Algorithmic Modeling: Teaching Architecture in Digital Age

Nathan Howe

Kansas State University (K-State)

Can a working knowledge of algorithmic modeling augment student understanding of building architecture? This question is fundamental when addressing student design education today. This paper demonstrates that when students apply a reductive process more in line with Newell, Shaw and Simon (Newell, Shaw and Simon 1957), they can break down a complex problem into simpler and simpler terms until the problem can be resolved. This type of reduction can be applied systematically to the parametric-driven form through reverse engineering. In the process of reverse engineering, students begin to connect descriptive geometry with complex form, breaking down the complex form into its simplest parts. This design process of reduction and reverse engineering leads designers to take a more systematic approach to theoretical ideas, at once creating complex constructs while pragmatically attacking the issues of buildable form. This paper will delve into teaching analytical tools so students not only comprehend the input of form-making, but the necessary output to test building and material concepts. Fostering a clear methodology for testing built form within the design process also furthers the student's development as a problem solver and design innovator.

Introduction

First of all, we must explain that the person standing in front of them is not someone who asks questions whose answers he already knows. Practicing architecture is asking oneself questions, finding one's own answers with the help of the teacher, whittling down, finding solutions. Over and over again.

--Peter Zumthor, on how students should be introduced to architecture school (Müller 1998).

While it may seem rather perverse to use a quote by Zumthor to begin a paper about teaching algorithmic modeling, his notion of "whittling down" is a poetic description of reductive thinking, where each problem is resolved by simplifying the complexities. Using the mindset of Zumthor, scripting gives young designers the potential to use the computational realm found in algorithmic modeling to apply systematic rigor to their design, while continuing to pursue the poetic reality of architecture.

Scripting offers new potentials of exploring complexities of form before rendered impractical. Teaching students methods of algorithmic modeling at once furthers their tools as a designer and develops previously untapped design processes to deduce technical as well conceptual ideas. In learning the process of reducing the complex to its simplest form, the students gain valuable insight in how to work through design issues. This paper will use a series of seminar projects that use script logic to facilitate student thinking in terms of complexity, tectonics and other realms of architecture.

The Basics

With scripting, students learn to navigate a world that explicitly describes every move through clear logic of geometric translations. The student must consider point, vector, curve and surface. While some of these moves are clearly similar to CAD manipulations, scripting adds discipline. Scripting, because of its associative relationship to other components, requires the designer to create an associative framework for desired absolute and relative connections. This type of connectivity allows students to objectify their work to more clearly construct intelligence into their design. The intelligence will be further elaborated upon in the material analysis portion of the paper.

When initiating discussion of the fundamentals of scripting language, it is important for students to begin with their hands. Introducing simple plane manipulations through the folding of material, such as card stock paper, is a clear way to understand how simple physical moves create geometry. Using scripting, students then use simple transformations of points and curves to input the geometries of these translations into the computer (Figure 1.). As students engage in this project, it is important to create a loop of information input and the output of making. This process creates immediacy to how the scripting language translates real world coordinates and manipulations into physical outcomes. This closes the loop from physical to virtual back to physical.

The process quickly accelerates from one component being driven by simple variable input to a large, complex surface that is populated by variations of the component (Figure 2.). The final connection of the loop is to make the component aggregation a physical reality, which is where the reductive process and reverse engineering are applied (Figure 3.).

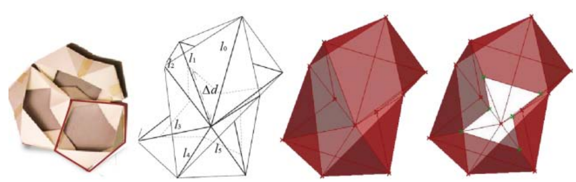

Figure 1. Component Aggregation, part 1. This assignment forms student understanding of input and output feedback loops. The student begins with a simple component created by hand and then designs a script to place the geometry into the computer.

Reductive Process

So the question becomes how to take the complexities of a design and reduce it down to a physical prototype? This is where a shift in thinking is necessary. In computer science, reductive thinking is often used to break down complexities. This reductive process defines the complex problem and reduces it to its simpler parts until each part can be dealt with given the knowledge at hand (Newell, Shaw and Simon 1957). When applied to architecture, algorithmic modeling challenges students to logically confront what they do not know about the complexity of a design problem.

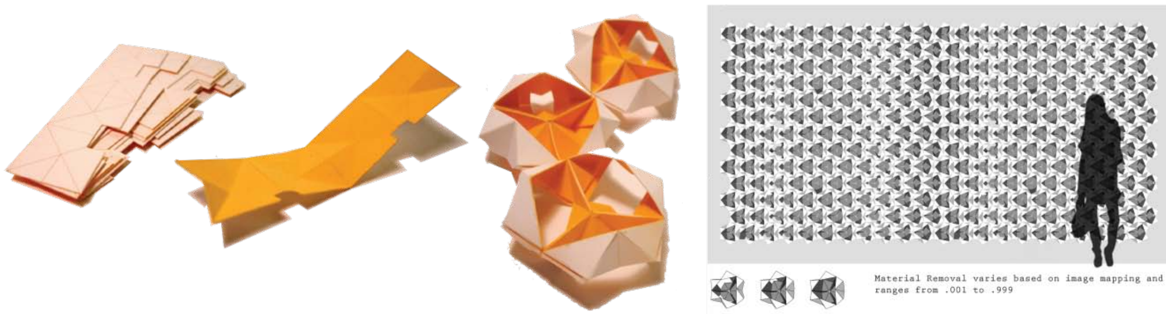

Figure 2. Component Aggregation, part 2. The component, when aggregated, then has sample data control variable(s), making each unique within the system. This new system of unique components must then be reverse engineered for final output. In this example, the component also has a field data control, in the form of an open and closed variable, to create a low resolution porosity study.

In this case, once students have aggregated their component across a complex surface, they are asked to reverse engineer their components for fabrication. This is a watershed moment for the students because most are completely overwhelmed at taking hundreds of unique pieces and developing part files. However, if this complexity is reduced, as demonstrated in the example project that follows, the student finds the information is at hand. They just need to clearly formulate it.

In Figure 1., the student's aggregation is a component that radiates around a point. Three components make a complete hexagonal pattern. The component has a variable *d* that is modified to simulate open and closed conditions. When placed together, these layers of complexity are daunting. However, by using the basic geometry the student used to create the component in the first place, the student can then break down the larger system into the necessary information. In this case, the important factors were each hexagonal curve's side size, l_n, along with the factor of *d* already discussed (Figure 3.). These two variables could be gathered from the information already at hand. The student then transfers this variable information to a horizontal plane given the variable's relationship to the original flattened geometry, as shown in Figure 2.

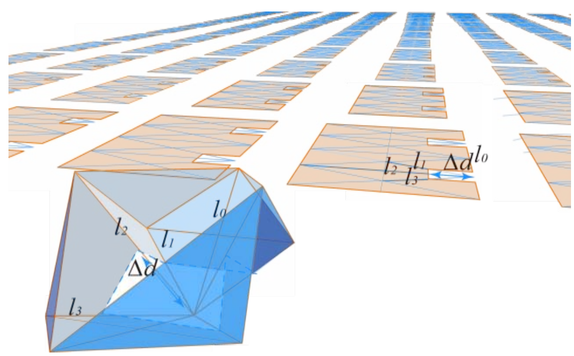

Figure 3. Component Aggregation, part 3. The final process is to reverse engineer the aggregation for fabrication.

With the major construct of the assignment laid out, it is necessary to analyze the process for its import in regards to algorithmic modeling and architectural design. If one believes the precept that physical models are necessary to design good architecture, then it is important to dwell on analogous modeling and fabrication in design processes. As parametric design becomes more prevalent, there is a propensity to create complexity. It is important to shift the student's attention to the execution of the design. Students can quickly design themselves into a corner, as the complex geometries drawn in the computer can quickly seem too complicated to physically construct, and students create buildings that can only be tested in the computer.

patterning within the material that would change its properties of form making.

Base test data was collected by testing the point of failure for chipboard when bent to tighter and tighter radii in both directions. The first outcome of this procedure was students discovered that what they thought was a mono-directional material actually had a difference in property, depending on which direction it was rolled.

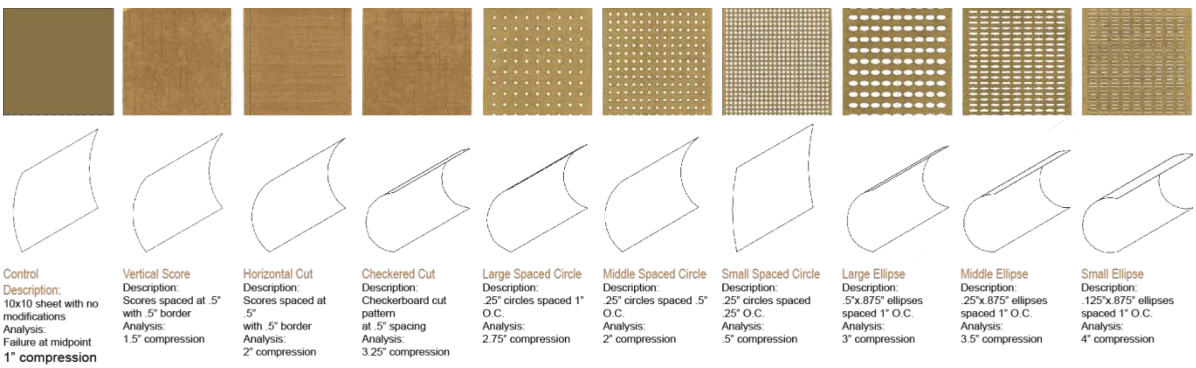

Figure 4. Material Analysis, part 1. This material intelligence project created analog data collected through a series of material testing of perforated and scored patterns. Each pattern was then tested for increases in material flexibility given these modifications.

Unfortunately, this condition often creates underdeveloped designs. The student may resort to sectioning or 3-d printing. However, the former often becomes a literal skeleton of the idea and not the true analog of the concept. The use of 3-d printing may show physical results, yet nothing in this realization delves into the process of building the idea. The tradition of model building has become a tool to show analogous process of the design idea becoming a physical reality. Therefore, the ideal is to continue to develop thought processes, such as reductive logic, that allow students to procedurally breakdown complex geometries to analogous fabrication techniques.

Material Analysis

One project developed to demonstrate the dialogue of trial and error and scientific reduction is a simple "material analysis" assignment. Students were asked to take a material, such as chipboard, and develop

As seen in Figure 4., students tested various patterns of scoring and perforating the chipboard. Each iteration was tested for failure, increasing the data set. Students then took this data and created topographical studies with complex form. While still holding to the given minimum radii, students extrapolated whether the material would naturally conform to prescribed constraints under conditions of valley and ridge (Figure 5.). The students measured the amount of curvature to further augment the radii data. From these tests, students took the most successful patterns and applied this information to complex surfaces that were predetermined by the data. Interestingly enough, the chipboard reacted best as a conforming material only under scored pattern conditions. Initially, the perforations seemed to have as much success in changing the properties of the material (Figure 4.) as scoring did. However, students found the perforations weakened the material to the point of failure when attempting to form complex curvatures.

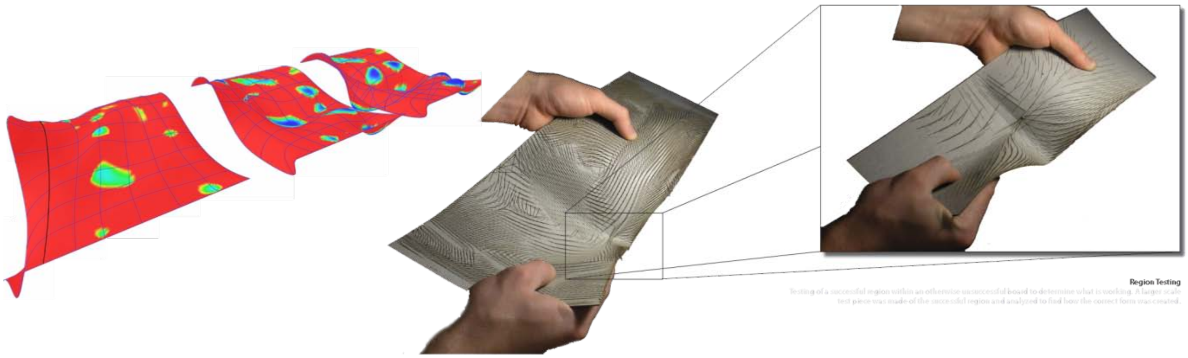

Figure 5. Material Analysis, part 2. The surface used for testing the patterning was analyzed for minimum radii curvature. This surface was iterated until the desired minimum curvature was achieved. Then various patterns were tested for natural deformation.

One might ask, why was chipboard used? Why not some other substrate that is more of an architecturally applicable material, along the lines of the wood studies by Achim Menges (Menges, 2009)? While a more architectural material is not out of the question, the most important outcome of this study was to teach methodologies for formulating base measures, collecting data, and then interpreting the data and formulating further hypotheses for testing. This type of scientific methodology creates a more rigorous process for students to evaluate not only their design work, but from which to crystallize a more clear process for experimenting with formal complexities.

These assignments desire to teach a more rigorous methodology through answerable problem solving. This type of thinking is quite foreign to architecture students, who are used to more subjective architectural measurements. Yet, this type of processing offers designers a whole new horizon of possibilities. The following section will outline a final independent seminar project that applied reductive processes and algorithmic rigor to the design. This project exemplified development of a methodology that harnessed the power of scripting associative logic to not only render an architectural idea, but more accurately and objectively evaluate the design.

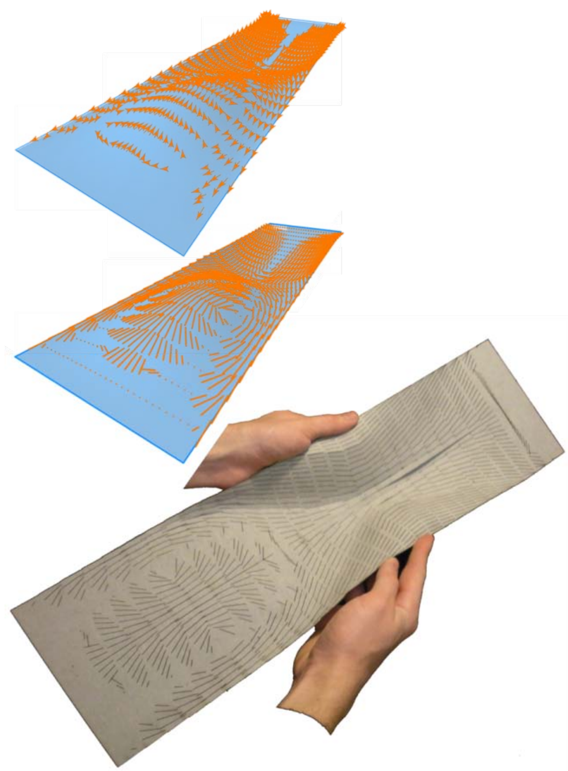

Figure 6. Material Analysis, part 3. With a multiplier controlling the various lengths of scores, and the maximum curvature vector providing direction, a rather convincing natural deformation resulted.

Elaboration

The project brief called for a design for a new school of architecture on the Kansas State University campus, and was driven by an extensive façade study of the main architecture studio bar. Graduate student Tim Meyers investigated two contradictory design motivations: a desire to create a glass façade, and a need to conform to the campus façade policy that requires the majority of material to be native limestone.

As seen in Figure 7., the façade was conceived as an intricate network of layers of limestone blocks. These layers both activated the façade and created diffused light within the studios, no matter which direction the studios faced. In the middle of the façade were horizontal shelves of steel structure. The spacing of the shelves was based on the altitude angle of winter solstice. On the southwestern façade, a vertical pattern of limestone veneer was developed. This layer was designed to continue creating diffuse light during the afternoon, when the studios would be in full operation. The interior layer was a more traditional limestone veneer that also functioned to attenuate sunlight to the ideal indirect lighting.

Meyers further experimented with various levels of porosity using Galapagos (an evolutionary component within Grasshopper). The major variables of position and location as per sun angles could be modified and allow this script to be applied to any region on any building. Meyers successfully developed a script that not only created specificity to its site with exacting sun angles, but also provided formal complexity around a rather rigid geometry. The outcome was at once delightful and pragmatic.

Conclusion

The profession must nurture the poetic and artistic character of the student, but must also foster analytical thinking of materiality and structural principles. In addition to the subjective design processes architecture students learn, development of a systematic problem solving methodology such as the reductive process is an important part of design education. As this last project demonstrates, the use of algorithmic processes delivers a more analytical type of thinking invaluable to architects. These thought processes enhance the education of the architecture student by allowing them to better understand how to decipher complex structures and ideas.

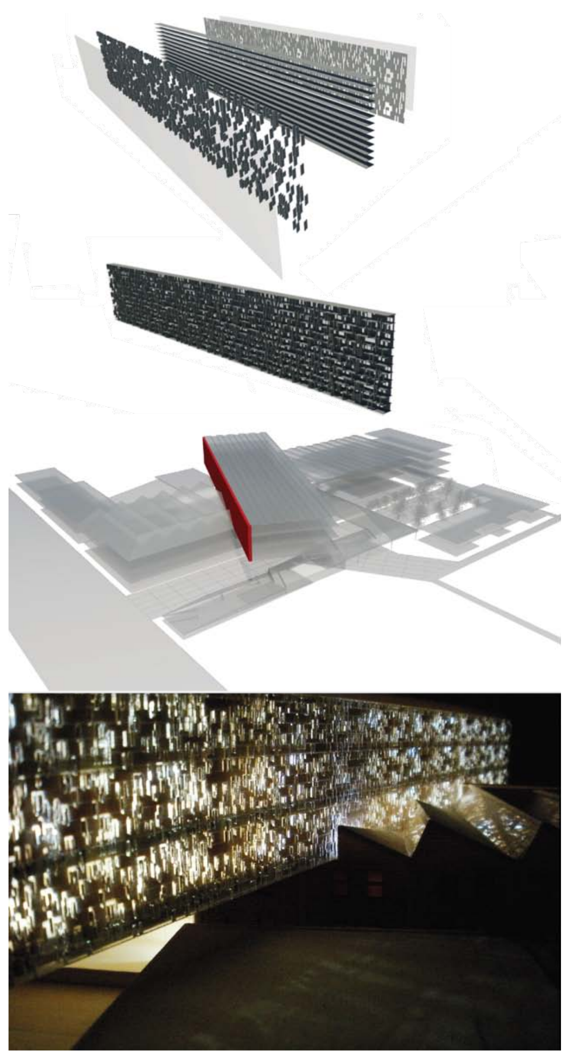

Figure 7. The various layers each received specific attributes based on sun angles and porosity to create a rich pattern with performative qualities.

References

Menges, A. “Performative Wood: Integral Computational Design for Timber Construction”. reform: Building a Better Tomorrow, Proceeding of the 29th Conference of the Association For Computer Aided Design In Architecture (ACADIA), Chicago, 2009, pp. 66-74.

Müller, L.. Peter Zumthor - Thinking Architecture. Lars Müller Publishers, 1998.

Newell, A. Shaw, J.C. Simon, H.A. Elements of a Theory of Human Problem Solving. Rand Corporation, 1957.