming that this drawing represents a building, and its individual flower-like objects represent dorm units, and the central form represents the common spaces, we describe it in a top down manner, as follows:

$$\text{building} = \Delta\mathbf{C}_3(\text{dormCluster}) \wedge \text{commonSpace}$$

$$\text{dormCluster} = \Delta\mathbf{RD}(\text{flowerShape})$$

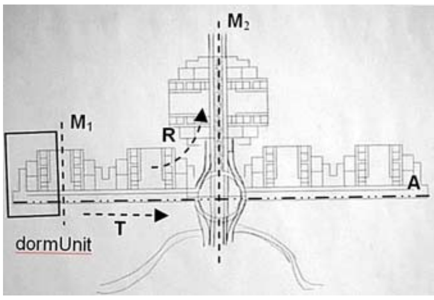
$$\text{flowerShape} = \Delta\mathbf{RD}(\text{room})$$

$$\text{room} = \Delta\mathbf{C}_\mathbf{1}(\text{p}_1)$$

$$\text{commonSpace} = \Delta\mathbf{C}_\mathbf{2}(\text{p}_2)$$

In the next configuration, table 4, the form of the dorm units is well defined. To derive the next configuration from the previous one, it is necessary to replace form of the dormCluster, and to replace the curve regulator $\Delta\mathbf{C}_\mathbf{3}$ by the sequence of regulators $\Delta\mathbf{M}(\Delta\mathbf{T}(), \Delta\mathbf{R}())$

TABLE 4. Tuesday, June 4, 2002

<table>
<tr>
<td>Subject-W is presenting a rectilinear scheme in which the modular bays of the dormitory scheme are being clustered to create a large and integrated form on the site which faces a long public edge of the campus proper as well as the service façade of the student activities building. This creates a "beads-on-a-string" type scheme.</td>
<td>
</td>
</tr>
<tr>
<td colspan="2">

$$\text{dormCluster}_1 = \Delta\mathbf{M}_\mathbf{1}(\text{dormUnit}_1)$$

$$\text{building} = \Delta\mathbf{M}_\mathbf{2}(\Delta\mathbf{T}(\text{dormCluster}_1), \Delta\mathbf{R}(\text{dormCluster}_2)) \wedge$$

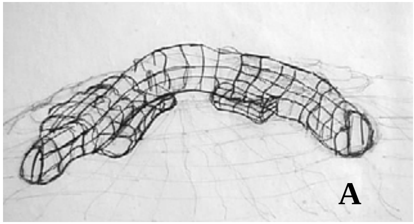
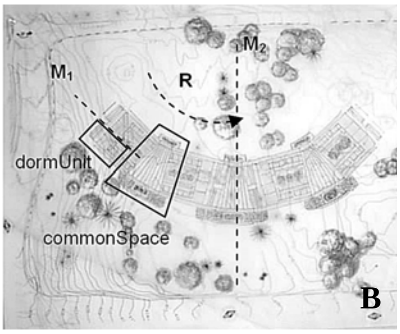
$$\Phi\mathbf{A}(\text{dormCluster}_1\text{-dormCluster}_4)$$

$$\text{Entrance} = \Delta\mathbf{M}_\mathbf{2}(\Delta\mathbf{C}(\text{p}))$$

The building is defined by (1) reflecting the dorm unit to form the dorm cluster, (2) translating and rotating the dorm cluster (3) reflecting the results of the previous generations and (4) aligning (regulator **A**) the dorm clusters (1 to 4). The entrance is defined by sweeping points along a curve then reflecting the curved lines.</td>
</tr>
</table>

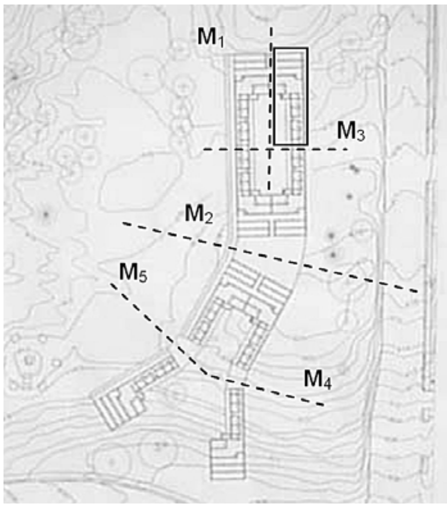
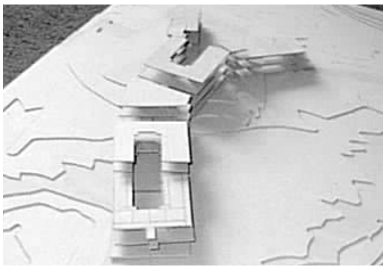
The next configuration (figure B in table 5) goes back to a curvilinear theme. The main reflection axis $\Delta\mathbf{M_2}$ is maintained. To obtain the curved axis from the previous regulator sequence $\Delta\mathbf{M_2}(\Delta\mathbf{T}(), \Delta\mathbf{R}())$, $\Delta\mathbf{T}$ is deleted and $\Delta\mathbf{R}$'s rotation degree is adjusted, leading to $\Delta\mathbf{M_2}(\Delta\mathbf{R}())$. The horizontal alignment is replaced by an implicit curvilinear alignment along the curve. Within the dorm cluster, the dorm units are repositioned and re-oriented. Their reflection $\Delta\mathbf{M}_1$ axis is rotated and the common spaces become more defined.

TABLE 5. Thursday, June 6, 2002

<table>
<tr><td>The beads-on-a-string type arrangement has yielded to a “serpentine” form that curves with the contours, creating a concave edge for the public and a convex one for the private side of the site lot. Intuition seems to guide the form.</td><td></td></tr>
<tr><td colspan="2">$\text{building} = \text{serpentine} \wedge \text{commonSpace}_1 \wedge \text{commonSpace}_2$
$\text{serpentine} = \Delta\mathbf{C_2}(\Delta\mathbf{C_1}(p_1))$
$\text{commonSpace}_1 = \Delta\mathbf{C_4}(\Delta\mathbf{C_3}(p_2))$
The serpentine building form is generated by sweeping a point along a curve, then sweeping the curve along another curve. Both common spaces are generated in the same way.</td></tr>
<tr><td>Here the serpentine form is refined. Curves turn into rotations and the central axis of symmetry from the previous is reinstituted.</td><td rowspan="2"></td></tr>
<tr><td>$\text{dormCluster}_1 = \Delta\mathbf{M_1}(\text{dormUnit}_1)$
$\text{building} = \Delta\mathbf{M_2}(\Delta\mathbf{R}(\text{dormCluster}_1)) \wedge$
$\Delta\mathbf{R}(\text{commonSpace})$
The building is generated rotating the dorm unit then reflecting it, and by rotating the common spaces (trapezoidal forms)</td></tr>
</table>

In the next configuration (table 6) the curve is broken into segments, Reflection is the dominant relationship. The central axis is maintained, but slightly rotated. The $\Delta\mathbf{M_2}(\Delta\mathbf{R}())$ sequence of regulators is replaced by $\Delta\mathbf{M_2}(\Delta\mathbf{M}_3())$.

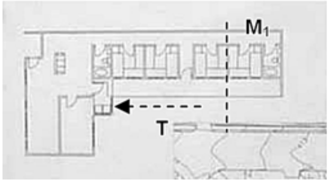
TABLE 6. Wednesday, June 12

<table>
<tr><td>The next formal overhaul involves one end of the “serpentine” form bifurcating into two wings, allowing the development of a “commons” area and lobby from one of the major access edges of the site. This remains the principal parti for Subject-W’s solution.</td><td rowspan="2">
</td></tr>
<tr><td></td></tr>
<tr><td colspan="2">3-D description
$dormCluster_1 = \Delta\mathbf{M_1}(\Delta\mathbf{TD}(dormUnit_1))$
$building = \Delta\mathbf{M_2}(\Delta\mathbf{M}_3(dormCluster_1)_{\#1}) \wedge$
$\Delta\mathbf{M_4}(dormUnit_4) \wedge \Delta\mathbf{M_5}(dormUnit_5) \wedge \Delta\mathbf{TD}(commonSpace)$
The building is generated by reflecting the dorm cluster twice, then reflecting the individual dormUnits to achieve the bifurcations. The 3D configuration is an extrusion of the 2D configuration with a small scaling factor. The $\Delta\mathbf{TD}$ regulator copies the floor slab vertically and scales them. The subscript indicates that the reflection $\Delta\mathbf{M_2}$ only reflects the last element of $\Delta\mathbf{M_3}$ not all the dorm units.</td></tr>
</table>

The following two configurations (table 7) are the variations suggested during the midterm review. Variation A is achieved by rotating the dorm cluster 180 degrees and converting $\Delta\mathbf{M_6}$ and $\Delta\mathbf{M_5}$ into rotations; while variation B is achieved by inserting another dorm cluster in the configuration, which is carried out in the notation by making $\Delta\mathbf{M_2}$ mirror both dorm units generated by $\Delta\mathbf{M_3}$.

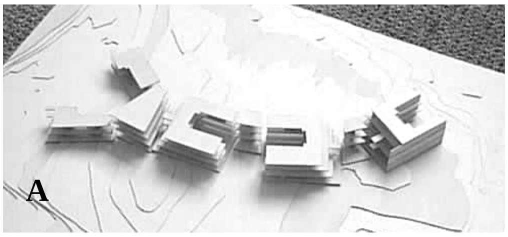
TABLE 7. Monday, June 17, 2002, Midterm evaluation of Subject-W's work

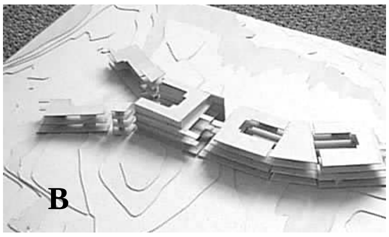
During midterm review, Subject-W's work shows little development over the previous critic. The most significant development is the cross axis that marks the secondary entrance, along the long side of the building.



$$\text{dormUnit}_1 = \Phi\mathbf{H}\,(\Delta\mathbf{T}(\Delta\mathbf{M}(\text{room})), \text{kitchen}, \text{bathroom}, \text{balcony})$$

The containment regulator, $\Phi\mathbf{H}$, indicates that the dorm unit consists of (a translation and a reflection of the room) as well as a bathroom, a kitchen and a balcony. A containment relation imposes restrictions on constituents, such as a transformation of the container will propagate the constituents, etc.





Variations suggested by the professor

3-**D** description (of variation A)

$$\text{dormCluster}_1 = \Delta\mathbf{M_1}(\Delta\mathbf{TD}(\text{dormUnit}_1))$$
$$\text{building} = \Delta\mathbf{M_2}(\Delta\mathbf{M}_3(\text{dormCluster}_1)_{\#1}) \wedge \Delta\mathbf{R_4}(\text{dormUnit}_4) \wedge \Delta\mathbf{R_5}(\text{dormUnit}_5) \wedge \text{commonSpace}$$

3-**D** description (of variation B)

$$\text{dormCluster}_1 = \Delta\mathbf{M_1}(\Delta\mathbf{TD}(\text{dormUnit}_1))$$
$$\text{building} = \Delta\mathbf{M_2}(\Delta\mathbf{M}_3(\text{dormCluster}_1)) \wedge \Delta\mathbf{M_4}(\text{dormUnit}_4) \wedge \Delta\mathbf{M_5}(\text{dormUnit}_5) \wedge \text{commonSpace}$$

The configuration in table 8 shows a return to the curvilinear axis and the rotation of the dorm clusters. To achieve this configuration from the midterm configuration (table 6), the bifurcation mirrors $\Delta\mathbf{M_4}$ and $\Delta\mathbf{M_5}$ are removed and $\Delta\mathbf{M_3}$ is replaced by a rotation $\Delta\mathbf{R}$. Although, this is speculation, it appears that Subject-W has created this configuration not by developing the midterm solution, but by working from the drawings in (table 5), while pair-wise integrating the common spaces from the midterm's configuration.

The configuration in table 9 is a further development of the previous one, focusing on the redefining the lower section. Axial symmetry is still maintained.

TABLE 8. Friday, June 21, 2002

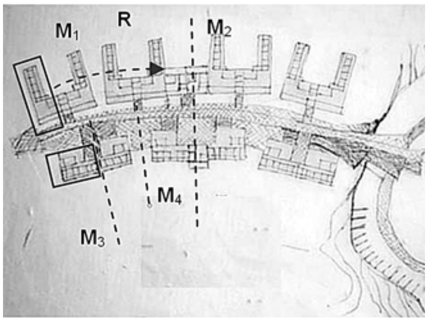
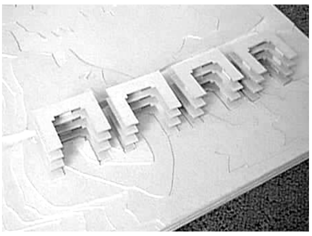
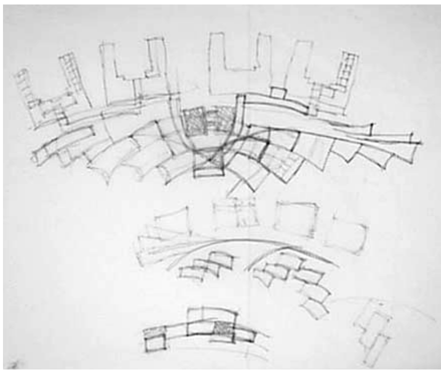
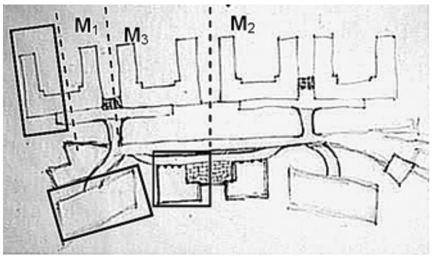
A new aspect of the scheme emerges. Precedent exploration based on Sant'Elias' work pushed the scheme towards "futuristic" features. The result however is not promising since the new synthesis appears to have a cartoonish resemblance to architecture. Drawings lack architectonic qualities, such as material, construction and structural specificity.	
$\text{dormCluster}_1 = \Delta\mathbf{M_1}(\text{dormUnit}_1)$ $\text{dormCluster}_2 = \Delta\mathbf{M_4}(\Delta\mathbf{M_3}(\text{dormUnit}_2)_{\#1})$ $\text{building} = \Delta\mathbf{M_2}(\Delta\mathbf{R}(\text{dormCluster}_1), \text{dormCluster}_2, \text{commonSpace})$	

TABLE 9. Monday, June 24, 2002

 This is a mixed bag. While the dorm units gain architectonic clarity, the main entrance, circulation and commons areas continue to resemble "spaghetti".	
$\text{dormCluster}_1 = \Delta\mathbf{M_1}(\text{dormUnit}_1)$ $\text{building} = \Delta\mathbf{M_2}(\Delta\mathbf{R}(\text{dormCluster}_1), \text{dormCluster}_2, \text{spagettiSpace})$	

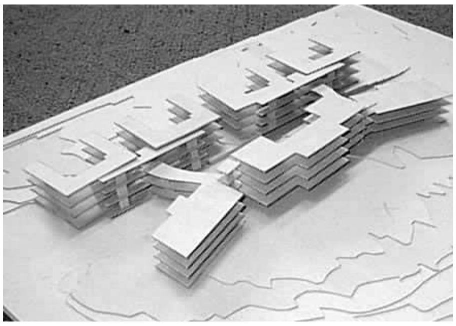
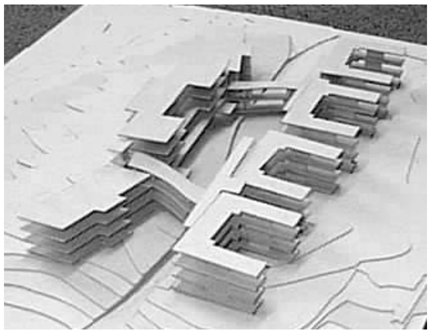
The general structure of the next configuration (table 10) is identical to the previous configuration, but the detail of the units for the lower dorm cluster is changing. The rotation $\Delta\mathbf{R}$ is replaced by a reflection $\Delta\mathbf{M_3}$.

TABLE 10. Wednesday June 26, 2002

This submission continues along the same lines as before. The “spaghetti” scheme dominates the formal development. Circulation paths are configured as tubes that go from point A to point B, without circulation and social hubs worked into the fabric. The scheme appears to be an inside-out path diagram, not architecture.	
$\text{dormCluster}_1 = \Delta\mathbf{M_1}(\text{dormUnit}_1)$ $\text{building} = \Delta\mathbf{M_2}(\Delta\mathbf{M_3}(\text{dormCluster}_1), \text{dormCluster}_2,$ $\text{dormCluster3}, \text{commonSpace})$	

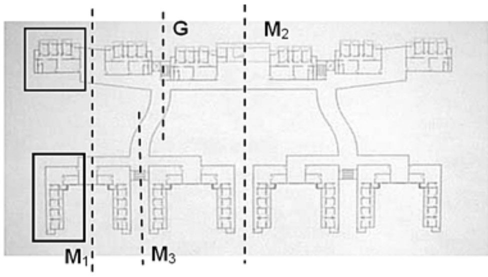
The next configuration (table 11) is a further development of the previous one, but maintains the same..

TABLE 11. Monday July 1, 2002

	
This marks a significant return to architecture and architectonics. The “spaghetti” is gone, dissolved in the interstitial space between two parallel dorm wings. The challenge from here on will be to establish the dialogue between these two spines and the cladding of the open space between them.	
$\text{dormCluster}_1 = \Delta\mathbf{M_1}(\text{dormUnit}_1)$ $\text{building} = \Delta\mathbf{M_2}(\Delta\mathbf{M_3}(\text{dormCluster}_1), \text{commonSpace}) \wedge$ $\Delta\mathbf{M_2}(\text{dormCluster}_2, \text{circulation})$	

The next configuration (table 12) is a slight development in the dorm cluster, where a glide relationship, depicted by $\Delta\mathbf{G}$, is explored.

TABLE 12. Wednesday, July 3, 2002

<table>
<tr><td>This brings issues of complexity vs. simplicity to the table. The scheme does a few things well. Other building systems solutions can be layered on top of it, which remains a challenge for Subject-W.</td><td>
</td></tr>
<tr><td colspan="2">

$$\text{dormCluster}_1 = \Delta\mathbf{M_1}(\text{dormUnit}_1)$$

$$\text{building} = \Delta\mathbf{M_2}(\Delta\mathbf{M_3}(\text{dormCluster}_1), \text{commonSpace}) \wedge$$

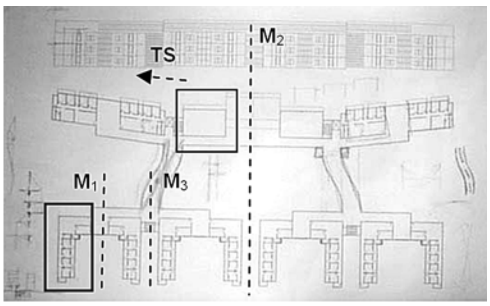
$$\Delta\mathbf{M_2}(\Delta\mathbf{G}\ (\Delta\mathbf{M_1}(\text{dormCluster}_2))_{\#1}) \wedge$$

$$\Delta\mathbf{M_2}(\text{circulation})$$

</td></tr>
</table>

The final configuration still shows some exploration in the dorm clusters. This time the units are slightly sheared. The reflection and glide regulator sequence $\Delta\mathbf{G}\ (\Delta\mathbf{M_1}(\,))$ is replaced by a composition of translation and shear $\Delta\mathbf{TS}$ regulators.

TABLE 13. Tuesday, July 10, 2002

<table>
<tr><td>Two days prior to the final review there are still basic issues of development and resolution. The final review does not bring any surprises or further development of the scheme.</td><td>
</td></tr>
<tr><td colspan="2">

$$\text{dormCluster}_1 = \Delta\mathbf{M_1}(\text{dormUnit}_1)$$

$$\text{building} = \Delta\mathbf{M_2}(\Delta\mathbf{M_3}(\text{dormCluster}_1), \text{commonSpace}) \wedge$$

$$\Delta\mathbf{M_2}(\Delta\mathbf{TS}\ (\text{dormCluster}_2))_{\#1} \wedge$$

$$\Delta\mathbf{M_2}(\text{circulation})$$

</td></tr>
</table>

4. Implications of the Notation

The ICE notation is designed parallel to a generative/manipulation system. Every regulator described in this paper, has been implemented or is currently being implemented. In the ICE system, every parameter of regulators can be manipulated, thus generating highly flexible models. Furthermore, regulators can be inserted, deleted, or replaced to accommodate redefinition of the notation string, and consequently, the redefinition of the configuration's structure.

For this paper, we have described the subset of the ICE notation that is relevant to Subject W's design sequence. However, the ICE notation has numerous other features, which we will briefly review in this section.

It is important to note that ICE is not the only notation in its class. Leyton (2001) developed a generative theory of shape, which uses the same principles of mathematics as ICE. Leyton's work focused on the mathematical theory of shape generation, while the ICE system/notation focuses on the practical aspects of implementation and usability, the most important of which the ability to manipulate the configurations generated by ICE. Leyton also addressed the issue of process-capture, which he refers to as recoverability; an issue that will be revisited it in the following section. Cha and Gero (2001) have developed a shape schema representation, based on Isometry transformations and used it to describe numerous buildings of notable architects. In addition to being part of a computational system, the ICE notation extends the aforementioned representations in the following ways:

1. ICE is designed to work in 3-dimensions. All parameters and operations in ICE are based on 3-dimentional geometry principles.
2. The regulator construct subsumes generative transformations, and encode other functions such as constraints (for instance alignment, boundaries, proportions, containments), operations (for instance subdivision and Boolean operations) and variations (such as rhythm, gradation and differential sweeping).
3. The ICE notation's support for sub-part generation is a unique feature of ICE which magnifies the possibilities for shape generation and shape manipulation.
4. The support for different levels of information greatly simplifies the representation. The short and long forms allow the ICE notation string to be viewed at two crucial levels of abstraction (relational level, and parameterization level). The shape encapsulation feature in the ICE notation helps structure the string and avoids redundancy in descriptions.

Another distinction of our approach is that we are using a formal notation to represent an evolving design. Most attempts to formalize design representations have either encoded completed architectural design, or encoded hypothetical designs. Our paper, on the other hand, has addressed a more challenging task, a constantly changing design, which may be imperfect and often incomplete, but nevertheless illustrates a natural progress of a student at work. With the ICE notation, we codified each drawing in the development sequence as well as each transformation from one drawing to the next, thus demonstrating that the ICE notation can follow a student's exploration path.

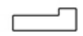
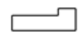
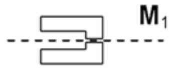
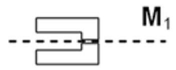
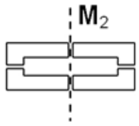
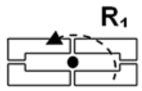
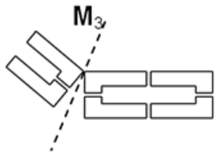
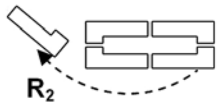
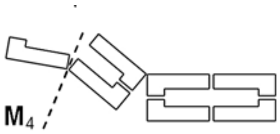
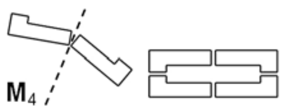
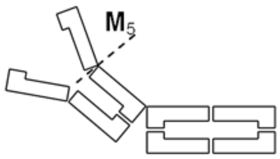
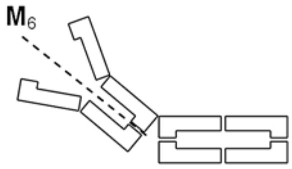
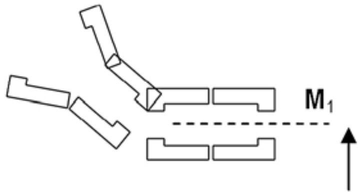
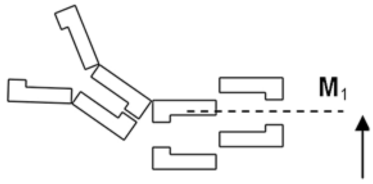
5. Implications for Process and Cognitive Analysis

In the area of cognitive models of the design process one of the difficult challenges is to formally measure and compare intermediary states in a design state space and draw generalizations about human design behavior (Akin 1996). Purcell *et.al.* (1994) have made progress in this direction. They have devised ways of unambiguously codifying and characterizing individual design activities throughout design protocols. Their codification relies on interpretations by human coders of the data and like most qualitative analysis methods on entities that emerge from the data.

The contributions of this paper are as follows: (1) we use a notation that is formal and unambiguous, (2) we use an a priori notation to code the data that is not derived from the data, and (3) we use a graphic representation that is capable of encoding both graphic entities and process transformations.

Formality of the ICE notation enables us to show quantifiable differences in the information content of both the design state representations and their transformations. The ICE notation supports of multiple descriptions for the same configuration. Therefore, it captures, for each description, different processes for generation, different applicable transformations, and consequently, different manipulation handles. For example, consider the graphic sequences in table 14. This is subject W's midterm submission (see table 6) as it is generated in an early version of the ICE system using two distinct generation paths, and consequently yielding different ICE notation strings. In steps 1 and 2, a dorm unit is created then reflected about $\Delta\mathbf{M}_1$. In step 3, the same arrangement is obtained (step 3A) by a reflection about $\Delta\mathbf{M}_2$, and in (step 3B) a rotation about $\Delta\mathbf{R}_1$. The generation sequence continues in distinct paths though steps 4 and 5, yielding different arrangements. In step 6 however, two different actions, reflecting about $\Delta\mathbf{M}_5$ and reflecting about $\Delta\mathbf{M}_6$, bring the arrangement back to equivalence. At this point the two shapes are identical, but their notation is not since the notation also captures the way in which each shape was generated. We can formally show this in the notation and unambiguously express in the final shape the difference(s) between the two generative sequences (see table 15).

TABLE 14. Generative sequence for subject W's midterm submission

	A	B
1		
2		
3		
4		
5		
6		
	$\Delta\mathbf{M}_3(\Delta\mathbf{M}_2(\Delta\mathbf{M}_1(\text{dormUnit}_1)))$ $\Delta\mathbf{M}_4(\text{dormUnit}_4)$ $\Delta\mathbf{M}_5(\text{dormUnit}_5)$	$\Delta\mathbf{R}_1(\Delta\mathbf{M}_1(\text{dormUnit}_1))$ $\Delta\mathbf{M}_6(\Delta\mathbf{M}_4(\Delta\mathbf{R}_2(\text{dormUnit}_1)))$
7		

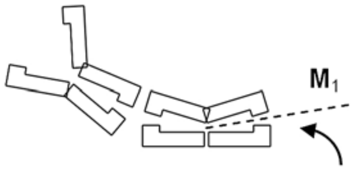
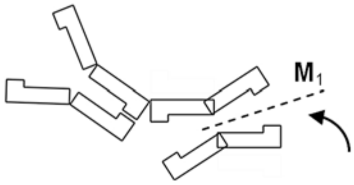
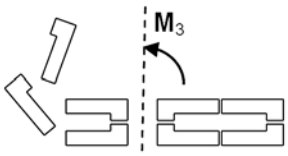
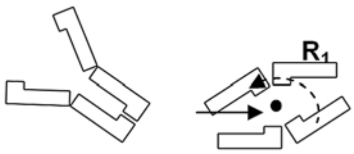
	Move mirror $\Delta \mathbf{M}_1$ upward	Move mirror $\Delta \mathbf{M}_1$ upward
9		
	Rotate mirror $\Delta \mathbf{M}_1$ counterclockwise	Rotate mirror $\Delta \mathbf{M}_1$ counterclockwise
		
	Rotate mirror $\Delta \mathbf{M}_3$ counterclockwise	Move rotation point $\Delta \mathbf{R}$ to the right

TABLE 15. Graphic, generative and manipulation equivalencies between sequences A and B of table 14

Steps	**Graphics Information**	**Generative Information**	**Manipulation information**
1	equivalent	equivalent	
2	equivalent	equivalent	
3	equivalent	NOT equivalent	
4	NOT equivalent	NOT equivalent	
5	NOT equivalent	NOT equivalent	
6	equivalent	NOT equivalent	
7	NOT equivalent	NOT equivalent	equivalent
8	NOT equivalent	NOT equivalent	equivalent
9	NOT equivalent	NOT equivalent	NOT equivalent

6. Implications for Intent Capture and Cognition

This capability in the ICE notation allows us to not only encode graphic and generative design sequences but also to “replay” them in the way the graphic entities were generated in the first place. This has advantages in assisting designers to visualize the genesis of a form. This can be helpful in encoding not only design histories but also the design intent. The subcomponents that make up a graphic element can help retrieve the functional requirements that go into the final form.

Secondly, we can formally and quantitatively measure the information content of each state in the state space of design representations. We can go beyond surface similarities of graphically equivalent entities and measure the steps and stages that went into creating each one. This can be used to quantify the information content of graphic designs. One goal for doing this would be to determine more parsimonious ways of producing forms. Another purpose which is orthogonal to the first is to be able to embed handles (or structure) into shapes for further manipulation.

For example the resulting form (step6) in Sequence B of table 14 has different handles than the same one in Sequence A. This has two significant results, which are illustrated in steps 7-9 of table 14 (1) Identical manipulation-actions (for instance moving shared regulators) would result in totally different graphic configurations, step 7 and 8. (2) The different handles (non-shared regulators) allow for a different set of manipulations per graphic configuration, such as the ones shown in step 9 (moving the rotation point $\Delta\mathbf{R_1}$ or rotating the mirror line $\Delta\mathbf{M_3}$.

There are numerous possible manipulations for each sequence; those shown were just a few. Additionally, redefining the notation string by insertion, deletion, or replacement would expand the manipulation possibilities even further and redirect the exploration paths.

These types of interactions within the ICE system suggest debatable questions regarding cognition. Suppose Subject-W had a system such as ICE, would she have followed the same exploration paths as she did in the annotated studio? Would she have explored other paths and came up with different configurations? Would she have completed the exploration faster, thus giving her more time to develop details further, or would she have done many more explorations, thus sidetracking from developing the focused completed design? The more general question is whether such capabilities offer a relief in cognitive loads, or places an additional burden on the designer of understanding the structure handles and their manipulations.

Furthermore, we can codify all graphic entities by surface structure and generative structure. We believe that this has important consequences for codifying and analyzing cognitive representations of designs. Our future work in this direction will be to codify protocol (not ethnographic) data to exploit this possibility. We will also compare our approach to others in the field such as those of Purcell, *et.al*, (1994) and Suwa, *et.al.* (1998).

7. Potential Application Areas

The discussion provided in the above section explores the functionalities that ICE affords (or would be able to afford) us, more or less independent of the specific design applications or even specific design problems. Yet, there remain questions about how these functionalities could impact the world of computational applications. What are the potential benefits of having multiple representations of the genesis of designs? How can these descriptive techniques be helpful in prescriptive strategies in design? Are these capabilities best used in *post facto* or generative descriptions of designs?

The fact that ICE specifies *multiple ways* of creating the identical graphic entity affords us many distinct representations for each entity. We can form a rectangle by sweeping a point into a line and the line into a surface. Alternatively, we can create the same rectangle by mirroring exactly half of it along a symmetry axis. Multiple representations enable us to capture precisely the manner in which an entity is created as well as what it is. This leads to interesting design application opportunities. Can we capture different ways of making shapes that are preferred by different users? Do these correspond to drawing performance measures such as: faster, easier, consistent with the geometry of the form, and so on? While, currently, we do not have sufficient data to answer these questions, they present interesting future research avenues. Such investigations may lead to generic and customizable approaches to making graphic elements for design.

Another important goal of this approach is to develop tools that go beyond the *descriptive* accounts of design processes and assist in *prescriptive* design strategies. One of the ways this can be accomplished is through design libraries. A collection of ICE notations can constitute a library of design elements that can be used to create new design assemblies. As with a case base, this library can be used to adapt past designs and design elements to new problems. The difficulty with most case- or library-based system is the excessive overhead of populating the library or the case-base with design instances. The effort needed to build libraries is so large that most of them have impoverished instance sets. One way of overcoming this problem is to capture cases during design. ICE is perfectly suitable for this approach. It's process capture functionality can be adapted to an interactive format for the designer to store away, on the fly, instances as they design them.

Furthermore, the ICE notation can become the basis of a tool to capture *design history*. As in most complex design environments the history of designs (i.e., why some things are configured the way they are) is the most difficult requirement to satisfy with available representations. Drawings capture the “what,” and the specifications capture the “how” of building designs. There are no representation systems designed to deal with the “why.” We believe ICE is a natural to fill this gap. With ICE one can play back the sequence of entities created, down to the last line or point of a graphic entity, however complex they may be. We believe this will become the armature, much more effectively than any static design representation, to capture the history of design entities; to provide information about the formal genesis of each design component; to enable the modification of the design without losing information about its history; and to augment this history while preserving the design.

Finally, we believe all of these functionalities are essential for building CAD applications that capture formal design *intent* effectively and efficiently. Our future research will address some if not all of these issues.

7. References:

Akin, O: 2002 *Design: the Art and Science of Synthesis ©,* unpublished manuscript, School of Architecture, Carnegie Mellon University, Pittsburgh, PA 15123, USA

Akin, O and C T Lin: 1996 “Design protocol data and novel design decisions” in *Analysing Design Activity*, edited by N. Cross and K. Dorst, John Wiley and Sons, Chichester, West Sussex, 1996, pp. 35-64.

Cha, M, Gero, J, 2004, Shape Pattern Representation for Design Computation, *(http://www.arch.usyd.edu.au/%7Ejohn/publications/progress.html)*

Leyton, M: 2001, A Generative Theory of Shape. *Springer Verlag, New York*.

Moustapha, H: forthcoming, A Formal Representation for Generation and Transformation in Design, *GCAD’04 Symposium Proceedings, Carnegie Mellon University, Pittsburgh...*

Moustapha, H, Krishnamurti, R: 2001, Arabic Calligraphy: A Computational Exploration, *Mathematics and Design 2001, Third International Conference, Deakin University, Geelong.*

Purcell T, J. Gero, H. Edwards, T. McNeill: 1994 “The data in design protocols: the issue of data coding, data analysis in the development of models of the design process,” in J. S. Gero and F. Sudweeks (eds.) Artificial Intelligence in Design’94, Kluwer, pp. 225-252.

Suwa, M, T. Purcell, and J. S. Gero: 1998 “Macroscopic analysis of design process based on a scheme for coding designers’ cognitive actions,” *Design Studies* 19 (4): 455-483.