Adaptable Architecture
A Computational Exploration into Responsive Design Systems

by Alejandro Zulas

Arquitecto
Instituto Tecnológico y de Estudios Superiores de Monterrey
ITESM - Campus Monterrey, 1999

Submitted to the Department of Architecture
in Partial Fulfillment of the Requirements for the
Degree of Master of Science in Architecture Studies
at the Massachusetts Institute of Technology
June 2004

Signature of Author:

Department of Architecture
May 19, 2004

Certified by:

Terry W. Knight
Associate Dean and Associate
Professor of Design & Computation
Thesis Advisor

Accepted by:

Julian Beinart
Professor of Architecture
Chair, Committee on Graduate
Students

Readers:

.....

William Lyman Porter
Norman B. and Muriel Leventhal
Professor of Architecture and
Planning
Thesis Reader

.....

Stephen M. Ervin
Director of Computer Resources
Assistant Dean for Information
Technology
Harvard Design School
Thesis Reader

Adaptable Architecture
A Computational Exploration into Responsive Design Systems

by Alejandro Zulas

Submitted to the Department of Architecture
On May 19, 2004
in Partial Fulfillment of the Requirements for the
Degree of Master of Science in Architecture Studies

abstract:

Based on the fact that architecture is, among other things, the crystallization of a mediation among design intentions (function), meaning and contextual constraints (performance), we as designers are obligated to produce morphologically flexible & adaptive design solutions; both during the design process and as a final outcome.

In that sense this thesis is an open ended exploration of embedding rational adaptability to object design through computational tools.

This thesis will speculate on the advantages of thinking architecture in terms of "adaptation" in an action-reaction fashion, evolving from the seed idea of "motion" in architecture but rather pushing and exploring the potential of digitally designed responsive buildings and the dissection of its methodological approach. Empirically, it will look into some of nature's responsive designs, arguing that buildings can be conceptualized as adaptable living organisms. It will also analyze the role of computational tools and programming languages as meaningful mediums that help designers to better understand, set-up, define and re-define design problems. It will argue that more than an automated provider of an endless number of design solution computers can work as a systematic tool, making us more conscious during the design process.

Thesis Supervisor: Terry W. Knight
Supervisor Title: Associate Dean and Associate Professor of Design & Computation

acknowledgements:

I would like to thank:
My thesis supervisor, Terry Knight for her time, guidance and for encouraging me to always keep exploring.
My thesis readers, Bill Porter and Stephen Ervin for sharing their ideas and experiences.
My Design & Computation fellow students, for their support and constructive critiques. Especially to Axel Kilian, Victor Gane and Gonçalo Ducla Soares.

This work is dedicated:
To my parents Teresa and Piros for advising and supporting me unconditionally and for always believing in me.
To my sister Tere and my brother Piros for sharing with me their space and time and for their sincere and unreserved companionship.
To Karla Elena for being an endless source of inspiration.

table of contents:

*"...The problem posed in **Futurist architecture** is not one of linear rearrangement. It is not a question of finding new moldings and frames for windows and doors. **It is a question of tending the healthy growth of the Futuristic building** [house], of **constructing it with all the resources of technology and science, satisfying the demands** of our **habits** and our **spirit**. We have lost our predilection for the monumental, the heavy, the static and we have enriched our **sensibility** with the **taste for the light, the practical, the ephemeral and the swift**..."*

Antonio Sant'Elia, 1914.[1]

000. Introduction

I have always been intrigued by the fascinating idea of adaptability of living organisms as a very powerful medium through which they accomplish certain performances and therefore they survive under the given circumstances in which they exist in. Furthermore, to me, adaptability is not just a matter of survival but also a matter of surviving with the minimum necessary, in other words, it is also a matter of efficiency reaching.

In architecture one could imagine a number of different analogies between living beings and/or its organs and buildings and/or its components. Let's just think how the skin of a reptile is not only a hard protective shell but is also an adaptable changing interface between the outside world and the internal organs of the reptile (i.e. capturing energy through sun light absorption, color-camouflage, extreme temperature and weather conditions resistant, mobility-flexibility, etc.)

0.01 Embedded responsive systems in living organisms.
Reptiles' skin and sunflowers are examples of adaptable-responsive systems present in the natural world.

[1] Apollonio, Umbro; ed. (2001). *Futurists Manifestos*. Boston: MFA Publications.

Motivated by these issues I have asked myself the question of how one could imagine an architectural system (i.e. structural, outer envelope, etc.) that could be adaptive in a similar way that the skin of a reptile or the sun-following behavior of the sunflower are adaptive to its surrounding conditions.

In recent years and during my time at MIT I have worked around the concepts of collapsible structures and kinetic systems and its potential applications in architecture. After some digital and physical explorations I have realized that the use of kinetic systems is a very powerful way of conceiving adaptable architectural elements or components.

0.02 Kinetic Systems.
Examples of collapsible structures using conventional mechanical kinetic systems developed through coursework at MIT.
Explored concepts: scissor effect, folding, hinging and sliding.

Although the application of these conventional kinetic systems into architecture is powerful, these systems are yet somewhat limiting due to their formal conception. Conventional kinetic systems are, in most cases, restricted by the morphological configuration of the mechanical space frame in which their components are embedded (i.e. scissor effect, folding, hinging, sliding, etc.) and at the same time these systems confront material restrictions and gravity forces of our physical world. Although these systems are adaptive, they only respond to the very specific objective of: space-saving, reaching compactness and deploy-ability from a relation between adjustable elements and the relocation of members between connecting nodes and hinges.

The same understanding that made me aware of such limitations was the motivation that pushed me to confront the problem with different eyes. I decided to focus on the exploration

of the concept of "adaptation" viewed as flexible design system. In order to develop my research I took the computer as the main medium through which I could carry on my design experiments taking programming as the "common" language (control mechanism) through which I could express and translate my ideas into a virtual context.

Based on that, this thesis is an exploratory research on alternative ways that synthetic universes would be designed, conceiving them as platforms through which it is possible to conceive, simulate and model adaptable objects responding to extrinsic and / or intrinsic forces, hidden or obvious to the eye.

Questions

After being exposed to my previous explorations, understanding and analysis of such a particular problem, made me raise the next set of questions[2] which helped me define and tailor the line of research conducted through this thesis:

- Within the design process, could objects (buildings) be conceived as self adapting 'smart' entities, responding to a particular set of conditions?

- Is it possible to set up a computational system in which specific contextual conditions, design intentions and meaning could be mapped and mediated?

- How restrictive or flexible would be a rule-based design approach? Would mediation really be achievable?

- Could such a system be able to allocate adaptability in terms of design? Would this allow the generation of several design options?

[2] It is worth mentioning that this never was a final set of questions but rather a 'mutable' set of personal concerns that were changing along this research.

- Is it possible to set up a computational system in which is feasible to overlap several layers of interlinked adaptive systems? Embedding them into others?

Significance of the problem

Based on the fact that architecture is the crystallization (partially) of a mediation among design intentions (function), meaning and contextual constraints (performance), we as designers are obligated to produce morphologically flexible and adaptive design solutions; both during the design process and as a final outcome.

In that sense this thesis is an open ended exploration of embedding rational adaptability to object design through computational tools.

The importance of this research work dwells in its conceptual level. This work attempts to frame my personal vision of what could be the next generation of buildings, the new "adaptable" architecture. My hope for this work is that it becomes a seductive seed of ideas, provoking further research thinking in terms of architectural design.

Methodology

This thesis speculates on the advantages of thinking architecture in terms of "adaptation" in an action-reaction fashion. It evolves from the initial idea of mobile architecture but rather pushes and explores the potential of digitally designed responsive buildings and the dissection of its methodological approach. I will look into some natural responsive systems, arguing that buildings can be conceptualized as adaptable living organisms, following nature's methodology.

To demonstrate this I will present a set of computational design explorations that I have conducted through my two years of

studies at the School of Architecture at MIT. These explorations will serve as platforms to explain step by step the methodology followed to set up an adaptive / responsive design system for architectural applications. Furthermore, I will use these exercises as pretexts to explain the importance of computational programming as a mechanism to rationalize, archive and manipulate the forces that drive the design during its conceptual process. In addition, indirectly I will also analyze the role of computational tools and programming languages as meaningful mediums that help designers to better understand, set-up, define and re-define design problems. It will argue that more than an automated provider of an endless number of design solution computers can work as a systematic tool, making us more conscious during the design process.

001. Background

In this section I will mention briefly two topics from which I think the idea of adaptive / responsive systems, directly or indirectly evolved. On the one hand, I am referring to the idea of the machine as the technological tool propelling social and cultural changes. On the other hand, I am referring to nature as a model of a universe in which adaptive / responsive systems are present in most of living organisms.

*"I'm an eye. A **mechanical eye**. **I**, the machine, **show you a world the way only I can see it**. I free myself for today and forever from human immobility. I'm in constant movement. I approach and pull away from objects. I creep under them. I move alongside a running horse's mouth. I fall and rise with the falling and rising bodies. This is I, the machine, maneuvering in the chaotic movements, recording one movement after another in the most complex combinations. **Freed from the boundaries of time and space, I co-ordinate** any and **all points of the universe**, wherever I want them to be. **My way leads towards** the creation of **a fresh perception of the world**. Thus **I explain** in **a** new way the **world unknown to you**"*

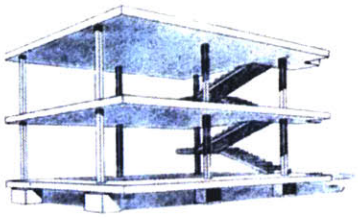
"Quotation from an article written in 1923 by the revolutionary Soviet film director:"
Dziga Vertov, 1923.[3]

011. The Machine / Technology Transfer

I would like to begin stating that architecture has always found in the idea of "the machine" an intriguing concept in which design ideas can be reflected upon. Terms such as innovation, progress, efficiency and technology can be easily associated with "the machine" concept. That was the case of the machine-inspired architecture of Le Corbusier, in which he attempted to introduce technological thinking into architectural design, "transferring" from

[3] Berger, John 1997. *Ways of Seen*. New York: Penguin Putnam Inc.

the industrial thinking an economic and utilitarian vision of architecture seeing it as "a machine for living in." [figure 1.01][4]

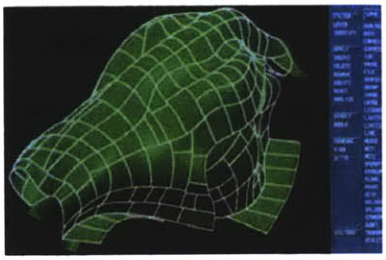


1.01 "a machine for living in".
Le Corbusier: Maison Domino, 1914.

That progressive vision can be illustrated in a more direct way by the suggestive metaphor of the image-reproducer machine: "the camera." Photographs and movies transformed the way designers perceived architecture, [figure 1.02][5] associating through these entities animation-motion with technology. It is possible to see how technology has served as a "lens" through which people can see and analyze things in a different way; therefore, approach things differently.

1.02 Sequence of photographs.
Eadweard Muybridge: Woman and Child, 1877.

Technological production and machines are constantly changing in a similar way that nature follows a ceaseless evolutionary process. Nowadays, designers turn their eyes to the contemporary version of the machine: "the computer" and its software variations, in search for a new of looking into things or just simply to improve existing ones.



1.03 CATIA model.
Frank Gehry: CATIA (Computer Aided Three-Dimensional Interactive Application) model, conference center, DZ Bank at Pariser Platz.

"The computer is a tool, *not a partner.* ***An instrument for catching the curve, not for inventing it"***

Gehry Partners.[6]

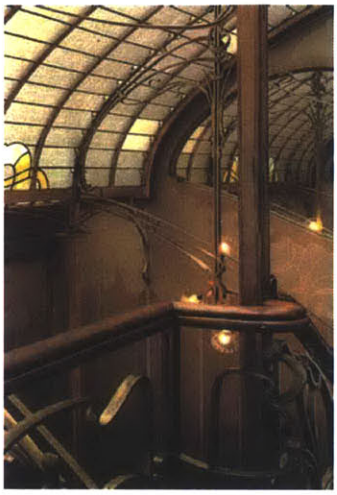
That is the case of architect Frank Gehry, whom like many others, looked outside the box in search for answers to solve his own understanding of architectural limitations; finding in the aerospace industry the right computational tool to overcome those restrictions [figure 1.03].[7] The point that I am getting at is to recognize that technology transfer has been a refreshing activity for architectural practice and still is. I guess, "conceptually" the ideal of the "moving machine" as a technological tool meaning advancement and innovation was an inspiration for me. On the other hand, the implementation of the computer in my design

---

[4] McCarter, Robert "Escape from the Revolving Door: Architecture and the Machine," *Pamphlet Architecture No. 12* (1987): 7-12
[5] Friedman, Mildred; ed. 2002. *Gehry Talks: Architecture + Process*. New York: Universe Publishing.
[6] Friedman (2002).
[7] Friedman (2002).

exercises was the "procedural" motor I relied on to motivate my exploration.

012. Nature: an adaptive system model



1.04 Organic shapes resembling vegetation forms.
Victor Horta: Hotel Tassel, Brussels.

As stated in the introduction section of this thesis, nature has also been a great inspiration that helped me conceptualize the idea of designing adaptable / responsive systems. In the next section, I will illustrate with a few specific examples why there is a direct relation between my research topic and nature.

Historically, nature has always been a source of inspiration for painters, musicians and designers in general. Architecture is a field in which some topological forms from nature have been explored [figure 1.04].[8]

Why nature? By definition, nature is "the forces that control the events of the physical world."[9]

*"**The form**, then, of any portion of matter and **the changes of form** which are apparent in **its movements and** in **its growth**, may in all cases alike be described as **due to the action of force**. In short, **the form of an object is a diagram of forces**; in this sense, at least, that from it we can judge of or deduce the forces that are acting or have acted upon it; in this strict and particular sense, it is a diagram"*

D'Arcy W. Thompson, 1942.[10]

121. Contextual Responsive systems

As I mention before, responsive systems refer to the mechanisms from which some living organisms or objects are capable of understanding the contextual conditions in which they live in. This "understanding" of the surroundings is true when

---

[8] Pearson, David 2001. *New Organic Architecture: The Breaking Wave*. Berkeley and Los Angeles: University of California Press.
[9] Hornby, A. S. 1995. *Oxford Advanced Learner's Dictionary*. Oxford: Oxford University Press.
[10] Thompson, D'Arcy W. 1992. *On Growth and Form*. New York: Dover Publication, INC.

those organisms are capable of sensing and reacting to the environment. Nature has provided these sensorial tools to living organisms in order to achieve adaptation as a mechanism of auto-preservation. One clear simple example of this is the way the sunflower "smartly" reacts to the direction from which the sunlight is coming, following its path in order to get the energy vital to its performance and survival. In a similar fashion, the skin of the chameleon has the ability to change according to specific contextual elements like the color of the surroundings. "Chameleons can change their color thanks to the very complex pigmentation of their skin. Their bodies contain cells called *chromatophores* and *melanophores*. *Chromatophores* are pigment cells that give the skin its color. Chameleons change their skin color by opening and closing the cells called *melanophores*. These cells are used to direct sunlight to specific pigments in the chameleon's skin. *Melanophores* allow the sun to shine on different pigments and as a result the light is reflected back in different colors."[11]

1.05 Chameleon's adaptive skin.
This living being possesses the ability to instantly camouflage itself by changing its skin color to match its surroundings.
Photographs property of Kammerflage Kreations, taken from the site: http://calumma.com

One can argue that the adaptive mechanisms present in most of living organisms, like in the case of the *chromatophores* and *melanophores* cells present in the chameleon, correspond to an automated behavior triggered by an already embedded intelligence [figure 1.05]. In this respect the way nature executes designs is quite different from the way people do it. This is because for us (designers) "automation" not always fulfills our design goals or aesthetic expectations. Designers in this regard will feel compromised and obligated to intervene directly on the designed object to cover those, sometimes subconscious, compositional needs. Having illustrated in a simple way these basic examples I can, empirically, recognize a certain design methodology followed by nature:

[11] "PageWise, Inc." *Why do chameleons change colors?* 2001. <http://www.allsands.com/Science/chameleonschang_rft_gn.htm> (19 March 2004).

122. Nature as a Methodology for Design

Adaptation in Nature is reflected in form.

1. Form as result of action forces. Particular contextual conditions or forces work as signals to living organisms. Based on the reception of those signals, living organisms react (transformations) in order to preserve their existence. **"Form comes from growth, or from the way forces affect materials."**[12]
2. These responsive systems appear at different levels and in different scales, in a layering way. System over system fashion. **"Shapes are influenced by factors ranging in scale from the molecular to the environmental level. Wind, weather, color, force of gravity..."**[13]
3. Those forces "constrain" the organisms existing in the natural universe. **"But constraints are not necessary limitations; they are opportunities for new variations on old themes."**[14]
4. The Question:
   How can we produce a mapping of contextual conditions (external forces) and designer's intuitive decisions (internal forces), when referring to object designing, specifically architectural design?

---

[12] *The Shape of Things*, Produced for NOVA by Peace River Films, 60 min., WGBH, 1985, videocassette.
[13] *The Shape of Things* (1985).
[14] *The Shape of Things* (1985).

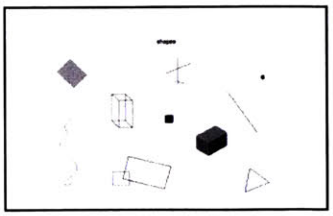
002. Design Explorations

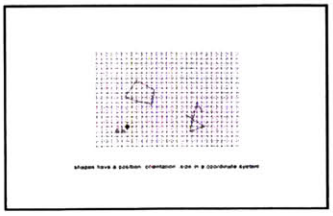
021. Computation: A Methodology for Design

I have presented some examples on how the natural universe (or world) follows a set of rules in order to preserve life, responding to a particular "forces" and contextual conditions. Moreover, and for those reasons, I have stated that nature can be explored not only for the vast range of topological variations it presents but also understand the "hidden" methodology followed by the natural universe in order to achieve "adaptation." I have also mentioned already how in the designers' worlds also exist certain explicit or implicit rules that take an important role during the design process. Furthermore, it has been pointed out how the design activity and in particular the practice of architecture involves a series of mental steps and rational decisions. After outlining these points it is possible to perceive certain similarities in the way "design" is being carried from both worlds: the natural universe and the designer's mind universe. This usually occurs, as I briefly described before, through the implementation of rules and constraints, topics which I will cover in the next subsections: Computation a methodology for design and Design Drivers.

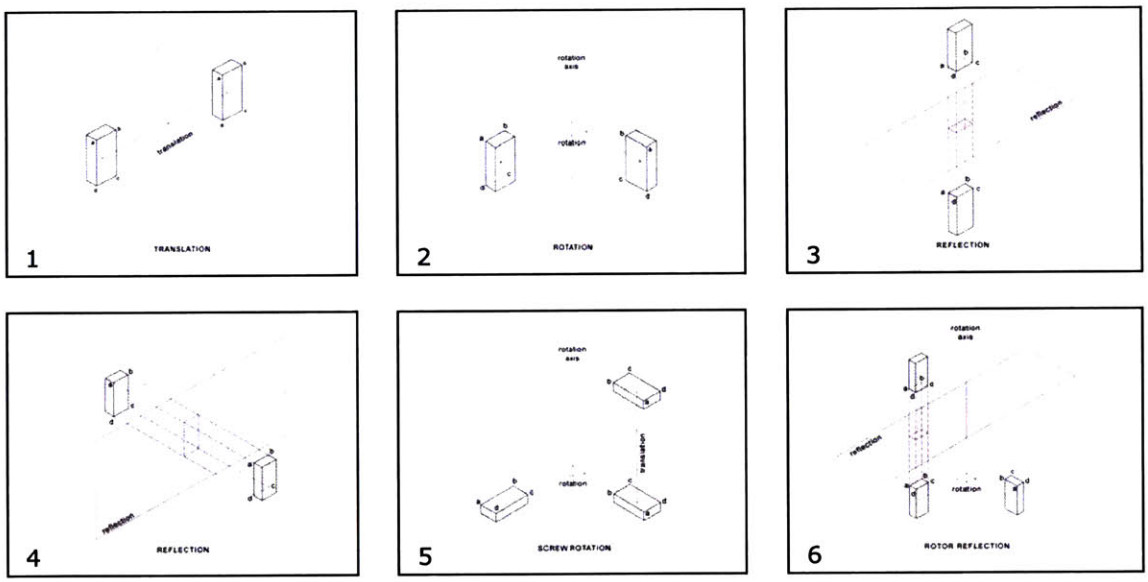
By understanding and analyzing the way living organisms adapt to particular conditions, we are only half way through our challenging task of conceiving "adaptable architecture". It is important to recognize that nature operates as a platform in which all and every one of the living organisms are the "designed objects" and that those objects follow the global rules regulated by the platform. At the same time, living organisms are capable of contextualizing those universal rules according to individual necessities. After understanding that relationship between nature and living organisms as a relation between a governing platform and the objects contained in it, it became clear to me that in order to generate adaptive and responsive objects it was indispensable to set up first a platform or "artificial system" in which one was able to manipulate those objects through rules and conditionals, what I call "controlling mechanisms".

211. Kinetic joints: a Shape Grammar exercise

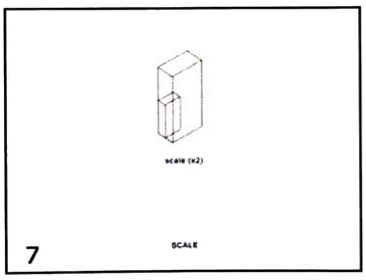




My first personal exposure to the concept of rules as design drivers was through the course Computational Design I: Theory and Applications instructed by Professor Terry Knight during the fall semester of 2002 at the Massachusetts Institute of Technology. This subject introduced me to the topics of design generation through graphical computation or rules called "Shape Grammars."[15] Shape Grammars is a system that describes and generates 2D as well as 3D shapes or objects through computational operations. The 3-dimensional transformations are described as follows: 1. translation, 2. rotation, 3. screw rotation, 4. reflection, 5. glide reflection, 6. rotor reflection and 7. scale[16]. [figure 2.01].



2.01 Shape Grammars transformations.

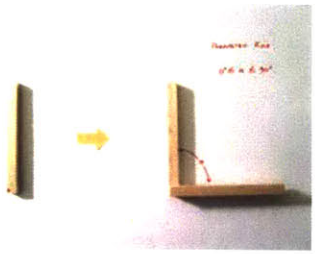


During this course, I was also exposed to the idea of contextualization of rules, in other words to the rationalization activity of connecting rules and their implementation with a goal in mind achieving specific design intentions. To give a concrete example of the applications of Shape Grammars, I am going to touch in my own research work. As an attempt to test the flexibility of the Shape Grammar platform I wanted to explore the

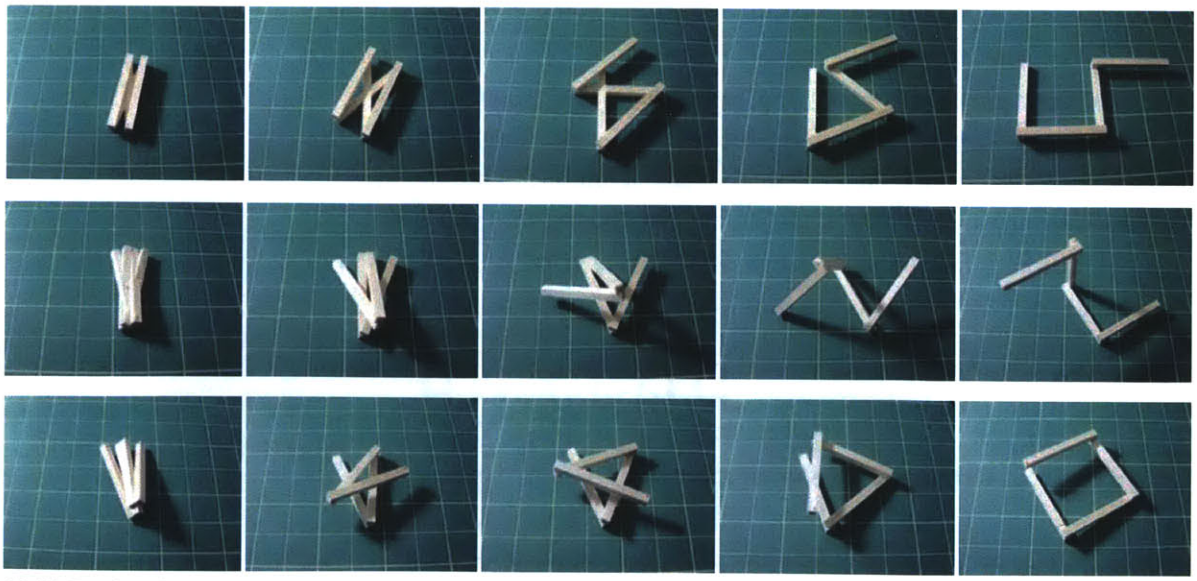
---

[15] System developed by George Stiny and James Gips.
[16] Subtracted from the course: Computational Design I: Theory and Applications; instructed by professor Terry Knight. Fall 2002.

idea of generating a set of rule-based "kinetic" joints[17]. Within this exercise I explored the concepts of expansion-contraction of tectonic nodes or connection points, with the clear intention of embedding levels of complexity and control by manipulating a simple addition rule and the variation of one parameter: the value of an angle. [Figure 2.02] For this exercise all the computation was carried by hand using sketches and balsa-made models, in a trial and error fashion. [Figure 2.03] This was quite interesting because throughout the iterative "looping" process I was able to distinguish a more rigorous method of approaching the act of designing. Moreover, I discovered that by manipulating simple variations in shape configurations (grammars) or numerical values such as degree of angle I was already achieving a low level of controlled "adaptation", in this case responding exclusively to the designer's decisions (me). Although the exercise was restricted by the material constraints of the physical models as well as by the limitations of carrying computation manually, I was able to realize and understand the potential of *computation* as a common (universal) platform, similar to the natural world, capable of controlling the designed objects from a superior level, manipulating them to respond to global conditions, parameters and rules with a specific design goal in mind.



2.02 Parametric rule.
Angle variations.



2.03 Design iterations.
Joint designs based on different geometric relationships and angle variations.

[17] The kinetic connections are just in concept; they are not mechanically designed to perform motion or reconfiguration. From final project proposal for the course: Computational Design I: Theory and Applications. Fall 2002.

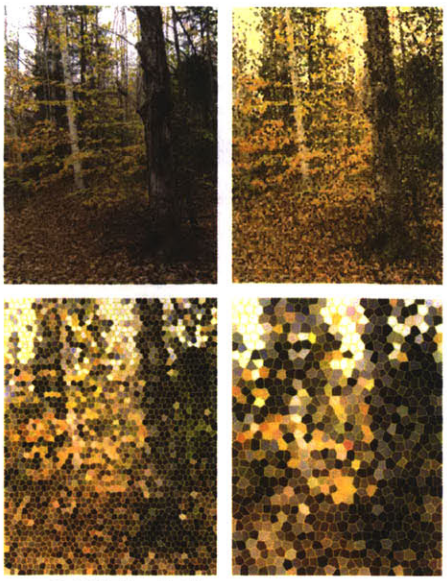
2.03 Design iterations.
Joint designs based on different geometric relationships and angle variations.

From my exposure to Shape Grammars I personally would state that the nicest feature of the tool is the whole idea of computing the conditions through labels, shapes and objects, being especially helpful for people who tend to process information more graphically-visually. Furthermore, after this exploration I can conclude that *computation* is a methodological platform ideal to handle design tasks based on its systematic manipulation of data. I might argue that computation is especially apt to handle design situations where controlled transformations are required, thus computation is a suitable engine capable of overcoming the complexity of designing adaptable objects.

Once I established that computation was the appropriate instrument to control the design of adaptable-responsive objects, I still had in mind the frustration left by the limitations of doing the computation by hand. That limitation made me jump into a series of new explorations. The main objective of those consequent explorations was still the search of embedding adaptation and responsiveness into objects (architectural artifacts). At the same time I wanted to overcome the restrictions of crafting and manipulating manually the rules and conditions, but preserving simultaneously the tangibility of working with physical objects. In this new exploration I was also after achieving certain degree of automation.

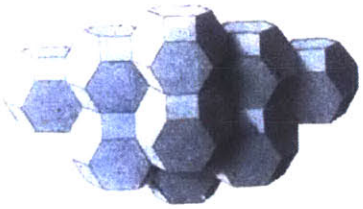
212. Cellular Configurations exercise

This project was an attempt to bridge the advantages of working with physical objects, enhancing them with the highly processing capabilities of the computer, substituting the limitations of manual computation for an automated version. For this exercise



2.04 Pixelation.

I was also looking directly into nature for hints and ideas on how responsive systems can work. It is important to remember that adaptive-responsive natural systems are based on a series of complex communications at a chemical level, just like the production of white cells in blood is triggered by other chemical components present in blood, acting as contextual signals. Based on this understanding I was biased to work with physical models in a kind of "cellular" level[18] [figure 2.05]. It is worth mentioning that the idea of working with tetrakaidekahedrons[19] "cells" was also an inspiration from the intriguing way of how pixels can be manipulated in a computer screen in an intriguing effect called pixelation. [figure 2.04]. With this idea in mind I was trying to produce an infinite number of object reconfigurations, just as simple as how plasma screens rearrange colors of pixels every time it is refreshed.



2.05 Set of tetrakaidekahedra.
Fourteen-sided "cells" achieving close-packing.

"...a platform has to be malleable to be useable as a creative medium. If everything made from the medium looks like a rearranged version of the medium, then it is not particularly useable..."

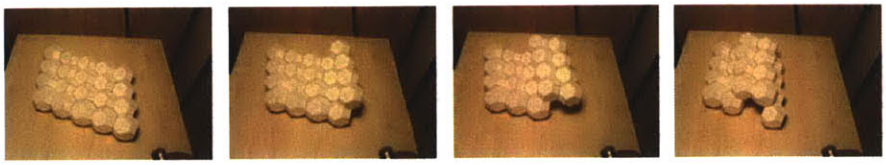
Kelly Heaton, 2000.[20]

But by keeping a physical interface I ran into a different set of problems and complications. In order to generate communication between the cells and readings from the context I was forced to design the means to achieve such a system. I looked back into nature and I saw that living organisms have been provided with "sensorial" mechanisms; just like the five powers of the human body: sight, hearing, smell, taste and touch. Then, the implementation of electromechanical sensors into the cells was necessary to accomplish the desired communication among cells and the required connection to the computer doing all the data processing. Electromagnets were used as mechanical controls or "sensorial" devices, just because they are capable of recognizing

---

[18] I decided to work with physical components at a small scale in a modular fashion in order to accommodate reconfiguration with the least number of objects in terms of shapes, maintaining a degree of simplicity throughout the design exploration.
[19] Thompson (1992).
[20] Heaton, Kelly B. 2000. *Physical Pixels*. Cambridge, MA: MIT.

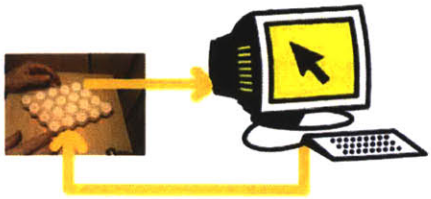
polarities of other magnets, following the principle of attraction-repulsion of physics. The tetrahedron cells were then equipped with electromagnets in order to perform physical transformations / reconfigurations. The idea was that by changing the polarity of the embedded electromagnets the cells would have had the ability to rearrange themselves snapping on and off from other cells, responding to computer signals manipulated directly by the designer acting with a particular design intention. [figure 2.06]



2.06 Cellular re-configurations.
Physical interface responsive to the designer allowing direct manipulation.

In principle the idea of the cells rearrangement works fine but the tangible interface implementation resulted to be a quite restrictive tool, being this the biggest limitation. The controlling platform proposed for this exercise has three different levels. The first correspond to the *computational* transformations executed to the cells (objects) having a local and a global impact regarding objects' relationships, this in theory works similarly to my previous Shape Grammars-based exercise. The second level, which also is similar to the Shape Grammars exercise, is the fact that the designer can directly intervene modifying manually the arrangement of the composition, capable of bending a little the rules or conditions along the design process. Although there are some transformation operations from Shape Grammars like the "scale" rule that can not be applied in this platform, the system still presents a higher level of flexibility over S-G due to the characteristic of being real-time user responsive. The third and final level is the introduction of the computer (synthetic processor) as second controlling mechanism apart from the user [figure 2.07]. This platform also takes the advantages of the computer in terms of the quantity and precision of the information processed, adding some automation characteristics. Regardless of its physical

implementation limitations this project achieves from an abstract point of view a more complex level of adaptation and responsiveness to specific design intentions and designer manipulation.



2.07 Flow of information diagram. The cells can be manipulated directly by the designer and from the computer, receiving input from both ends.

Up to this point I always wanted to keep a tangible interface in order to satisfy my personal need of "direct control" over the artifacts. On the other hand, I became aware that these systems allowed me to have control over individual elements (cells) but not over the system as a whole. Furthermore, I recognized the need of a more organized and powerful way of manipulating and controlling the rules and conditions within a platform in which the concept of adaptation had to be carried out. New computing alternatives needed to be explored. Based on their linear methodology of encoding (storing-accessing) data, conventional programming languages appeared to be the right answer to overcome my evolving design challenge.

213. Responding to Virtual Forces: Small Java Programs

2.08 Virtual canvas area.

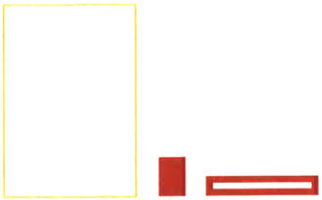
In order to get myself exposed to this new alternative of conceiving objects, using exclusively digital (virtual) platforms, I registered for the course JAVA Programming for Designers, instructed by Professor Steven Ervin during the fall semester of 2003 at the Harvard Graduate School of Design. This was my first direct encounter with conventional programming, specifically using JAVA language. Over the course of this subject I discovered that conventional programming presented a very structured and rigorous way of organizing ideas, storing and accessing data and ultimately executing commands, actions and operations. This has to do with the fact that writing a computer program has to be sequential due to the simple reason that all operations follow a chronological ordered number of steps. To contextualize this idea one can think about a particular operation, like the "translation" operation. If one would like to move (translate) a 2-dimensional object, a rectangle for example, from point X to point Y, one could

give a list of necessary steps to complete the operation. These can be roughly categorized as follows:

1. Determine the area or canvas in which I am allowed to play or move the object in discussion. This action makes us almost simultaneously think about a unit system, in order to declare and solve issues of orientation and scale. [figure 2.08].
2. Determine the actual shape of the rectangle(s), describing its size in terms of height and width. Likewise, here one could determine the color, line thickness, etc. [figure 2.09]
3. Locate the predefined rectangle in its initial default position in terms of X and Y coordinates (treating them as variables) in relation to the canvas. [figure 2.10]
4. Determine the new value of X and Y in which the rectangle is going to be relocated.
5. Finally, execute the operation. [figure 2.11]

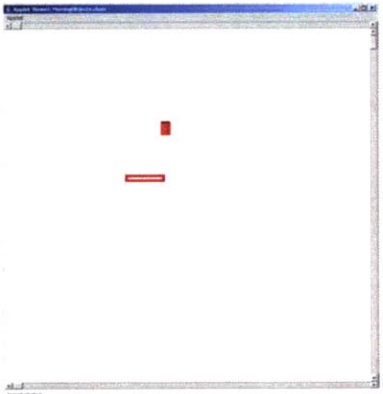
This way of computing is basically what I have done graphically using Shape Grammars. It is more difficult to picture the operation by having no graphics, I have to state that reflecting on a written list of actions helps clarifying the initial objectives of the operation. Similarly, this thorough processing methodology promotes stopping, rethinking, restating and detailing in a deeper manner the entire operational process. As an example of these, I urge you to look (loop) back into my five steps list in an iterative way and encourage you to further detail the objects and operations as you like (adding chamfers, rounding corners, applying other transformation operations such as rotation, changing the color of the canvas or even adding new elements, etc.), modifying them according to your own design criteria, objectives or expectations. [figure 2.12]

As we can see, this methodology (conventional computer programming) allowed me to be more conscious not only about the designed objects and the transformations to be executed on them, but also more conscious about the context and the global conditions in which those objects were sitting and being affected by. Being aware of these capabilities I was ready to attempt to

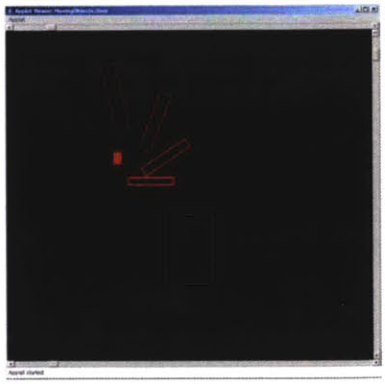


2.09 Shape of initial rectangles.

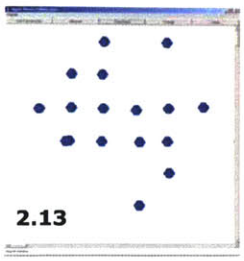
2.10 Rectangles positioned on the canvas area.



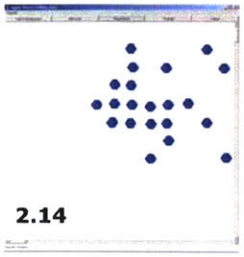
2.11 New position of rectangles.



2.12 Transformations through iterative design process.

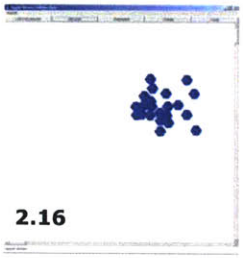


2.13



2.14

2.15

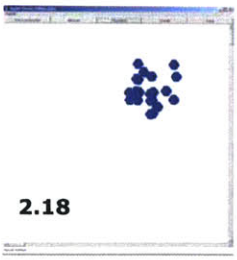


2.16

design a series of context adaptive-responsive objects using this platform, tying it back to the ideal adaptive-responsive concepts present in nature.

In my second JAVA exercise I wanted to create a world in which the objects contained in it would be directly affected by outside forces or contextual conditions. The question here was: What kind of "forces" can affect objects in physical terms? The answer came from one of my previous exercises: The Cellular Configurations experiment. There, I had the chance to explore and implement electromagnets, and that gave me the idea. For my new design experiment I set up the challenge of: How can I synthetically reproduce magnetic and gravitational forces, forcing the objects to react. This problem was quite interesting in principle because for the first time I was trying to virtually map a real physical behavior of natural materials. I basically decided to represent the forces of attraction, repulsion and gravity. In addition, I also wanted to keep the cellular aspect of the objects to be affected by these forces, deciding to represent them as small hexagonal shapes. The outcome was rewarding. Abstractly and regardless its limitations, I designed a responsive virtual environment, the generated "dropped" objects were affected directly by three outside forces: attraction [figures 2.13-2.17], repulsion [figures 2.17-2.19] and gravity [figures 2.19-2.21].

2.17



2.18

2.19

2.20

2.21

After this experiment I was able to conclude that the use of the computer and the systematic way of encoding were essential to attain an artificial world in which it is possible to set up

and manipulate "universal" abstract laws representing some kind of forces to which synthetic organisms would have to adapt / respond accordingly.

Summarizing, it is important to identify which are, in general terms, the valuable and limiting aspects of each computational platform tested up to this stage:

Main positive points:

1. From Shape Grammars: The idea of performing operations using visual elements through shapes and objects.
2. From Cellular Configurations: Direct intervention of the designer along the design process. On the other hand the introduction of the computer, bringing in high speed data processing.
3. From JAVA (conventional) programming: The systematic approach of performing operations and the advantage of having a text file, an actual program or code available to reflect upon.

Main negative points:

1. From Shape Grammars: If the computations are carried manually the iterative process can be time consuming.
2. From Cellular Configurations: The tangible interface resulted to be limiting and restrictive, incapable to topologically deform the shape of the cells.
3. From JAVA (conventional) programming: Specialization of operations (language code) and complexity can be reached quickly. While programming there is no visual feedback, feedback only through functions embedded in the text. For non-experts this platform works fine at a graphic abstract level. It can not be treated as a regular CAD system; JAVA, or any other programming language, can not be considered a drafting tool. In order to achieve such a system it is necessary to manually program the drafting engine.[21]

[21] Refer to the Appendix section: Programming a 3D drafting engine with Java.

The main problem I encountered while working on the Java design experiment was contextualizing my professional necessities; in other words, jumping from working with abstract objects into representing objects with more architectural meaning.

After exposing myself to this encoded way of thinking and producing objects, I started noticing the benefits and set-backs of using a wide-ranging tool like JAVA, which is not exclusively designed to perform architectural applications. It was important for me to recognize the complications of using an open-ended programming platform versus perhaps an already semi-crafted programming tool when pursuing design objectives. Here, as a designer, I am referring to a pre-programmed specialized or somewhat specialized drafting system. For architects, such a platform would be AutoCAD or Rhinoceros for example, tools that already have embedded really powerful drafting engines that have been designed and enhanced through years and years of iterative improvement done by large highly specialized companies like AutoDesk. From this standpoint, it is completely unaffordable for an individual to try to craft from scratch a drafting software using an open-ended programming platform; rather, it would seem more efficient to reshape or mold an existing drafting tool according to my necessities.

The next challenge in my quest to conceptually produce adaptive-responsive architecture was to start working with actual representations of building elements. This was possible by personalizing existing drafting software through scripting:

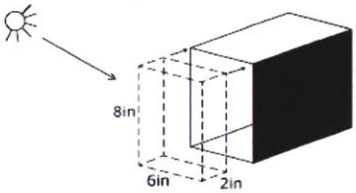
"...a script is a set of instructions written in computer code and executed within a specific software environment. Scripting languages are written using the same basic structures as full-fledged programming languages; variables, loops, conditionals, and functions. ...In addition, scripts can make use of functions already coded into the parent software environment..."

Yanni Loukissas, 2003.[22]

---

[22] Loukissas, Yanni. 2003. *Rulebuilding: Exploring Design Worlds through End-User Programming.* Cambridge, MA: MIT.

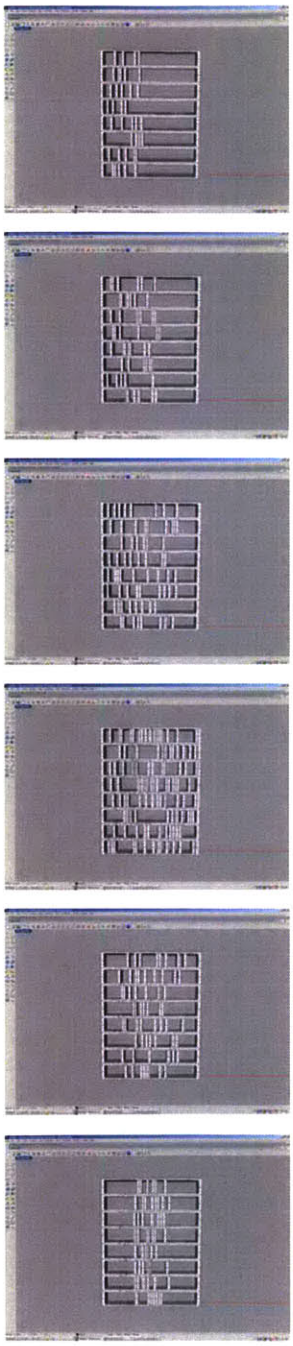
214. The Mutable Curtain: a Rhinoscripting[23] exercise

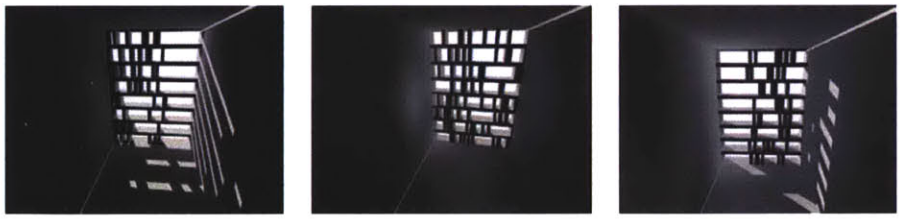


2.22 The enclosure.

This alternative way of programming was first introduced to me through the course Generative and Parametric Tools for Design and Fabrication by instructor Axel Kilian and teaching assistant Yanni Loukissas,[24] during the spring 2003 at the Massachusetts Institute of Technology.

As part of the first assignment we were asked to design a wall enclosure for a box, capable of modulating the light passing through it. [figure 2.22][25] This was a perfect opportunity for me to explore, first of all, scripting programming using Rhino software. Secondly, the problem of lighting was the perfect pretext to adapt this exercise into my own line of inquiry regarding adaptable-responsive architectural systems. For this exercise I considered the sun as a contextual element determining particular conditions to be solved responsively by the designed objects. To tackle this design task I decided to work with the idea of density, playing with two basic components: solids and voids.



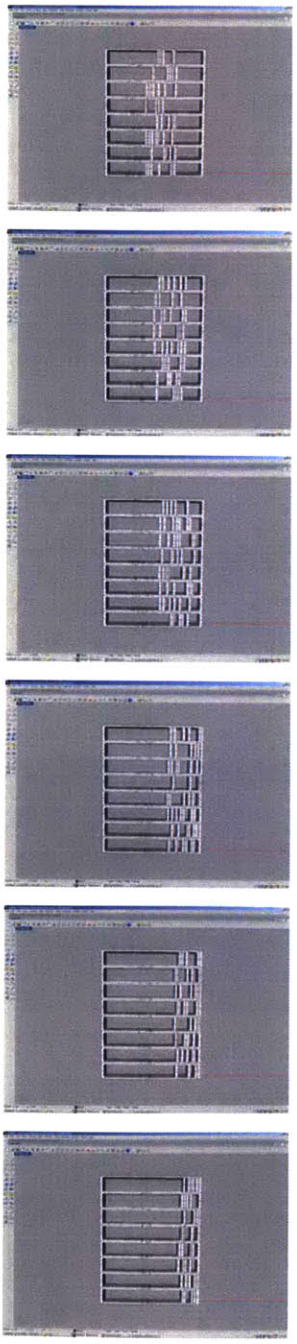


The intention was to produce a façade from a generative design perspective. The approach was to design an architectural 'device' in which the amount of solids versus voids could be pragmatically manipulated; combined with a random function

[23] Scripting language for Rhino based on Visual Basic Programming Language developed by Microsoft.
[24] Axel Kilian and Yanni Loukissas are PhD students at the Design and Computation Group. Department of Architecture at the Massachusetts Institute of Technology.
[25] Subtracted from Assignment 1 of the Parametric and Generative Tools for Design and Fabrication course; instructed by Axel Kilian, Terry Knight and Larry Sass. Spring 2003.

embedding a more spontaneous natural character to the overall design. [figure 2.23]

2.23 Sequence of wall transformations.



As anticipated, Rhinoscripting essentially provided me with a superior drafting modeling engine, giving me the ability to architecturally contextualize my design tasks. Therefore, I was capable to have a better and more accurate representation of architecture-like objects. On the other hand, since scripting is just another variety of computer programming, I ran into the same problem of having no graphical feedback while working on the code. For this particular reason I still was not entirely able to map systematically the effects of contextual forces (the sun) into the object performance (the wall). Instead, I had to manipulate the performance of the wall in terms of the percentage of openings (solids vs. voids) by directly changing some values a specific function, in order to control the quantity of light penetrating into the enclosed box. From this perspective, although the action of the sun was being considered to be the design driver "affecting" the behavior of the wall, the system was being responsive only to the designer's decisions.

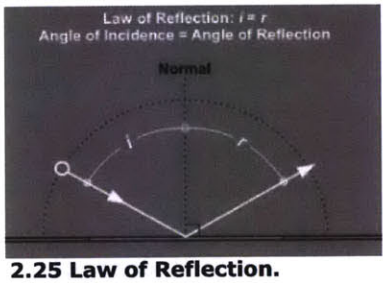
Using scripting, and referring back to Loukissas' statement, I was able to achieve more complex and detailed objects, approximating better representations of architectural devices. In addition, scripting provided me with pragmatic methodology to define procedures and execute functions, similar to computer programming languages like JAVA. These qualities offered me an artificial environment in which I was able to map explicit and implicit forces driving the design. Nevertheless, scripting does not allow having an immediate visual connection between rules and objects just like Shape Grammars does. Another limitation encountered while using the scripting platform was the designer's inability to directly modify the digital representation of the encoded objects, conceivable as a digital version analogous to the "malleable" feature of the tangible interface presented in the Cellular Configurations exercise.

As part of this same assignment we were also asked to solve the same design problem exploring an alternative computational tool besides Rhinoscripting.

215. Responsive Louvers System: a CATIA exercise

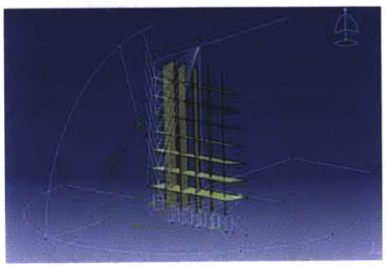
For the second design proposal solution we were encouraged to play with the parametric-based CATIA (Computer Aided Three-Dimensional Interactive Application) software. The exercise's objectives were the same: modulate lighting. In my quest of conceiving adaptable architecture I took it as an opportunity to test my ideas and intentions into this new computational platform. This time I looked back into nature for some hints and inspiration. In nature there are several organisms that in order to survive have adapted to become light sensitive thus light responsive beings. That is the case, among others, of most plants and some reptiles. One of the most obvious examples is the sunflower as explained in the first chapter. [figure 2.24] For this exercise I wanted to translate that responsive behavior into my design.

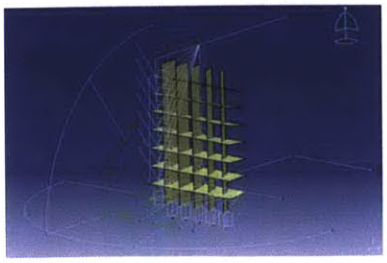
2.24 Responsive Sunflower.



2.25 Law of Reflection.

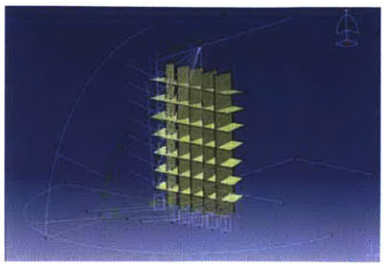
For this second proposal the design intention was to make a set of vertical and horizontal louvers in such a way to be responsive in relation to seasonal and daily sun patterns; being this, the first design condition. The second design condition was to make the louvers orientate themselves in a precise way in order to redirect the reflection[26] of light [figure 2.25][27] into a 3-dimensional point located inside the enclosed space [figure 2.22]. The location of that 3-dimensional point can also be relocated as desired, affecting directly the behavior and performance of the wall [figure 2.26].

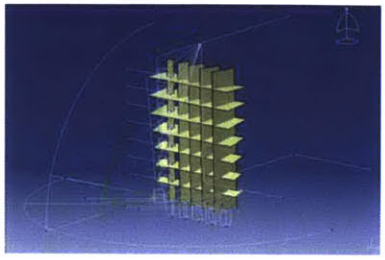


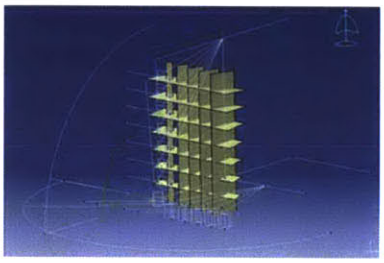


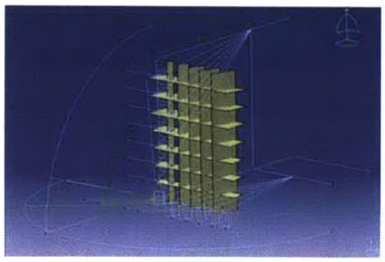
In my search for a tool that would allow me to set up an artificial environment capable of graphically representing "hidden" forces or contextual conditions and associating them at the same

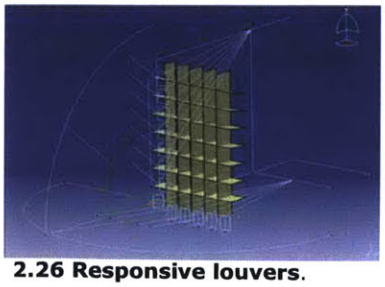
[26] Following the Law of Reflection: where angle of incidence is equal to the angle of reflection.
[27] Image subtracted from Assignment 1 of the Parametric and Generative Tools for Design and Fabrication course; instructed by Axel Kilian, Terry Knight and Larry Sass. Spring 2003.











2.26 Responsive louvers.

time with the graphical representation of "real" objects, CATIA presented several advantages over the previously tested computational platforms. To my eyes, CATIA includes some of the advantages of conventional programming and the advantage of having direct graphic feedback, comparable to shape grammars. In this respect I would catalogue CATIA as a platform that allows encoding functions into objects, in other words programming with graphics instead of using text.[28]

Exploring the parametric capabilities of CATIA I was able to easily interlink relationships and conditionals from objects to objects. In this sense I could metaphorically embed sensorial "intelligence" to the louvers, in order to make them "aware" of contextual conditions (external forces), having a local response or reaction on each individual blade.

After using and comparing CATIA with other alternative computational platforms formerly discussed in this chapter, I decided to continue exploring the capabilities of this software in relation to the conception, generation and production of adaptable architecture.

Before continuing the description of the evolution of design explorations, it is worth pointing out how CATIA (Computer Aided Three-Dimensional Interactive Application) a product of Dassault Systèmes, initially designed to serve the French aerospace industry with the objective to represent complex three-dimensional objects,[29] became the appropriate tool to conceptually test and explore my own thoughts regarding adaptation and responsiveness with an architectural connotation.

In this regard, it is almost mandatory to recognize the crucial role of Frank Gehry, as he was the first to use CATIA as a "foreign" instrument into his own research work. Foreign in the

[28] Besides presenting parametric capabilities in a ingenious way of programming with objects, CATIA also supports scripting using VisualBasic programming language. This topic will be described in the subsection 232. Description of Control Mechanisms under the 232.3 Generative emergence rules topic.
[29] Friedman (2002).

sense that it was implemented in a different realm from the one in which it was conceived to be used. As one can see, Gehry's innovative technological transfer opened endless venues of potential exploration within architectural research. Furthermore, as mentioned in the first chapter, technology transfer becomes, by looking outside the confinements of the profession, the motor that keeps innovation and the promotion of new ideas flowing.

It is interesting to point out how throughout this century the idea of the "machine" has played an important role regarding imprinting new visions into architectural design. Here, one can perceive how the evolution of the "machine" through this period of time has shaped differently the practice of design at every step; starting from the industrialization period at the beginning of the 20th Century with the attraction to the "machine" as the mechanical set of mobile gears,[30] to the current technological perception of the actual "machine" referring to computational processing and automation.

This description touches on the general ideas of mechanical (manual) and automated behavior of the machines. Results interesting how these two characteristics of the machine could be metaphorically related to the ways in which the creative process is being carried-out and executed by designers. Here I am referring to the methodological approach in which the designer, sometimes without being aware, follows a series of steps or descriptions in order to accomplish a particular design objective, series of steps that eventually could be executed "automatically" by a computer program. On the other hand, designers can also manipulate their designs in a more intuitive fashion, taking decisions based on a more empirical understanding of aesthetics or from previous experiences, demanding freedom to "manually" adjust the design outcome along the creative process. This introduces the important concept of Design Drivers.

[30] Refer to the chapter 001. Background.

022. Design Drivers

Design drivers can be described as the motive mechanisms which are controlling the direction and development of a particular design. Architectural design, as mentioned in the first chapter of this thesis, is the process of mapping and representing a series of specific intentions into buildings. These intentions can range from aesthetical personal preferences, functional, financial, etc. depending on the nature of every project and its specific conditions. Therefore objectives and intentions can be understood as design drivers.

In order to avoid confusion I think is also worth making a distinction between design drivers and the means through which those design drivers are being represented. These channels through which design intentions can be produced vary as well; they can be hand made sketches (pencil), physical (materials) and virtual models (computational), etc.

As mentioned before, design drivers in general terms can be categorized as follows:

1. **Following predefined set of objectives and intentions** particular to specific project conditions. Systematic approach described step by step and almost *executable as an automated action*.
2. **Following intuition**. Spontaneous adjustments made based on experience or personal preferences, capable or not to neutralize previous design rules, depending on the *designer's decisions*.

In my quest of designing an adaptable design system I had to take into consideration these two ways in which designers conceive architectural objects. Ultimately, in order to be useful, a responsive design system will have to be responsive to a context or particular conditions; having at the same time the capabilities to be responsive to the designer's intuitive decisions. In other words, the system will need to be flexible enough to accommodate designer's direct manipulation over the virtual objects.

After defining "adaptation" as the design objectives of my exercises and by taking into consideration the two main branches in which design drivers can be categorized, I can distinguish two different possibilities particular to the exploration of adaptable design systems:

1. **Adaptability to external forces**. Referring to *contextual conditionals*. These can include orientation, direction of sunlight, neighbor buildings or objects, streets and pedestrian paths, among other elements.
2. **Adaptability to internal forces**. Referring to *designer's direct intervention*.

Once understanding these two levels in which adaptability can be conceived I would like to illustrate each case in the consequent points: 221. External forces: Context and 222. Internal forces: Designer intervention.

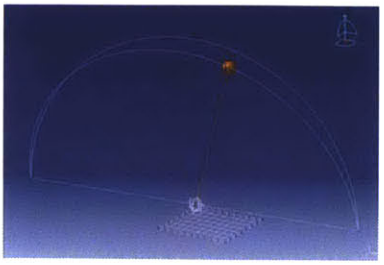
221. External forces: Context

It has been said that in order to design a framework in which adaptability can be carried-out is necessary to contemplate external and internal forces. It has also been established that external forces correspond to contextual conditions and likewise internal forces correspond to the designer's intuition.

As stated previously, after several stages of exploration CATIA proved to be the most appropriate computational platform to virtually model and describe adaptation. Although contextual awareness and responsiveness has been achieved already in the Responsive Louvers Systems CATIA exercise, the intentions of presenting the next CATIA exercise, is to explain graphically in a simplified way how an architectural object can virtually respond to external forces, being specifically for this exercise the location of the sun and the impact of it over the designed object.

In order to continue the set of evolutionary computational explorations, my goal was also to increase complexity along the

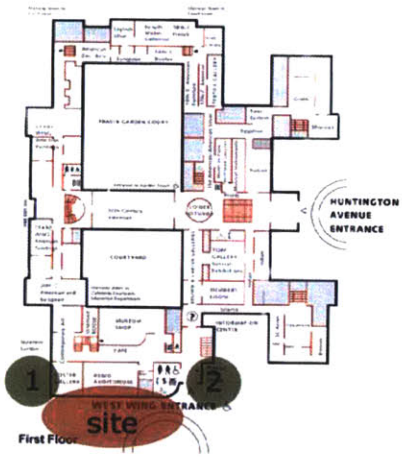
way, jumping from just the "wall" exercise into the task of designing a small pavilion, testing what obtained at elementary level at a more architectural character.



2.27 Solar System. Considering only sun day path. Linear variations from morning to evening.

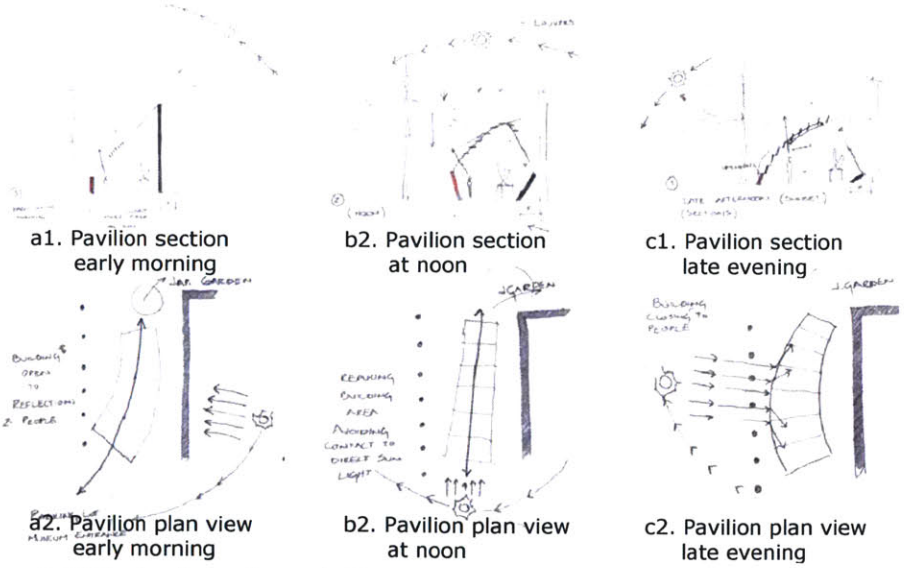
Adaptable Pavilion: Direct Contextual Awareness

This exercise was the second assignment for the course Generative and Parametric Tools for Design and Fabrication co-instructed by Axel Kilian, Professor Larry Sass and Professor Terry Knight during the spring semester of 2003 at the Massachusetts Institute of Technology. In this case, the task of this CATIA exploration was to design a temporary exhibition pavilion for the Museum of Fine Arts in Boston (MFA), serving as a provisional sculpture gallery during the museum's expansion construction. The exercise focuses on the conception of a contextual responsive building, addressing different variations on the overall building's form and its envelope in relation to sunlight [figure 2.27], site [figures 2.28 and 2.29][31] and the program / design intentions [figure 2.30].



2.28 The Location. West wing of the MFA, between (1) the Japanese garden and (2) the west wing entrance.

2.29 The Site. MFA's West yard, taking into consideration the orientation and existing elements: line of trees and MFA's blind wall.



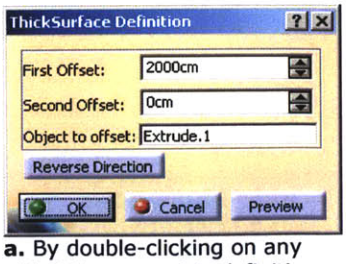
2.30 Design Intentions: 1. Generate an appropriate enclosed environment for the sculptures to be exhibited; 2. Implementation of a responsive louver system as a filter to redirect light incidence inside the pavilion; 3. Analyzing the behavior of the building towards conceiving a sun responsive building. In sketches a, b & c is possible to identify, in plan and section, the basic geometric characteristics of the pavilion at three different stages: early morning, at noon and late at night; aiming to achieve a linear adaptation in relation to the sun path.

[31] Plan taken from the MFA's general information brochure. Boston, MA.

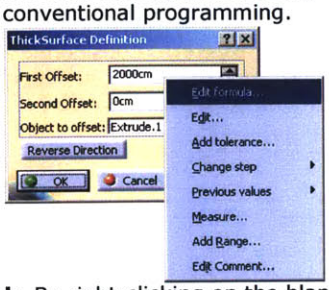
After explaining the main elements and conditions involved in this design exercise it is worth mentioning that the context consists of three basic objects: the sun, the museum blind wall and the trees defining the yard to be used as site. In which the blind wall and the set of trees were static given elements, meaning that relocation of these objects was not allowed. On the other hand, the sun had to be understood as a dynamic object; therefore, its variations in space and time had to be taken into consideration as input, having a direct influence on the behavior of the pavilion already predetermined by the designer [figure 2.30].

It is obvious that all initial design decisions, such as the pavilion's shape, length, height, proportions, etc., were taken by me (the designer) in the first place. It is also important to mention that I was the one deciding deliberately to be the location of the sun the "external driver" controlling the overall performance of the design. After having decided among all these choices the next step was to compute (program) all these data into a CATIA file.

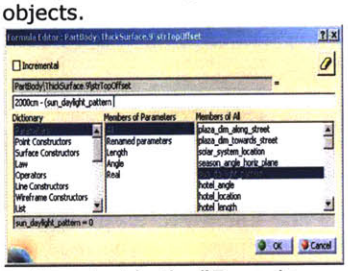
As stated earlier in this section, after having outlined the objectives and intentions that the building has to accomplish, they can be arranged systematically step by step. Therefore, programming these design drivers and geometric conditions is a matter of mapping object relationships [figure 2.31], following the conditional *"if : then :: else : that"* fashion. To be more project-specific, *if* the location of the sun in relation to its day-path line is *"n"*, *then* the line defining the pavilion's profile gets increased or reduced or rotated or thickened, etc. by predefined *"n"* factor [figure 2.32]. After feeding the computational platform with all these data, the conditionals can be executed "automatically" by the computer. Therefore, just by changing a single numerical data linked to the contextual object, the sun location for the case, I have achieved the indirect manipulation of the pavilion's geometry through a set of highly complex computational operations, previously programmed. This is what I refer to "automated" computer processing. What is interesting is that the objects



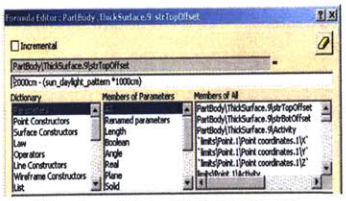
a. By double-clicking on any object its geometric definition expressed in numerical terms pops-up in dialogue box. These values are editable; this action is analogue to the act of accessing the code or script when doing conventional programming.



b. By right-clicking on the blanks, where the dimensions of the objects are defined, I can select "Edit formula"; by doing this a new window, shown in the figure below, will pop-up. There, I can set-up relationship rules between objects.



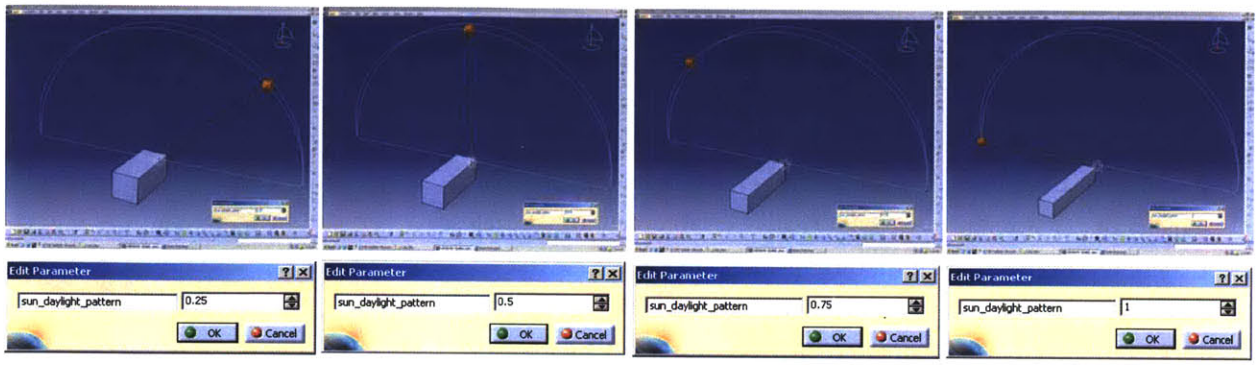
c. Once inside the "Formula editor" I can interlink objects just by simply clicking directly on existing elements present in the virtual universe.



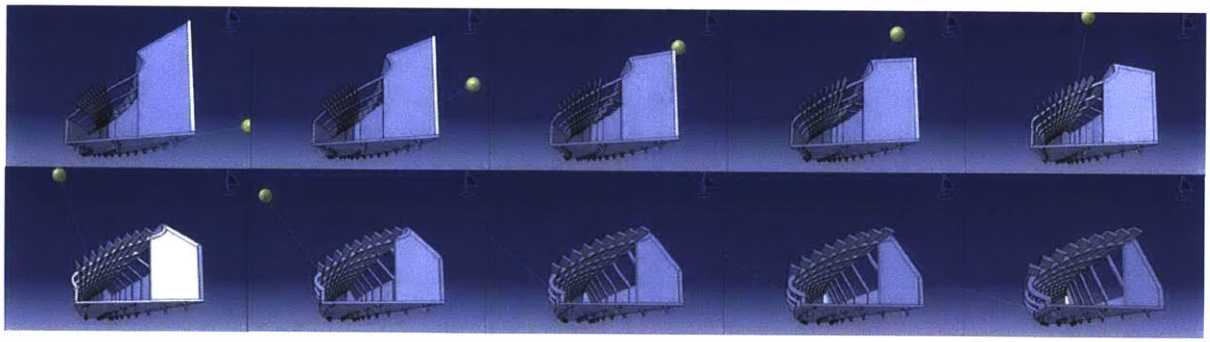
d. The location of the sun is defined in a percentage (i.e. 0.0, 0.1, 0.2... 0.9, 1) of the total length of the arc representing its path. These numerical values are contained in the function **sun_daylight_pattern** programmed by me.

2.31 Mapping relationships. Process followed to interlink particular conditions between objects.

programmed in CATIA become also the interface through which I can manipulate the variations to the external design driver. This is performed by "sliding" the location of the sun thru its path, triggering with this action all the conditionals and up-dating, in a coordinated adaptable-responsive performance, the geometry of the pavilion [figure 2.33].



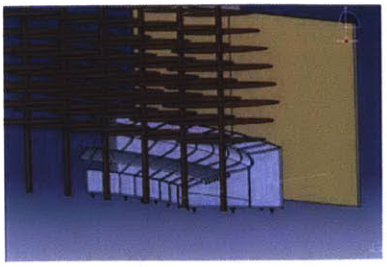
2.32 "Automated" adaptation. After setting-up the relationship between objects, I am able to manipulate, in a slider-like way, the parameter functions related to external design drivers having an indirect impact over the designed objects; achieving contextual awareness and responsiveness, therefore virtual adaptation. In this example the location of the element representing the sun in relation to its arc path is used to modify the shape of the rectangle. Then, I can argue that this "smart" rectangle is able to recognize the time of the day, being also capable to adapt to these contextual variations respectively.



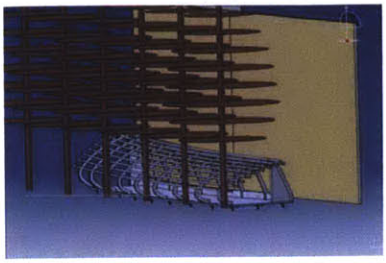
2.33 Adaptable Pavilion. In a more complex but similar way, this pavilion responds to the design intentions explained on the figure 2.30, being the sun path the external design driver.

Through this CATIA exercise I have explained the process of conceiving and implementing an adaptable architectural object being driven by contextual conditions. Although the museum's blind wall and the set of aligned trees were contemplated as important elements providing data input to the design process, they were basically, as stated earlier, static elements delimiting the usable area. The location of the sun was the only dynamic

(varying input) contextual element providing new conditional values. I also mentioned that a rigorous identification of the main intentions and design objectives can lead, in CATIA terms, to a systematic conception of the program, being able to achieve computational automation once the conditional data was embedded into the objects in question, by indirectly influencing the "adaptable" objects through the direct manipulation of data corresponding to contextual forces or elements.



a. Extreme A



b. Extreme B

2.34 Predefined performance. Operating from extreme "A" to extreme "B"

After this exploration I can conclude that in my quest of conceiving a responsive design system I have achieved the design of a synthetic platform, similar to the natural universe, capable of sensing and adapting to external / contextual forces. Nevertheless, it is important mentioning that in this particular exercise the designer is not allowed to perform any changes nor adjustments directly to the pavilion geometry. In other words, this particular system works in a predefined way operating from extreme "A" to extreme "B" [figure 2.34] and all the in-between steps, but leaving the designer's intuition aside.

222. Internal forces: Designer intervention

It has been pointed-out that in order to have a true responsive system it is necessary for such a system to be responsive to external and internal forces. I have also stated that internal forces refer to spontaneous intuitive decisions taken by the designer along the creative process, generally based on experience, empirical knowledge or personal aesthetic preferences.

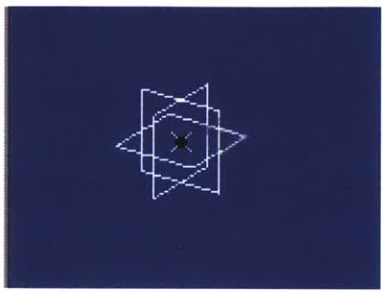
When referring to internal forces I can distinguish two ways in which the designer would be able to modify the designed objects:

1. By adding completely new elements into the design
2. By tweaking / transforming existing elements.

Based on this differentiation, I can argue that adding new objects to a design is equivalent to reprogramming the system, achievable by accessing the code and adding new functions or

commands. On the other hand, the question of directly modifying existing elements is a more challenging one. This is due to the fact that modifying an existing element requires re-setting the element's geometric definitions. It is this last point the one I am particularly interested in approaching, taking advantage of the easy manipulation of objects using CATIA.

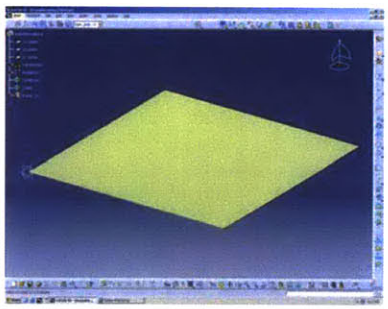
In this regard, my quest was to find a way in which I could directly manipulate (manually click and drag) the definition of the elements in my design without the necessity of accessing and editing the program's code. In the next CATIA exploration titled "Stretchable Surfaces" I am illustrating an alternative approach of conceiving a flexible platform capable of accommodating the designer's intervention by playing directly with the virtual objects.



2.35 CATIA origin. Referential point in space, provided by default, from which all the elements define their location inside this 3-Dimensional synthetic universe.

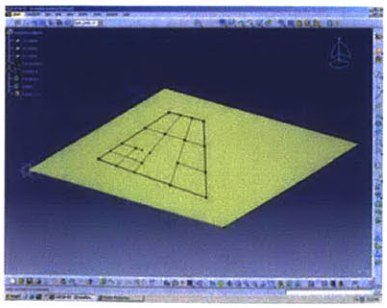
Stretchable Surfaces: Allowing Human Intervention

The intention of this exploration was to design a system in which direct manipulation of existing objects was viable, not by creating new elements but rather being able to transform the existing ones.



2.36 Surface programmed by extruding a line which starts from the origin.

In order to understand the "stretchable surfaces" concept, is important to mention that CATIA does not allow drawing isolated objects. This means that all the objects programmed in CATIA have to have other objects[32] as references, having ultimately a connection to the origin of the virtual universe, represented by the (x, y, z) coordinates with value equal to zero (0, 0, 0). [figure 2.35]. This constraint restricts the user to move objects arbitrarily inside the universe. I realize that in order to overcome this limitation I needed an object, a "virtual sheet of paper or surface" linked to the origin, giving me at the same time an area in which I was able to freely draw or move objects on top of it. By doing this, the objects programmed on top these surfaces are being directly and indirectly interconnected to the surface itself and to the origin respectively, solving the problem of origin-linkage that all CATIA objects have to maintain.



2.37 Objects on the surface. These sets of points and lines can be drawn just by selecting the surface as platform.

[32] Elements such as points, lines, planes, surfaces, solids, etc.

While exploring the idea of the Stretchable Surfaces, I discovered that drawing on top of a surface allowed me do four important things:

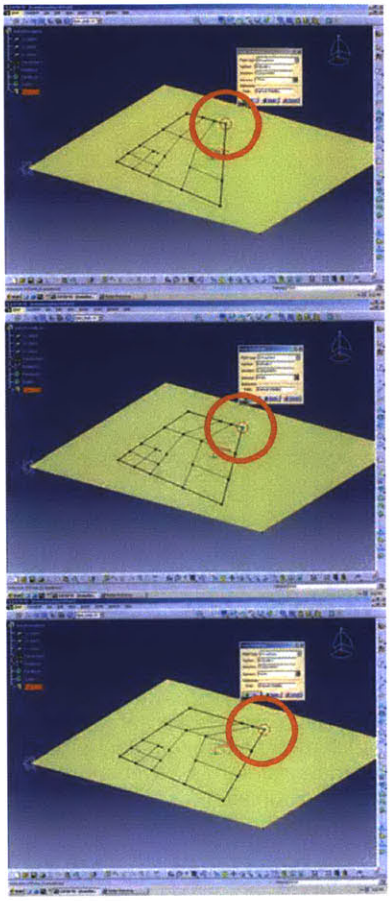
1. Delimit a usable territory or area. [figure 2.36]
2. Draw objects on top of the surface. [figure 2.37]
3. Freely move the objects placed on top of the surface, being the boundaries of the surface the geometrical limitations to perform the action. [figure 2.38]
4. Redefine the geometry of the surface and the objects drawn on top will follow. The objects drawn on top the surface will relocate themselves proportionally in relation to the new geometry of the surface. Action executed internally by CATIA [figure 2.39].

After these explorations I can summarize this section saying that design drivers are basically ways in which designers can control and manipulate objects during the design process. It is also important to remember that in the quest of designing responsive / adaptive systems, design drivers can be categorized in two main branches:

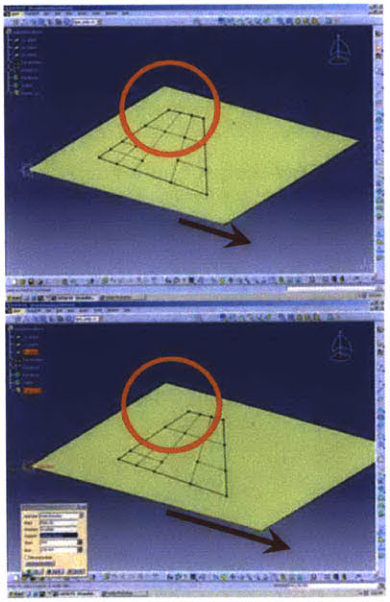
1. **External forces**. Referring to *contextual conditionals*. Following a predefined set of objectives and intentions.
2. **Internal forces**. Referring to *designer's direct intervention*. Spontaneous adjustments following intuition.

And finally I can conclude, as illustrated through these set of exercises, that CATIA, due to its parametric engine and the way it interlinks objects and conditionals, is a very reliable computational platform to conceive and model adaptable systems, providing the designer with the appropriate features necessary to conceptualize synthetic worlds where external and internal forces can be mapped and manipulated.

After exploring each of these topics in an independent way, I was eager to test them together under the same virtual universe; becoming this, the purpose of my next and last CATIA exercise.



2.38 Moving objects around. The points liberate by double-clicking on them, being able to drag them around inside the surface.



2.39 Expanding the surface. The geometry of the surface can also be redefined. Redefining simultaneously the geometry of the objects placed on top.

023. The "Kendall" Pavilion: A Comprehensive CATIA Exercise

This exercise was produced as means to test and explore design concepts in my quest of conceiving responsive design systems, during the spring 2003 at the Massachusetts Institute of Technology. This exercise is part of an independent design research in which professor Terry Knight advised me throughout its conception and development processes.

231. The Challenge

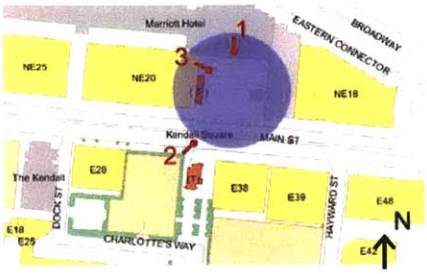
The main objective of this exercise, as it has been throughout previous design explorations, was to explore the idea of adaptation within an architectural context.

The speculation. Having demonstrated that I was able to map and model "adaptation" using CATIA software, my next challenge was to build a computational responsive system in which I could include all the different independent computational approaches, combining all the design concepts and control mechanisms discussed up until this point. Namely:

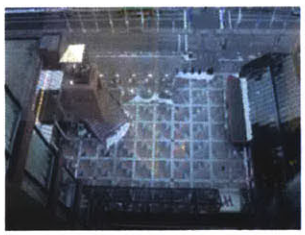
1. Programming with objects in a Shape Grammar fashion.
2. Conventional programming through coding or scripting.
3. Implementing the idea of designing a synthetic universal platform, able to map external and internal forces and control objects both globally and locally, (modeling situations similar to phenomena happening in the natural world).
4. And finally, designing a system capable of handling direct user intervention through the implementation of "Stretchable Surfaces" concept.

The task. In this explorative design task the main challenge is not the architectural program, rather the challenge

resides in designing an architectural envelope (enclosure system representing a building) capable of performing responsive geometrical reconfigurations based on the idea of adaptation in relation to a geographical location [figure 2.40], urban context [figure 2.41] and to specific design intentions, responding implicitly to the designer's direct manipulation. [figure 2.42]



2.40 The Location. Kendall Square, Cambridge. At the intersection of Main Street, Broadway, and Third Street.

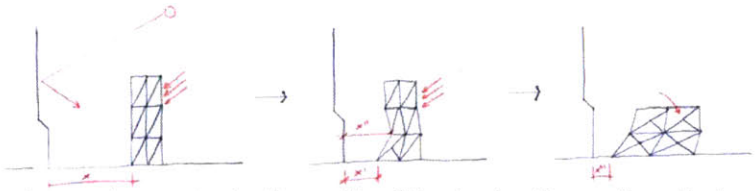


a. Aerial view (1)

b. Perspective across the street (2)

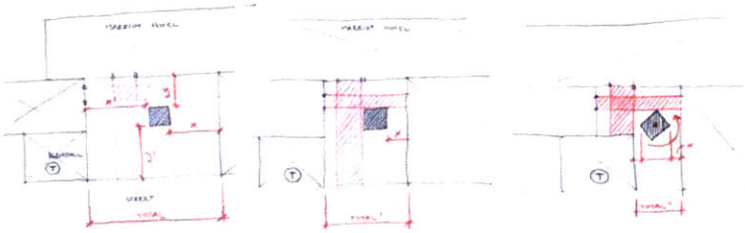
c. Panoramic view from Kendall Square (3)

2.41 The Urban Context Description.
The space designated to locate the pavilion is surrounded by high and medium size structures: Marriot Hotel, 26 stories height approx. (North); Office and retail space, 20 stories height approx. (East); Subway station office and retail spaces, 4 stories height approx. (West). Refer to the plan view [figure 2.21] to locate the position and direction from where the photographs were taken.



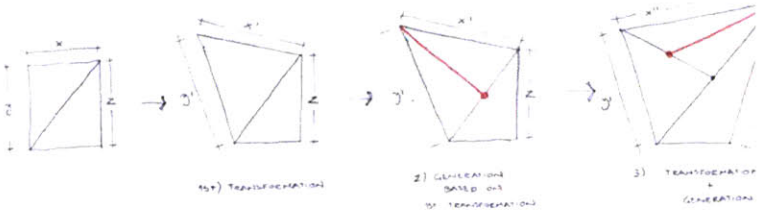
a. Proposed geometry for the pavilion (Simple elevation configuration)

2.42 Design Intentions:
1. Generate an architectural skin (enclosure system representing a building).



b. Pavilion (darker rectangle), responding to changes on immediate urban context. Considering relocation of surrounding buildings (outlined) and pedestrian paths (hatched rectangles in red).

2. Embed "sensorial intelligence" into the building, providing it with object recognition capabilities (surrounding buildings and pedestrian paths) and sun location responsiveness. (**Parametric design approach**)



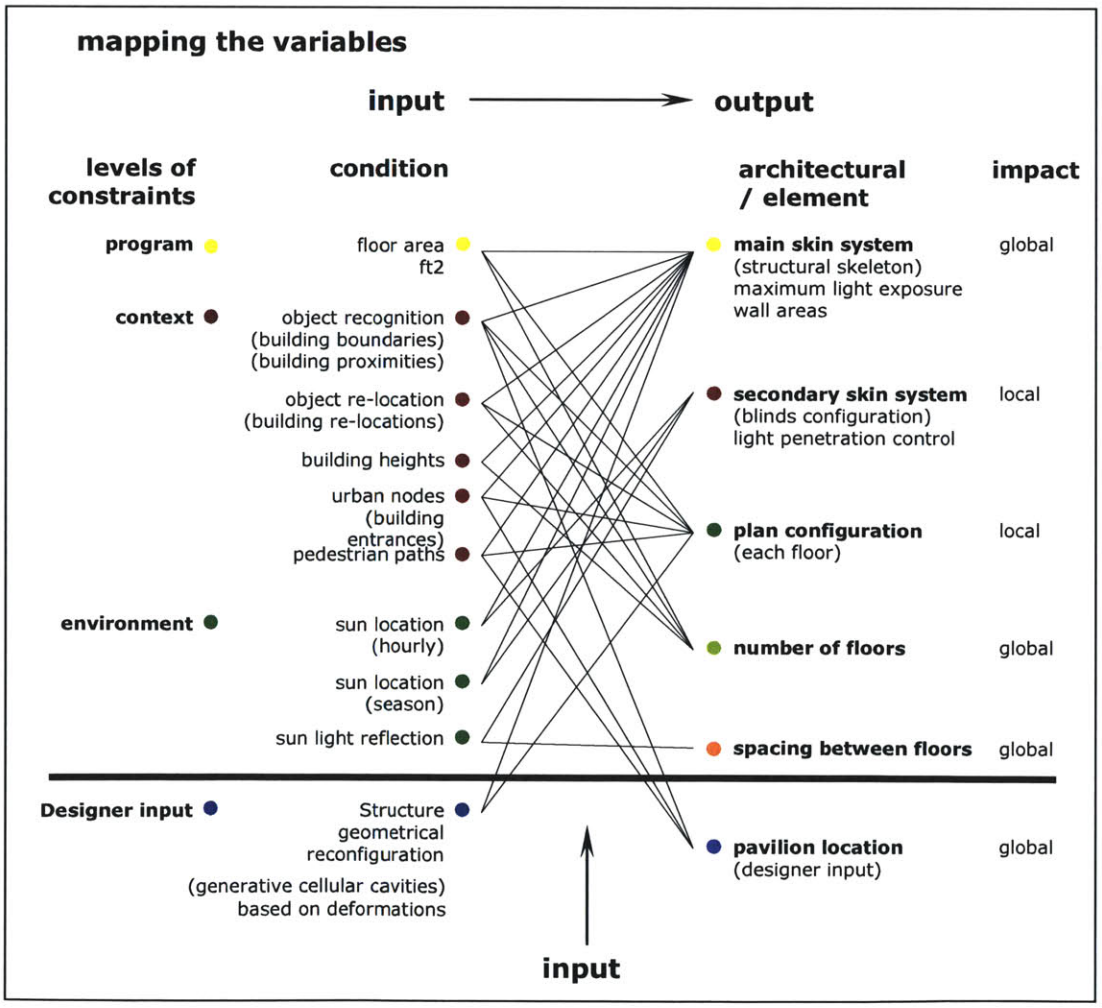
c. Pavilion' skin configuration close-up. These sketches show hypothetical deformation of the structure and execution of the "self-fixing" mechanism.

3. Embed "structural intelligence" into the building' skin system, recognizing and fixing potential structural instabilities, due to the fact that the building will suffer geometrical reconfigurations while responding to stimulus. As seen on the sketches the system will be "smart" enough to "drop" structural members when certain (arbitrary) dimensions get exceeded. **(Generative design approach)**

After understanding the context and stipulating the objectives, I had to find a way to understand all the factors and conditions affecting the pavilion before start "programming" it. In order to achieve that understanding, I started the task of decomposing and systematically mapping all the elements involved in the problem, analyzing their interrelationships.

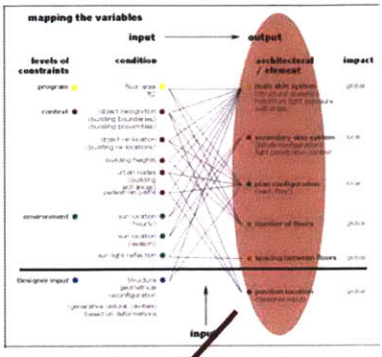
The mapping process. I can identify four different areas from which the building is going to be receiving stimulus:

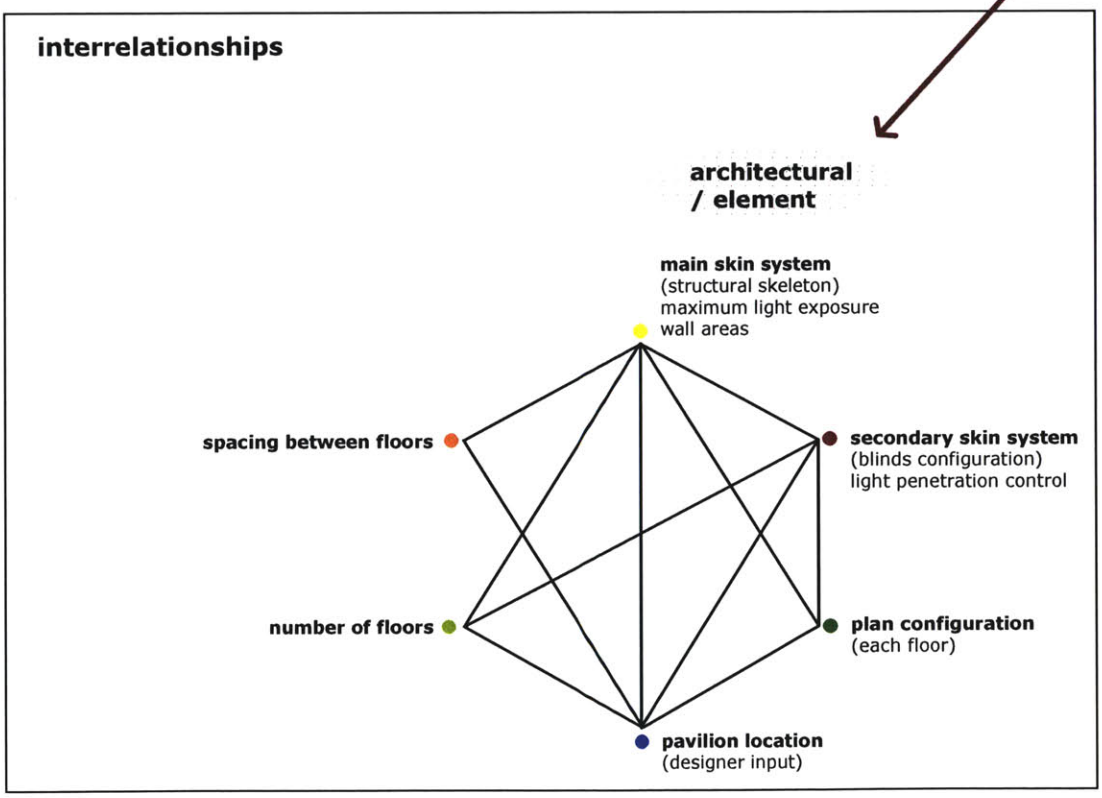
- **Program**
- **Context**
- **Environment (Geographical location)**
- **Designer input**



2.43 Mapping variables.
External design drivers: Program, context and location
Internal design drivers: Designer Input

In the graph 2.43 I am showing how the design drivers are decomposed into specific elements and conditionals which in the form of data will function as input affecting the design of specific architectural elements of the pavilion. This analysis made me realize that once the conditions were ruling the design, a continuous recursive loop of information was taking place during the design process. To illustrate this let's think about how the configuration of, what I call, the *main skin system* [figure 2.42a], which belongs to the *architectural element* category, gets affected every time the *pedestrian paths*, which are variables or conditionals, get relocated. Likewise, every time the *main skin system* gets reconfigured it will indirectly affect other elements belonging to the *architectural elements* category, such as the plan configuration. Due to this self-data-feeding I decided to map out the relationships among *architectural elements*. [figure 2.44]





2.44 Elements affecting elements.
Indirectly happening throughout the design process.

Although the exercise, due to computational and time limitations, did not in the end follow all the details in those maps, the actual process of mapping the elements, variables and conditionals made me more conscious about my own design process and helped me understand more clearly the implications of designing such a robust system.

232. Description of Control Mechanisms

Having the main variables and design intentions laid out, I was ready to start programming and modeling all the information and data. This exercise is about designing a universe in which different responsive design systems work simultaneously, feeding and complementing each other. In order to accomplish this task it was necessary to conceptualize mechanisms through which I was able to control all those systems embedded into that synthetic universe. Summarizing, this exploration is basically dealing with three central ideas:

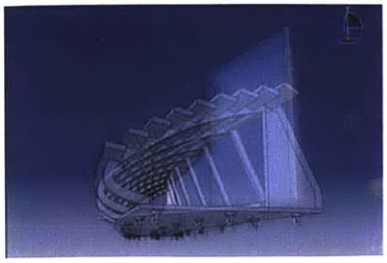
1. Embedding "sensorial intelligence" in order to achieve object recognition. To be achieved through contextual based conditionals, further explained in section 232.1
2. Embedding "smart flexibility" allowing the user to intervene directly and "freely" with virtual objects throughout the design process. Implementing the concept of "Stretchable Surfaces", discussed earlier. Put into operation through locally based conditions, concept further explained in section 232.2
3. Embedding "structural intelligence" in order to achieve specific self-fixing capabilities. Introducing the concept of emergence, to be implemented locally through a rule based approach, explained in section 232.3

These objectives will be tackled respectively and illustrated through these computational approaches:

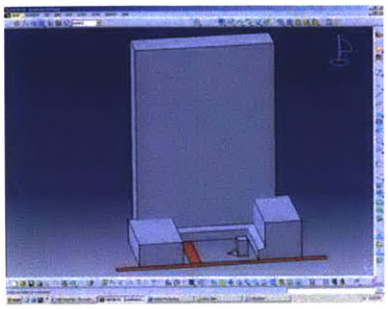
1. Parametric representation of context.
2. Parametric delimited action: Semi-automated human intervention.
3. Generative emergence rules.

232.1 Parametric representation of the context

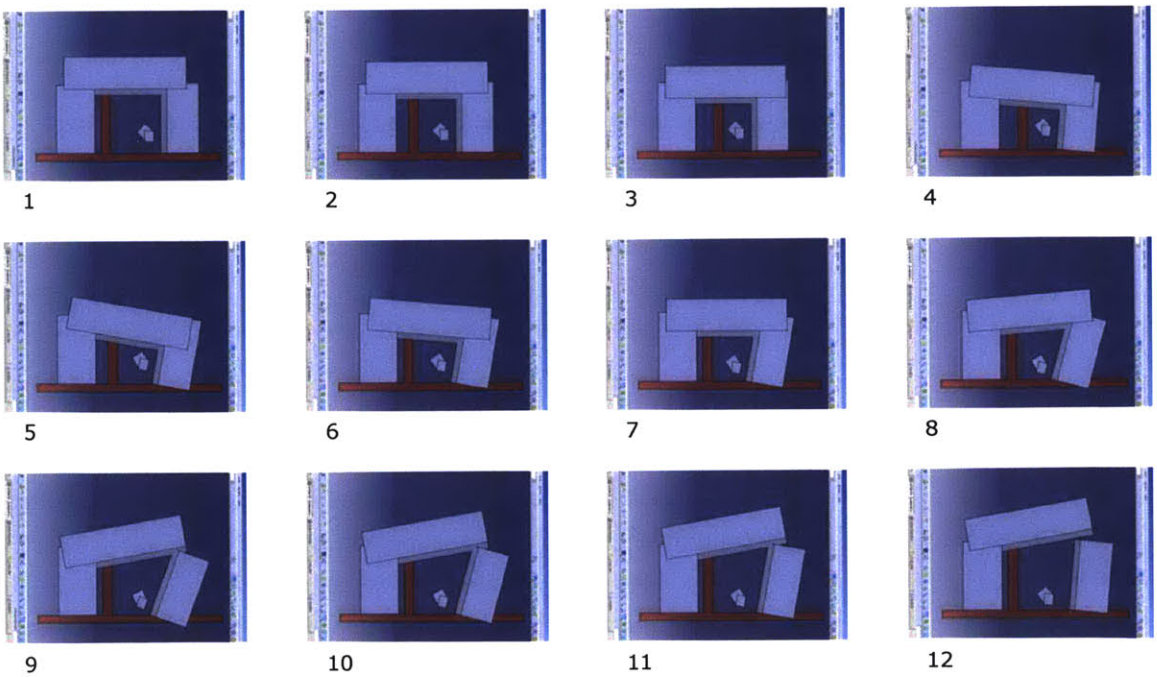
Setting up a parametric version of the context was a crucial action. If this exploration is directly related to "adaptation" and designing "responsive" systems, it means that not only the objects to be designed but the entire set of virtual elements embedded inside the synthetic universe have to accommodate adaptation up to a certain level. Prior to this exercise I presented an exercise where a building was performing adaptation in relation to its context. I am referring to the "Adaptable Pavilion" exercise done in CATIA [figure 2.45]. In that example, the sun was the only contextual element driving the design while performing some kind of adaptation. For the "Kendall Pavilion", on the other hand, I am conceiving not just the sun but the surrounding "buildings and pedestrian paths" as objects that can accommodate relocation [figures 2.46 – 2.47]. This measure was necessary in order to have a "pretext" to implement and test "sensorial intelligence" (object recognition) into the new pavilion. Here is where the importance of "programming" a parametric context resides.



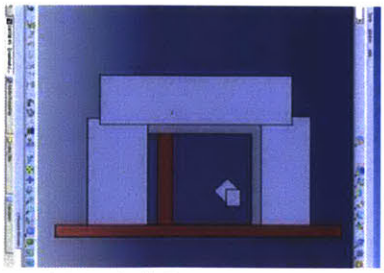
2.45 Adaptable Pavilion. Presented to illustrate external design drivers in section 221. in this thesis document.



2.46 Kendall Square. Isometric view, showing buildings (solids) and pedestrian paths shown in red.



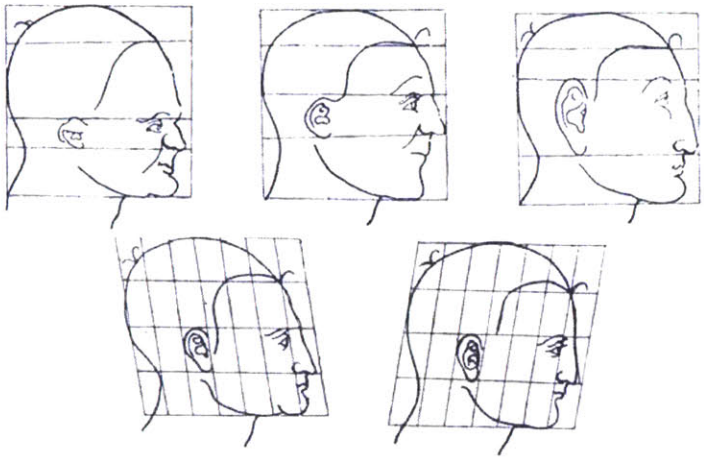
2.47 Contextual Parametric Variations. Plan view of Kendall Square area, showing how built context can be now manipulated.



2.48 Kendall Square.
Plan view

Once a parametrically malleable context was achieved, the next challenge was to design a system through which I was able to retrieve the resulting geometrical data after every single contextual reconfiguration. Executed by the designer. This was essential due to the fact that the "Kendall" pavilion was conceived to be located in the plaza confined by the objects and paths in question [figure 2.48].

This became an interesting problem. Basically I was looking for a structure flexible enough to accommodate transformations, tracking at the same time various locations in a 2-dimensional coordinate system. A grid system seemed to be a promising approach. "This method was used in the sixteenth and seventeenth centuries as an elementary application to the study of proportions and the human form. This method is probably much more ancient, and may even be classical; it is fully described and put in practice by Albert Dürer in his *Geometry*, and especially in his *Treatise on Proportion*." [figure 2.49].[33]

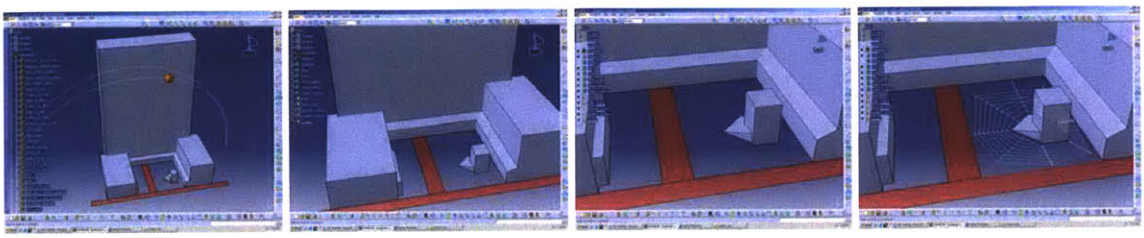


2.49 Geometric transformations based on grid deformations.
Dürer's human head proportions study.

Taking these previous studies as inspiration I decided to approach my problem in a similar way. By laying out a set of lines forming a responsive net I was able to track the configuration and proportions of the plaza. The process started by connecting the inside and outside corners and middle points of the emerging

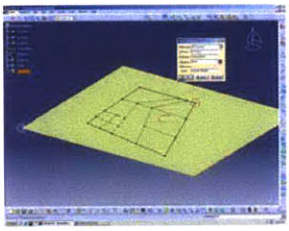
[33] Thompson (1992).

geometry of the urban composition, taking them as guidelines to generate radial lines taking the existing kiosk as central point [figure 2.50]. By doing this I was able to design a "sensorial" system from which I can retrieve data regarding the configuration of the area, to be used during the design process. In this regard, this is an adaptable / responsive platform, independent from the design but essential to read the context and its signals.



2.50 Transformations "Sensorial" System.
Based on a parametric structure of lines forming a net, synchronized geometrically with the contextual objects.

This system proved to be very helpful in the development of the entire design process. The parametric net facilitated the implementation of the "Stretchable Surfaces" system.



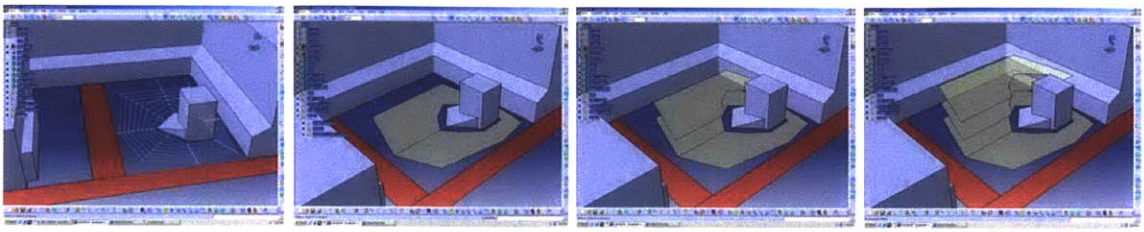
2.51 "Stretchable Surfaces".
Concept explained in the Design Drivers section.

232.2 Delimited action: Semi-automated Human Intervention

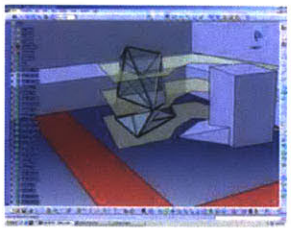
In this section, I will illustrate the implementation of the "Stretchable Surfaces" idea into a particular architectural application. Here I decided to take advantage of the parametric net implemented to recognize the context. The idea was to make the "Stretchable Surfaces" system part of an existing system, building up step by step a more complex but "smarter" design system [figure 2.51]

By embedding the surfaces into the net system I achieved a 2-directional mechanism. On the one hand, the surfaces became sensitive to the context, simply by the fact that they are "attached" to the responsive net. On the other hand, the "Stretchable Surfaces" system gives, by itself, all the flexibility

necessary to allow the designer to play directly with the objects "programmed" on top. Another important feature is that I can implement the concept of the surfaces not only at the ground floor level but surfaces can propagated into different floors [figure 2.52] or even used as vertical planes defining walls or vertical elements (not tested in this exercise).



2.52 Implementation of the "Stretchable Surfaces".
Connected to the parametric net and performing as a canvas in which the designer can play freely and directly with the objects placed on top of it.



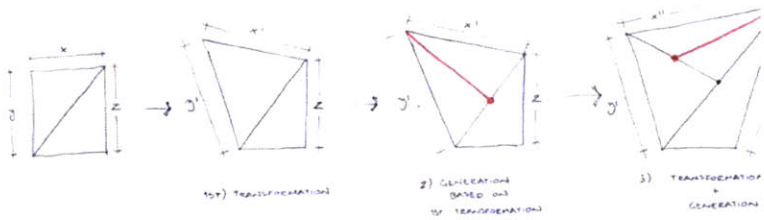
2.53 "Stretchable Surfaces".
Geometry drawn on top of the surface. Entities accessible just by double-clicking and dragging.

This is why I describe the system as a semi-automated design system. Firstly, a system responds to any contextual variations in a parametric preprogrammed "automatic" way. Secondly, the system allows direct human intervention. In this sense the system is "smart" enough to let the designer know the specific area in which he / she is permitted to modify the design [figure 2.53]. It delimits in a intelligent way the decisions of the designer.

232.3 Generative Emergence Rules

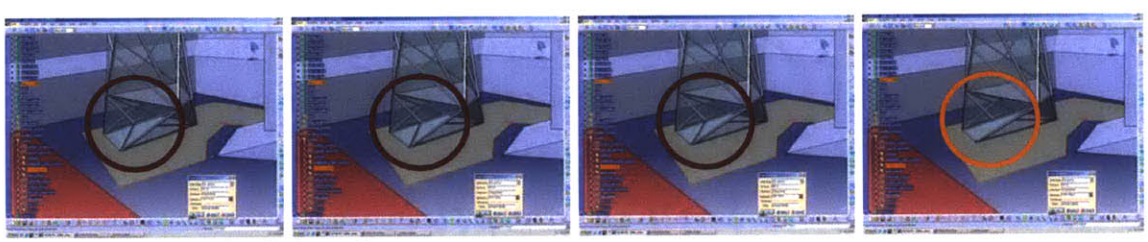
Finally, in this section I will introduce the concept of "emergence" through the use of rules. As I said before, CATIA like most drafting platforms presents a subsection within the software where the users can customize the tool by means of scripting. Through the implementation of rules I am intending to embed "structural intelligence" to the pavilion. This is a pretext to demonstrate the capabilities of CATIA not just as a parametric tool but also as a generative platform.

As stated in the design intentions, the emergence rule is simple. In this case the rule identifies the length of two sides of the basic composition of the pavilion's main structure. If that length is exceeded the system will provide an emergent structural element to "fix" the supposed structural instability [figure 2.54].

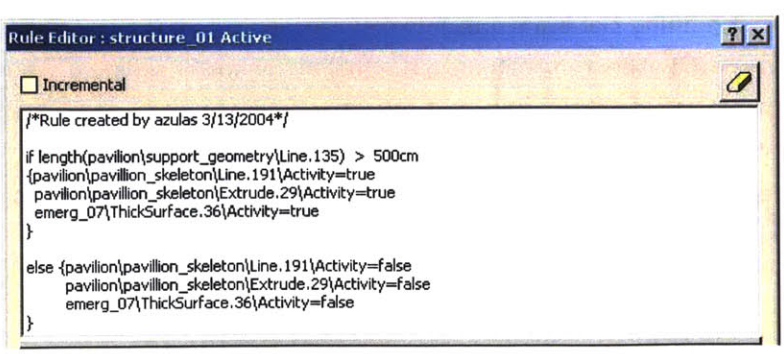


2.54 "Emergence" Rule.
Pavilion skin configuration close-up. These sketches show hypothetical deformation of the structure and execution of the "self-fixing" mechanism.

The emergence rule has been conceptualized to be compatible with the previous responsive systems, activated simultaneously when needed. It can be triggered when the pavilion responds automatically to contextual changes, while following the parametric net, and also when direct changes to the pavilion's geometry are applied when the designer manipulates the building [figure 2.55].



2.55 "Emergence" Rule in Action.
In the sequence above the low bar is decreasing lengthwise (dragged by the designer), and whenever the system recognizes that is "safe" to suppress the extra structural element, the rule is activated. The script can be seen in figure 2.56



Rule Editor : structure_01 Active

Incremental

```
/*Rule created by azulas 3/13/2004*/

if length(pavilion\support_geometry\Line.135) > 500cm
{pavilion\pavillion_skeleton\Line.191\Activity=true
 pavilion\pavillion_skeleton\Extrude.29\Activity=true
 emerg_07\ThickSurface.36\Activity=true
}

else {pavilion\pavillion_skeleton\Line.191\Activity=false
      pavilion\pavillion_skeleton\Extrude.29\Activity=false
      emerg_07\ThickSurface.36\Activity=false
}
```

2.56 The Script
The rule basically turns ON and OFF the emergent structural element.
As shown in the small script, whenever the horizontal element defining the façade exceeds 500cm the diagonal emergent member is activated. Likewise, when that dimension is less than 500cm the diagonal emergent member is turned OFF.

This proves that we can rely on the computer's automation if the necessary data has been fed into it. I would argue though, that this is possible only when the designer has a clear image of the expected behavior of the product.

233. Exploration Reflections

In my quest of designing responsive / adaptable platforms and after playing with the three main interrelated computational concepts experimented in this CATIA exercise:

- Parametric representation of context.
- Parametric delimited action: Semi-automated human intervention.
- Generative emergence rules.

I can conclude that:

1. If the design objectives are laid out in a systematic way it is possible to translate them into a series of sequential steps, therefore they are programmable in terms of computation.

2. It is possible by using CATIA to conceive a synthetic universe formed by more than one design system. It has been demonstrated that several "responsive" design platforms can be embedded into the same virtual world, systems depending on each other but each keeping their own independent characteristics. By these means, more versatility and flexibility is imprinted into the platform. I would associate this computational system to a set of mechanical gears working all together.

3. This analogy leads me to my last point. Recursively embedding simple systems into other simple systems, just like the interlinked mechanical gears, can increase complexity by exponential factors, causing eventual limitations such as decreasing efficiency regarding data-processing time.

003. Conclusions

This paper presents a series of alternative ways in which the idea of adaptability can be achieved through computational means. I have explored the ways in which it is possible to embed the concepts of sensorial and responsive intelligence into an architectural context. My vehicle for understanding the "embedment" of sensorial and responsive intelligence has been through the exploration of different computational platforms. Through the development of a series of design experiments, in which the common research line was designing responsive / adaptive objects, I argue that for designers, who usually have a stronger visual education and understanding, programming with geometrical shapes is more convenient. I can state in that sense that CATIA is an appropriate platform in which it is possible to perform operations, conditionals, and geometric relationships through the direct manipulation of the representation of objects or shapes.

As a side effect of using computational platforms as part the design process, I can state that the same organized-logical nature of computer programming generates a more methodological and systematic approach of conceiving architectural objects.

After these explorations I can also conclude that it is possible, as I have pointed out, to create a computational universe in which several responsive systems can be interconnected. This results in a more complete, flexible and powerful adaptive / responsive platform. However, this overlapping of systems ultimately generates a platform in which complexity scales up exponentially, resulting paradoxically in quite restrictive limitations.

This research on architectural adaptability focuses in a "large scale grain," where a general methodology has been "virtually" laid out establishing external and internal forces as examples of design drivers mapping out the impact on specific

architectural elements. Based on this I envision two main branches in which the idea of adaptation can be further developed:

4. Through the "grain refinement" of the conceptual implementation of this idea. Modeling computationally a specific real-life condition of a building. Jumping from the big-scale (building) into a smaller-scale (detail). Taking into consideration certain levels of measurement tolerances and physical-material constraints.
5. The translation of this "virtual" achievement into a physical reality. This project could evolve into research focusing on the design of a malleable architectural "tissue", capable of accommodating physical expansion and contractions while responding to specific conditions. Looking into material science and nanotechnologies in search for potential answers and mutual feedback regarding the conception and implementation of "smart" materials.

004. Bibliography

1. **Apollonio, Umbro**; ed. 2001. *Futurists Manifestos*. Boston: MFA Publications.
2. **Berger, John**. 1997. *Ways of Seen*. New York: Penguin Putnam Inc.
3. **Calenbuhr, Vera and Alexander Mikhailov** 2002. *From Cells to Societies: Models of Complex Coherent Action*. Berlin: Springer Verlag.
4. **Dollens, Dennis** 2002. *D2A: Digital to Analog*. Santa Fe: Lumen Press, Inc.
5. **Flagge, Herzog-Loibil and Meseure**; ed. 2002. *Thomas Herzog: Architecture and Technology*. New York: Prestel.
6. **Fox, Michael A**. 1996. *Novel Affordances of Computation to the Design Process of Kinetic Structures.* Cambridge, MA: MIT.
7. **Frazer, John**. 1995. *An Evolutionary Architecture*. London: Architectural Association.
8. **Friedman, Mildred**; ed. 2002. *Gehry Talks: Architecture + Process*. New York: Universe Publishing.
9. **Gans, Deborah and Kuz, Zehra** 2003. *The Organic Approach to Architecture*. Great Britain: Wiley-Academy.
10. **Heaton, Kelly B**. 2000. *Physical Pixels*. Cambridge, MA: MIT.
11. **Hornby, A. S**. 1995. *Oxford Advanced Learner's Dictionary*. Oxford: Oxford University Press.
12. **Jormakka, Kari**. 2002. *Flying Dutchmen: Motion in Architecture*. Basel, Switzerland: Birkhäuser.
13. **Knudsen, Jonathan**. 1999. *Java™ 2D Graphics.* Sebastopol, CA: O'Reilly & Associates, Inc.
14. **Loukissas, Yanni**. 2003. *Rulebuilding: Exploring Design Worlds through End-User Programming.* Cambridge, MA: MIT.
15. **McCarter, Robert**. 1987. "Escape from the Revolving Door: Architecture and the Machine." *Pamphlet Architecture* 12: 6-13.
16. **McMahon, Thomas and Tyler Bonner, John** 1983. *On Size and Life*. New York: Scientific American Books, Inc.

17. **Palumbo, Maria Luisa**. 2000. *New Wombs: Electronic Bodies and Architectural Disorders*. Basel, Switzerland: Birkhäuser.
18. **Pawley, Martin**. 1990. *Theory and Design in the Second Machine Age*. Great Britain: Butler and Tanner Ltd, Frome, Somerset.
19. **Pearce, Peter**. 1990. *Structure in Nature Is a Strategy for Design*. Cambridge: The MIT Press.
20. **Pearson, David**. 2001. *New Organic Architecture: The Breaking Wave*. Berkeley and Los Angeles: University of California Press.
21. **Thompson, D'Arcy W**. 1992. *On Growth and Form*. New York: Dover Publication, INC.
22. **Watanabe, Makoto Sei**. 2002. *Induction Design: A Method for Evolutionary Design*. Basel, Switzerland: Birkhäuser.
23. **Yeh, Bryant P**. 1998. *Kinetic Wall: An Exploration into Dynamic Structure.* Cambridge, MA: MIT.
24. **Zuk, William**. 1970. *Kinetic Architecture.* New York: Van Nostrand Reinhold.

Documentaries:

25. *The Shape of Things*, Produced for NOVA by Peace River Films, 60 min., WGBH, 1985, videocassette.
26. *Death by Design*, A film by Peter Friedman and Jean-François Brunet, 70 min., First Run Features, 1996, videocassette.

Internet:

27. "PageWise, Inc." *Why do chameleons change colors?* 2001. <http://www.allsands.com/Science/chameleonschang_rft_gn.htm> (19 March 2004).