An Exploration into Biomimicry and its Application in Digital & Parametric [Architectural] Design

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Master of Architecture
in
Architecture

Waterloo, Ontario, Canada, 2006

Author's Declaration for Electronic Submission of a Thesis

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

Biomimicry is an applied science that derives inspiration for solutions to human problems through the study of natural designs, systems and processes. This thesis represents an investigation into biomimicry and includes the development of a design method based on biomimetic principles that is applied to the design of curved building surfaces whose derived integral structure lends itself to ease of manufacture and construction.

Three design concepts are produced that utilize a selection of natural principles of design outlined in the initial biomimetic investigation. The first design visualizes the human genome as a template on which the process of architectural design and construction can be paralleled. This approach utilizes an organizational structure for design instructions, the adherence to an economy of means, and a holistic linking of all aspects of a design characteristic of the genetic parallel. The advancement of the first design concept is illustrated through the use of a particular form of parametric design software known as GenerativeComponents. The second design concept applies the biomimetic design approach outlined in concept one to the development of ruled surfaces with an integral structure in the form of developable flat sheets. The final concept documents the creation of arbitrary curved surfaces consisting of an integral reinforcing structure in the form of folded sheet chevrons.

Acknowledgements

I would like to express my gratitude to everyone who has been by my side throughout the development of this thesis as well as my architectural training up to this point. To my family and friends for their support, understanding and constant presence in my life.

I am indebted to those on my advisory panel including Philip Beesley, Michael Elmitt, and supervisor, Thomas Seebohm for their time and effort. Their knowledge and sensitivity to the practice of architecture have provided unique insights and a constant frame of reference with which to create this work. To my external reader, Mark Burry, for taking time out of his busy schedule to lend his expertise and wealth of experience to my thesis defense.

My thanks go out to Robert Aish, Rob Woodbury, and all the members of the GenerativeComponents design team and Smart Geometry group for allowing me to be involved in the ongoing development of parametric design software as well as the knowledge gained in my time spent at the workshop and conference in London and Cambridge, UK and the Subtle Technologies workshop in Toronto.

For my father

Contents

Appendix

Illustrations

Chapter 1

Chapter 2

15. Human Femur. (http://biomech.me.unr.edu/hip.htm)

16. Scallop Shell. (www.bmyersphoto.com/BWXRAY/animals43.html)

17. Snail Shell. (http://images.google.ca/images)

18. Tardigrade. (http://images.google.ca/images)

19. Sunflower. (http://images.google.ca/images)

20. Shell. (http://images.google.ca/images)

21. Spittlebug cocoon. (http://images.google.ca/images)

22. Ant nest. (http://images.google.ca/images)

23. Weaverbird Nest. (http://images.google.ca/images)

24. Spiderweb. (http://images.google.ca/images)

25. Termite tower. (http://images.google.ca/images)

26. Plant leaf. (http://images.google.ca/images)

27. Dragonfly wing. (http://images.google.ca/images)

28. Cactus. (http://images.google.ca/images)

29. Surface Tension in Cylindrical and Spherical Vessels.
(http://hyperphysics.phy-astr.gsu.edu/hbase/ptens.html#lap)

30. Relationship between radius and tension.
(http://hyperphysics.phy-astr.gsu.edu/hbase/ptens.html#lap)

31. Human pelvis. (http://biology.clc.uc.edu/Bone_Features.html)

Chapter 3

Chapter 4

Chapter 6

Note: All diagrams in Chapter 6 by author except as noted.

Preface

Architecture through its very nature is heavily involved in the development and integration of two key aspects of the built environment, those being form and function. For centuries, the dominant form of structure has been strongly influenced by the current technology available in the construction and manufacturing industry. With the proliferation of mass production and the development of the assembly line it became possible to create a construction industry based on discrete building assemblies and materials that serve to benefit a faster and easier method of raising structures. This increase of speed and relative ease of design due to unitization and standardization has come at the cost of maximal structural efficiency, minimization of materials and a relative compensatory need to artificially regulate the interior building environment. Recent advances in computer modeling and systems testing have allowed the architect to improve upon all of these aforementioned building variables. However, without a first principles approach to design that questions the validity of the structures and systems to which these new technologies of design and testing are applied, the whole process becomes burdened with an inefficiency that will always be inherent. The simple reason of advancement in a particular field is not an a priori reason for believing that the direction that field is going in will yield the most profound and bountiful results.

Like languages, architecture is a discipline that will always comprise a number of variations that are characteristic of the people, social and geographic climate that they serve. While this may be true, there is an underlying basis by which all of these variations may be linked together whether through a biological necessity to communicate with each other, as with language, or a similar biological desire for shelter. It is important to note here that each variant has both benefits and detractions as compared to its siblings. With architecture a number of intellectual and design philosophies have developed through time with some that remain and others that fall out of favour. For any object or idea to endure and in effect become timeless it must pass through a number of filters that measure its clarity and depth. If the characteristics derived are deemed valuable then what remains is a base that can be built upon and ultimately give rise to progeny that, while unique unto themselves, still retain the genetic makeup from which they stemmed.

In nature this has been well documented through the works of pioneers in the field of biology and evolution. Over many millennia the organisms that inhabit this planet have gone through countless environmental filters that have shaped and continue to inform the shape of organisms today. From early iterations to today's counterparts the wealth of biological diversity is staggering and is testament to the earth's testing ground. As supremely motivated and inquisitive creatures, gained from our ancestors. This intellectual base is constantly refined and rethought in an effort to sift through what is deemed unnecessary and excess and arrive at a new level of understanding and ability. Nature has provided this framework of constant improvement for us and it is this feature that is the basis for this thesis. The principle of biomimetics strives to learn how nature has learned and to not necessarily imitate but distill from nature the qualities and characteristics of natural form and systems that may be applicable to our interpretation of architecture.

My interest in the correlation between architecture and biology first developed during my time at McMaster University where I completed a Bachelor of Science specializing in biology. The knowledge gained in the area of genetics and biological form prompted an inquisition into the relevance of nature's method of design and construction with regard to human constructions.

1.0 Introduction

BIO–MIMICRY [From the Greek *bios*, life, and *mimesis*, imitation] (Benyus 1997)

The emulation or imitation of natural forms, structures and systems [in design and construction] that have proven to be optimized in terms of efficiency as a means to an end.

A biomimetic approach to design, while emulating natural systems, derives its solutions through the utilization of a design process that seeks to satisfy the core requisites of a design in a holistic manner. This approach avoids a sequential component design process and attempts to develop the design products in a concurrent manner whereby necessary changes that occur in the development of a particular design component will be propagated throughout the entire design to minimize repercussions for the realization of alternate design iterations.

This thesis begins with an investigation into Biomimicry as a new field of study that is applicable to a wide variety of disciplines. An examination of key principles of natural design relevant to the focus of the thesis will create a lens through which it will be possible to focus on design and manufacturing techniques that are appropriate to biomimetic design. A number of questions related to current deficiencies in design and construction methodologies will be asked in an effort to generate a set of answers that will aid in defining what objectives are to be met in the thesis and the direction by which they will be attained.

The aim of this thesis is to develop an innovative way in which to create curvilinear structural designs through a combination of the biomimetic principles of design that relate to and inform the process of digital and parametric design. The desire, in its realization, is to reduce the complexity of both design and construction in a manner that reduces the amount of instructions, documentation and visualization necessary to produce architectural works.

The design portion of the thesis will concentrate on creating three design concepts that will be developed based on varying levels of granularity with respect to the scope of biomimetic design in architecture. The purpose of this investigation is to begin with a broad interpretation of design, manufacturing and construction as it is today and propose a direction, based on the natural development of organisms, that could lead to a more efficient way in which to produce architectural works.

Based on the design methodology put forth in the first concept it will be possible to develop prototype design concepts that utilize the principles of natural design and construction.

This thesis does not deal with the cultural implications of what the formal physical appearance of a holistically designed architecture based on biomimetic principles should be or what cultural values it should reflect. Curvilinear architectural forms are often referred to as being organic or reflective of organic design principles and as such, a cultural layer, vis a vis nature, is applied to them.This thesis takes no position on the cultural significance of curvilinear architecture but focuses on this form of architecture because it is believed that the biomimetic principles of design proposed in the thesis are a significant improvement over current design approaches to such forms of architecture.

1.1 Introduction to Biomimetics

While Buckminster Fuller is often attributed with the early incarnations, it is Janine Benyus, a science writer and lecturer on the environment, who is responsible for the recent codification of Biomimicry as a field of research and study. Her 1997 book entitled *Biomimicry: Innovation Inspired by Nature* brought together the recent discoveries in a multitude of disciplines, from engineering to agriculture, that can be traced to research and investigations into the designs and processes found in nature. A number of propositions are put forth in the book that effectively illustrate the current trends and principles of Biomimetic investigation.

1. **Nature as Model** – Biomimicry is a science that studies nature's models and emulates or takes inspiration from their designs and processes to solve human problems.
2. **Nature as Measure** – Biomimicry uses an ecological standard to judge the 'rightness' of our innovations. After 3.8 billion years of evolution, nature has learned: What works. What is appropriate. What lasts.
3. **Nature as Mentor** – Biomimicry is a holistic way of viewing and valuing nature. It introduces an era based not on what we can extract from the natural world, but on what we can learn from it. (Benyus 1997)

Although its formal introduction as a scientific discipline has been relatively recent, the principles and directives inherent in Biomimetics as they relate to architecture are derived in part from a long line of contributors within a variety of biological and architectural streams.

From a historical standpoint the term biomimetics was introduced in the 1950s by Otto Schmitt, an American inventor, engineer and biophysicist who was responsible for developing the field of biophysics and founding the field of biomedical engineering.

Predating the work of Otto Schmitt is that of D'Arcy Thompson, an eminent biologist and mathematician who released his book entitled *On Growth and Form* in 1917. This incredible collection of work was instantly recognized for its originality and depth of scope. Often touted as "the first biomathematician" it was Thompson who suggested that the influences of physics and mechanics on the development of form and structure in organisms were underemphasized. His book sought to illustrate the connection between biological and mechanical forms. Thompson's book does not attempt to posit any type of discovery pervasive to all of biology, nor does he propose a causal relationship between emerging forms in engineering with similar forms in nature. His book presents a descriptive catalog of natural forms and the mathematics that define them. Since its release, the book has served as a wealth of inspiration for biologists, architects, artists and mathematicians. (O'Connor 2006)

"No organic forms exist save such are in conformity with physical and mathematical laws...

The form, then, of any portion of matter, whether it be living or dead, and the changes of form which are apparent in its movements and in its growth, may in all cases be described as due to the action of force. In short, the form of an object is a 'diagram of forces'." (Thompson 1963, p11)

The following forms of architectural design vary with regard to their adherence to a strict definition of biomimicry yet they all share a desire to derive architectural incentive from nature.

Organic Architecture – "…exalting the simple laws of common sense—or of super-sense if you prefer—determining form by way of the nature of materials..." (Wright 1939)

Evolutionary Architecture – "…an all-encompassing applied philosophy based upon the profound study of nature's processes, organisms, structures and materials at a multitude of levels, from sub atomic particles to the kinesiology of insect and animal anatomy, to the ecological relationships of living habitats, and then applies this knowledge to the design and construction of our built environment." (Tsui 2000)

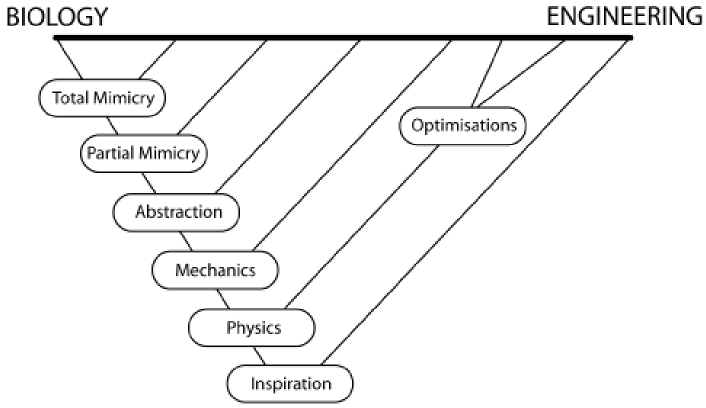
Anthroposophic Architecture – "…which seeks to respond to the human form and human needs [where] buildings should appear in harmony with the landscape in which they are built, with regard to both form and material." (Pearson 2001, p5)

Biomimetics goes further in that it strives to unify the knowledge contained within a diverse field of scientific disciplines into one cohesive unit. This approach to design is seen as an integrated network that is dependent upon a feedback system related to the key factors in design. These factors which comprise all of the relevant external and internal forces that can influence a design from occupancy, loading, seismic, HVAC to daylighting inform the direction of the design and interact with one another to create the final solution.

'The attraction of Biomimetics for architects is that it raises the prospect of closer integration of form and function [with regard to a holistic building design]. It promises to yield new means by which buildings respond to, and interact with, their users - means more subtle and more satisfying than present mechanical systems. At a deeper level, according to George Jeronimidis of the University of Reading, architects are drawn to the field 'because we are all part of the same biology'. The urge to build in closer sympathy with Nature is, he believes, a genuinely biological, and not merely a Romantic, urge.' (Aldersey-Williams 2003, p169)

In this thesis, function is seen as co-evolving with the development of form in that each exert an influence on one another. A desired shape (form) may be created and a structural system (function) derived from it, however, the requirements of the structural system may influence and require subsequent changes in the form. A feedback exists between form and function where the varying conformational possibilities of a design will lead to unique structural adaptations specific to that form.

The appeal of biomimetics stems not merely from a method for acquiring abstract design ideas from nature but also from the manner in which nature utilizes those ideas. Common to both natural and man-made environments is the issue of cost. There is always an issue of how much an object, structure, or organism will cost to design, manufacture, construct, maintain and ultimately recycle. In an architectural sense this can be reduced to a monetary cost where often times the lowest tender wins. In the natural world the cost is energy, where competition for available resources favors the organism that can survive and grow with the least amount of required materials and energy expenditure. Animals must fight for territory, sex, and food while plants develop innovative ways to harness more sunlight than their neighbors. In simple terms it can be proposed that the organism which survives best is the one that produces more viable offspring per unit of expended energy than its competitors. Similarly, an architect must balance a number of design variables that equate to the investment of cost which may be structure, appearance, efficiency, or any other number of requirements. The design that offers the best product for the least amount of investment will often be the one that is produced. It is worth noting however that the design capabilities, materials, manufacturing and construction methods we as designers have in our palette are different from those found in nature, and as such do not always translate from one to another in an efficient manner. Thus, a concept will become much more robust if we are able to distill innovative design and manufacturing inspiration (with regard to the current manufacturing techniques available) from natural phenomena rather than strictly attempting to mimic them. (Vincent 2002, p4) See Figure 1.



1. Map of biomimetic processes.

1.2 Direct Approach to Biomimetic Investigation

A direct method of investigation actively seeks to define the nature of the design problem and the context of its creation and use. With a clear understanding of the design requirements it is then possible to look to the natural world for examples that fulfill them. It is useful to investigate an array of divergent organisms that rely on different approaches to solve similar problems. This will yield a greater variety of ideas with which to develop. Structural solutions, for example, do not rest solely in mammalian bone but can be found in the composition of wood, the shell of an arthropod, the exoskeleton of an insect or in an individual plant leaf. Unique solutions can develop from a wide variety of inspirations.

1.3 Indirect Approach to Biomimetic Investigation

An indirect method of investigation seeks to find solutions through defining the general principles of natural design and using those as guidelines for developmental progression. While it is difficult to effectively categorize the entire collection of natural designs into discrete units there arise recurring principles, as described below, that have been observed which form a coherent strategy for investigation.

12 Methods by Which Nature Can Inform the Development of Technology: (Benyus 2004)

1. **Self Assembly** – The ability of an organism to direct its own process of development.

2. **Chemistry in Water** – Nature produces all of its compounds in normal environmental conditions without a necessity for extreme temperatures or harsh chemicals.

3. **Solar Transformations** – Many organisms respond actively to the sun to maximize their energy absorption.

4. **The Power of Shape** – Nature uses many structurally efficient non-orthogonal forms with which to create its structures.

5. **Materials as Systems** – Nature builds from small to large with a corresponding scaling of function in relation to the materials and components involved for particular functions.

6. **Natural selection as an innovative engine** – Environmental forces that act on an organism and affect its fitness will direct the development of future organisms.

7. **Material Recycling** – Create structures using materials that are non-toxic and can be fully recycled at the end of their life.

8. **Ecosystems that Grow Food** – Systems are created that have a net surplus of production without a corresponding draw-down of environmental resources.

9. **Energy savvy movement and transport** – Locomotion and internal circulation systems have adapted to require a minimal investment of energy for their purpose.

10. **Resilience and Healing** – Living organisms have the ability to absorb and rebound from impacts and can repair themselves if damage is incurred.

11. **Sensing and Responding** – A series of feedback systems within an organism allow it to sense a variety of environmen-

tal factors acting on it and to respond to these in a suitable manner.

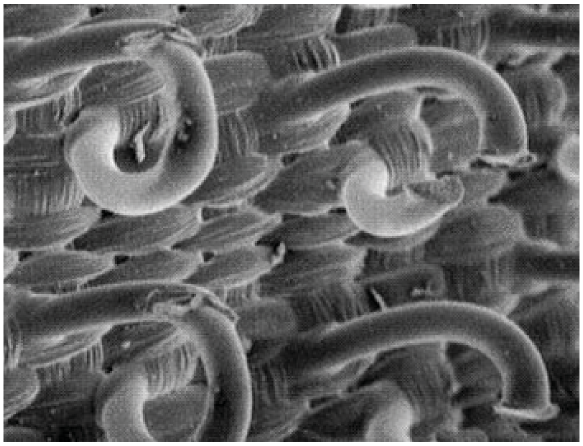
12. **Life creates conditions conducive to life** – The waste products and various by-products of growth and sustenance create materials that are beneficial to the growth of other organisms.

1.4 Biomimetic Solutions in Other Design Disciplines:

Man-made designs throughout history have been realized through observations and investigations into the natural world, albeit on varying degrees from imitation to inspiration. From the creations of Leonardo DaVinci, including his flying wing, to the present day work with nanotechnology, a variety of disciplines have realized the potential source of design inspiration that nature has. The following examples provide a brief list of areas where biomimetic influences can be found. (Vogel 1998, p276-279)

1. **Streamlined bodies** – The study of aquatic organisms led to advances in the development of streamlined shapes in technology. Like the trout or dolphin a body that travels through the air or water experiences least resistance if it is rounded in the front and tapers to a rear point.

2. **Airfoils** – Bird wings have curved tops and flatter bottoms. This aerodynamic shape is essential to provide lift for aircraft wings.



2. Rounded pleats of automobile air filter inspired from a dolphin's nose.
3. Pultrusion machine for carbon fiber.
4. High magnification of Velcro hooks.

3. **Maneuverability of Aircraft** – Upon observing the flight of buzzards the Wright Brothers determined that they regain their lateral balance when partially overturned by a gust of wind by torsion of the tips of their wings. This discovery prompted the development of ailerons that control the banking movement of the airplane which cause it to turn.

4. **Extruded fibers** – Silkworms and spiders. Extruded fibers such as carbon fiber are developed from the principles learned from these creatures. While the process of formation is not identical the theory behind the technology was established through their investigation.

5. **Telephone transducers** – Emulations of the components in an eardrum.

6. **Velcro** – Examination of the barbs on burdock burs.

7. **Drag reduction** – Fish slime and their use of long, linear, soluble polymers.

8. **Peristaltic pumps** – The intestines of many organisms move fluids through peristaltic action. In industry, peristaltic pumps use rotating rollers pressed against special flexible tubing to create a pressurized flow. The tube is compressed at a number of points in contact with the rollers or shoes. The media is moved through the tube with each rotating motion. Moving parts do not come in contact with the

2.0 Exploration of Biomimetic Design Principles

The natural world does not consciously organize itself based on singular and separate approaches to solve the twelve methods of design outlined in *Section 1.2.* Rather, its designs develop through an interdependency of each design method to arrive at a final product. While this approach would be ideal in the creation of man-made designs we must first delve into the unique characteristics and contribution to design that each holds before we can endeavor to formulate an efficient solution that encompasses them. The desired outcome for this thesis, being the development of a more efficient and streamlined overall approach to design and construction and specifically the use of natural design in the creation of non-orthogonal structurally supportive building skins, relies on a selection of five designs methods outlined in *Section 1.2.* The following subset of imperatives were chosen for their relevance to structure and design process at it relates to the development of the thesis. It should be noted however, that the further development of the thesis outcome need not be limited strictly to a subset of the design methods but could with further research grow to encompass all of them.

2.1 Self Assembly:

2.1.1 DNA and Genetic Coding:

'Theoreticians fiercely contest the precise relationship of morphogenesis to genetic coding, but there is an argument that it is not the form of the organism that is genetically encoded but rather the process of self-generation of the form within an environment. Geometry has a subtle role in morphogenesis. It is necessary to think of the geometry of a biological or computational form not only as the description of the fully developed form, but also the set of boundary constraints that act as a local organizing principle in the self-organization during morphogenesis.' (Weinstock 2004, p14)

Nature has adapted the plans from which it derives organisms to be based on a relatively simple set of instructions. The fertilized egg of a human or similar animal has approximately 10^{10} bits of information in its DNA that are responsible for the plan of the organism. A human is composed of around 10^{14} cells which is a magnitude of 10,000 times greater than the number of instructions contained within the egg. With the onset of computer aided design and 3D modeling we have come to realize that with every additional layer of complexity we introduce into a model there is a corresponding increase in file size and processing time. Organisms in the same way are three-dimensional and as a result should require a vastly greater amount of information for morphogenesis to take place than is available in the cell. From this it can be said that the form of an organism must be derived from a relatively unresolved set of plans. (Vogel 1998, p25)

'To a remarkable extent the dazzling diversity in nature represents superficial features of systems of an exceedingly conservative and stereotypical character' (Vogel 1998, p31)

The relative lack of information clearly underlies a lot of biological design. In 1950 an eminent physicist, Horace R. Crane, predicted that many subcellular structures would turn out to be helical in form, not because helices necessarily worked best but because they could be assembled with especially simple instructions. Crane anticipated not only the double helix of DNA but its supercoiling, the so called alpha helix of parts of many proteins, and, on a larger scale, helical **microtubules** and **microfilaments** important in maintaining the shape and motility of cells. Microtubules and microfilaments have a remarkable capacity for self-assembly; if all the components are put together (with perhaps a bit of the formed structure as a starter) they ordinarily fall into place without any need for mold of scaffolding or, more important, for any additional information. (Vogel 1998, p26)

Building large organisms out of many cells is probably made necessary by that shortage of information. Cells may look diverse, but they all have a lot in common; if you can build one kind, you need only a little more information, relatively speaking, to build all the others. Furthermore, in the development of each individual, one group of instructions can set more than one structure. In humans, hand size is an excellent predictor of foot size. Bilateral symmetry is an efficient method by which the number of instructions required to derive a developed form is essentially halved. A single alteration of the genetic material – a mutation – ordinarily affects both sides of the body of an animal. The heart and lungs of all of us are in the same position but at some level of detail the locations of our parts are unpredictable. Anatomy students learn the names of the large blood vessels, but the small ones stay anonymous – simply because their arrangement varies from one person to the next. (Vogel 1998, p27)

2.1.2 Self Assembly in Nature:

Nature uses the process of self-assembly as the fundamental principle which generates structural organization on all scales from molecules to galaxies. It is defined as a process whereby pre-existing parts or disordered components of a pre-existing system form structures of patterns. Self-assembly can be classified as either static or dynamic. Static self-assembly is an ordered state that occurs when the system is in equilibrium and does not dissipate energy. Dynamic self-assembly is when the ordered state requires dissipation of energy. Examples of self-assembling system include weather patterns, solar systems, histogenesis (the formation and development of tissues) and self-assembled monolayers (monomolecular films).

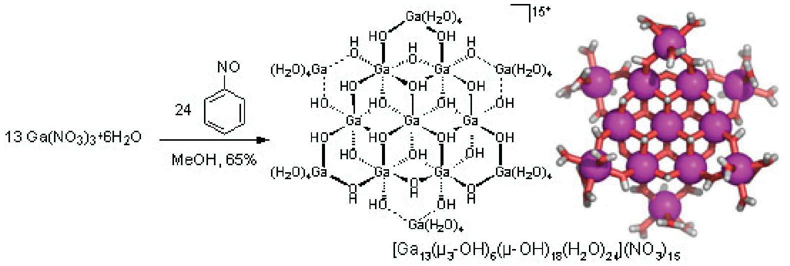
2.1.3 Molecular self-assembly:

Molecular self-assembly is the assembly of molecules without guidance or management from an outside source.There are two types of self-assembly, intramolecular self-assembly and intermolecular self-assembly. Intramolecular self-assembling molecules are often complex polymers (primary structure) with the ability to assemble from the random coil conformation into a well-defined stable structure (secondary and tertiary structure). An example of intramolecular self-assembly is protein folding. Intermolecular self-assembly is the ability of molecules to form supramolecular assemblies (quaternary structure).

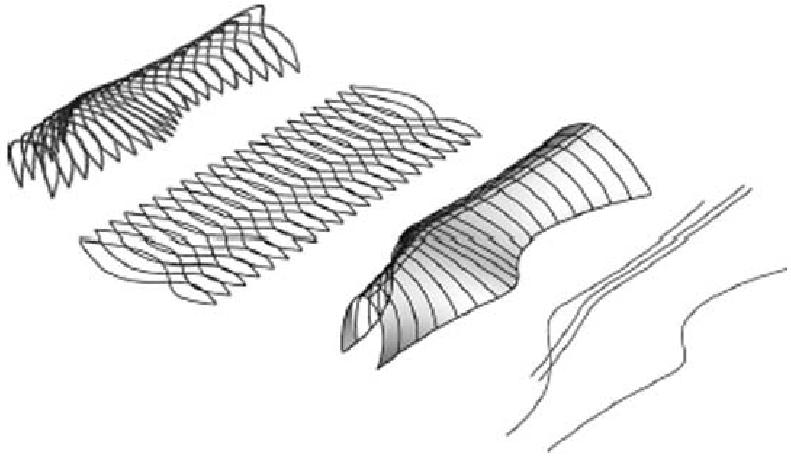
Self-assembly can occur spontaneously in nature, for example in cells (such as the self-assembly of the lipid bilayer membrane) and other biological systems. See Figure 5. It results in the increase in internal organization of the system. Many biological systems use self-assembly to assemble various molecules and structures. Imitating these strategies and creating novel molecules with the ability to self-assemble into supramolecular assemblies is an important technique in nanotechnology. (Whitesides 2002, p2418-21)

2.1.4 Structural Development

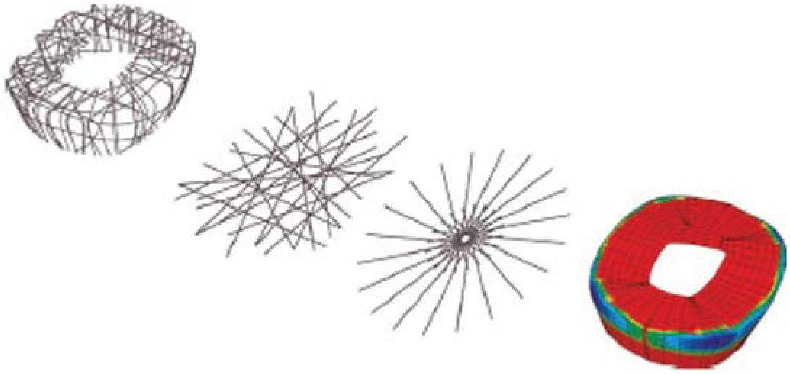
Patterns – "The interest in patterns is primary in that they are essential to the structural framework of natural and artificial systems. We can no longer reduce things to singular elements but instead see that everything



5. Self assembly of inorganic nanoclusters.



6. Process illustrating the evolution from path to surface, and pattern to structure.



7. Structural analysis of shell comprised of radial and random patterns.

is made up of a series of interrelated parts that perform together as a collective whole. From the cellular structure of living organisms to the networks that make up our connected society, patterns are always the agents that allow the total assembly to evolve and adapt to a changing environment… Traditionally, structural patterns are defined in Cartesian space and require prescribed repetition and a high degree of redundancy for structural integrity. By pursuing a reconfiguration of component relationships which reveal themselves in design solutions, forces are dissipated through a system in multiple directions and transferred to the substructures. Structurally patterned modularity is deployed at different scales, in various configurations, with adjustable degrees of density and directionality. See Figure 6. Specifically, it is now possible to see the joint, or point of intersection as a more dynamic aspect in the tectonic definition. No longer bound by identical repetition, the joint must now be capable of providing iterative difference if it is to respond to the surface transformations resulting from the structural and ornamental interplay." (Bell 2004) See Figure 7.

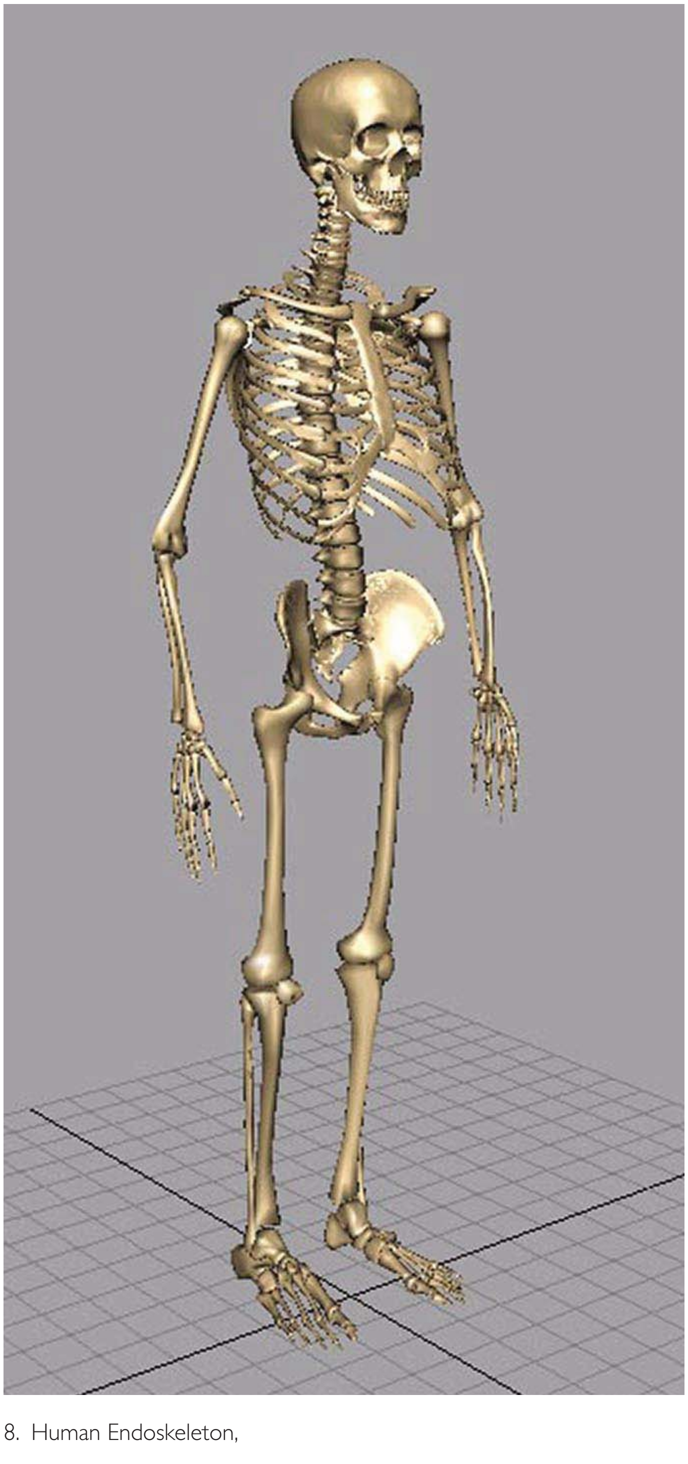
Essentially, the system of a structural hierarchy based on the gradual reduction of individually separate components that is favored today is reinterpreted so that the boundaries between successive structural layers is blurred and the building becomes one indivisible unit from the micro to macro scale. This approach reduces the vulnerability of a building to failure due to localized stresses, as the structural system has built in structural redundancy acting on a number of levels to dissipate localized stresses throughout the entire structure. The patterning that takes place in this method can occur in a variety of configurations from a simple scaled grid shaped layout to a more complex fractal geometry whose forms are identical at a number of scales.

2.1.5 Endoskeletons and Exoskeletons:

Terrestrial organisms must exist in an environment subject to both gravity and atmospheric pressure. Aquatic organisms deal with gravity, although to a lesser extent, as well as water pressure. In order to counteract the forces acting within and on them as well to maintain their form and possible requirement for locomotion and morphological fluidity, organisms must utilize a structural organization that can accommodate the same. The structural system used by the majority of multi-cellular organisms can be classified as belonging to one of two types:

1. **Endoskeletons** (Internal Structure) - Animals with endoskeletons can grow easily because there are no rigid outside boundaries to their bodies.They are vulnerable to wounding from the outside, but repair of the living tissue is usually not a problem. See Figure 8.

2. **Exoskeletons** (External Structure) - Exoskeletons are outside the body and encase it like armor.They are light and very strong, and provide attachment places for the muscles inside. They protect the body from dehydration, predators, and excessive sunlight. See Figure 9.



8. Human Endoskeleton,

9. Crab Exoskeleton.

Sphere					
Surface Area (x^2)	23	36	47	57	66
Volume (x^3)	10	20	30	40	50

Cube					
Surface Area (x^2)	28	44	58	70	81
Volume (x^3)	10	20	30	40	50

10. Surface Area and Volume Correlation for Sphere and Cube.

2.2 The Power of Shape:

2.2.1 Fundamentals of Natural Form

Nature utilizes a variety of forms and design methods in its constructions to ensure maximization in terms of structural efficiency and mobility while minimizing the required input of material.

1. **Maximize structural strength** – Nature employs a relatively small amount of materials in its assemblies as compared to human constructions. However, through unique configurations of these simple materials nature is able to create structures that outperform many man-made structures. (Tsui 1998)

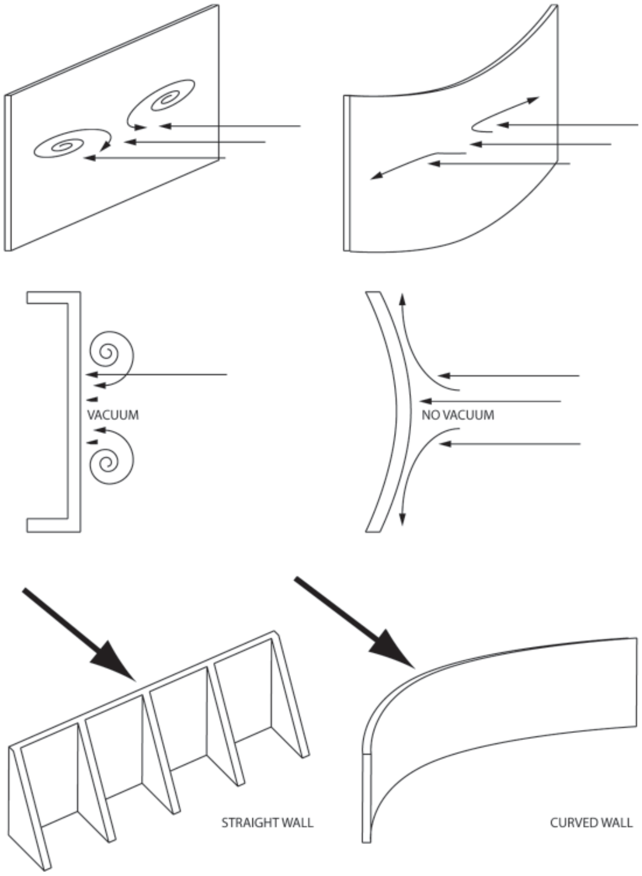
2. **Maximize enclosed volume** – In order to conserve heat organisms must maintain an efficient balance between their surface area and internal volume. Through the use of curvilinear forms nature is able to maximize the internal volume of an organism while minimizing its surface area. See Figure 10. This has the effect of reducing the amount of heat lost across the surface of an organism to a minimum, thus allowing it to remain warmer with less input of energy. Additionally, a smaller surface area results in a requirement for less input of materials to form the organism as well as a reduction in weight. (Tsui 1998)

3. **Create high strength-to-weight ratios** – Since there is competition for material resources within an ecosystem, natural organisms must utilizes unique methods of con-

struction that minimize the input of material and expenditure of energy while maximizing the subsequent strength achieved. Bones in an organism vary their cross section over their length to deposit material where it is most needed. In addition, cross-linking of the fibers in the bone contribute to strength increases without a corresponding increase in weight. (Tsui 1998) See Figure 11.

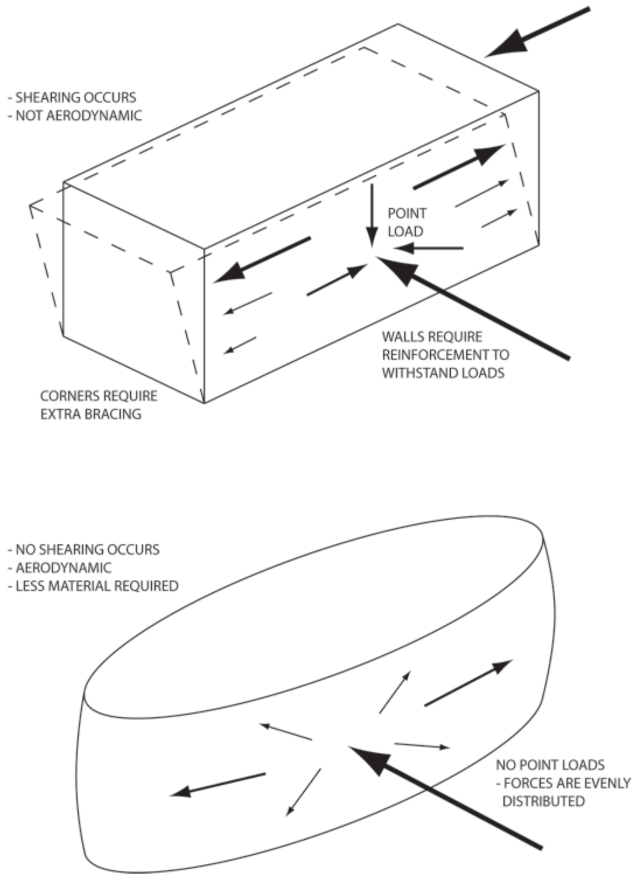
11. Cross-section of Bird Bone.

4. Use stress and strain as a basis for structural efficiency – Natural forms are derived from their varying rates of growth and these three dimensional shapes are dependent on an irregular rate of growth throughout the organism. The external environment exerts stresses on the developing object and its resulting form is a product of its response to the environment and the limits of the structural properties of the material used. This process occurs on both short and long term scales of time where evolution has contributed to the genetic code that defines the growth template while stresses acting on an within the organism shape the final and ongoing form. (Tsui 1998)



12. Effects of Wind and Live Load on Structure.

5. Integrate aerodynamic efficiency with structural form – Many organisms are mobile and as such are subjected to the laws of aerodynamics or hydrodynamics. To effectively inhabit their environment the form of the organism is often tailored to maximum efficiency for the minimal expenditure of energy for locomotion or resistance to environmental stresses such as wind on a tree. Similarly, a curved wall is able to more easily dissipate



13. Effects of Live Load on Structure.

wind load as well as requiring less material in order to do so. (Tsui 1998) See Figure 12.

6. **Curvilinear forms that disperse and dissipate multidirectional forces** – Through the use of curvilinear forms, organisms have the ability to absorb and dissipate loads throughout their structure which helps to reduces areas of collected stress and the need for unnecessary structural reinforcement. (Tsui 1998) See Figure 13.

2.2.2 Forms that Organisms in Nature are Composed of:

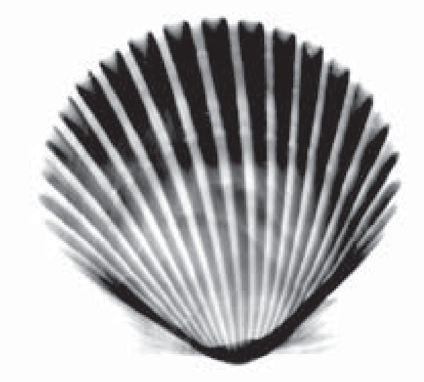
The natural world contains a wide array of organisms that are composed of many different forms and shapes. The variety of intricate forms however, can be thought of as belonging to a set of basic shapes and structures with each organism using them in different proportions. (Tsui 1999, p86-131). See Figures 14-19.

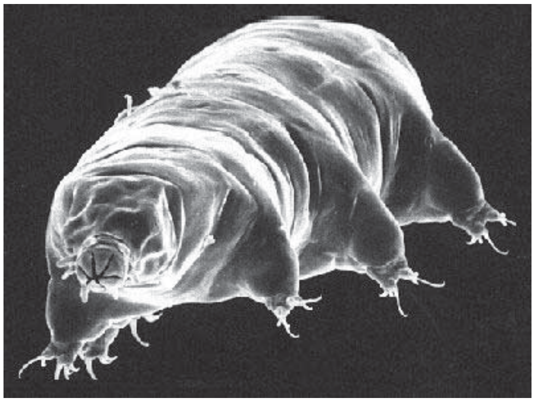
1. **Curved shells** – Skulls, eggs, exoskeletons (domed roofs)
2. **Columns** – Tree trunks, long bones, endoskeletons (posts)
3. **Stones embedded in matrices** – Worm tubes (concrete)
4. **Corrugated structures** – Scallop shells, cactus plants, stiffness without mass (doors, packing boxes, aircraft floors, roofs)
5. **Spirals** – Sunflowers, shells, horns of wild sheep, claws of the canary bird (domed roofs)
6. **Parabolic Forms** – Tardigrade (pneumatic structures)

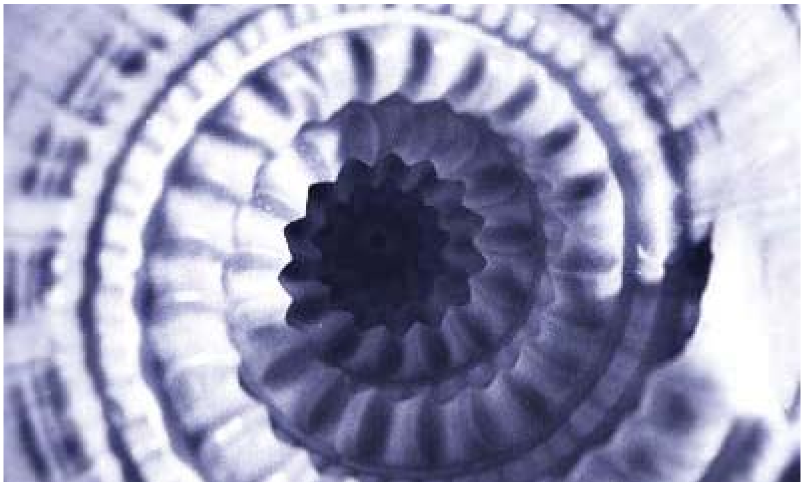
2.2.3 Forms of Structures that Organisms Build:

Many organisms fashion their shelters out of natural material located within their own habitat. Whether produced from found material or as a result of internal production, as with spiders, the variety of forms that organisms construct can also be categorized into a set of recurring forms and principles. (Tsui 1998) See Figures 21-25.

1. **Combined structural shapes and forms** – Termite towers, prairie dog burrows
2. **Parabolic Forms** – Bowerbird nests
3. **Hemisphere/mound forms** – Beaver







From top left. 14. Human skull. 15. Human femur. 16. Scallop shell. 17. Snail shell. 18. Tardigrade. 19. Sunflower, shell.

dams, ant nests,

4. **Tension/membrane structures** – Leaf cutter ant nest, weaver ant nest, silkworms, spider webs
5. **Hemisphere/sphere** – Potter wasp, ovenbird nest, cactus wren nest, spittlebug nest
6. **Egg/bell shapes** – Africa gray tree frog, paper wasp and honeybee nest, weaverbird nest
7. **Tube/cylinder forms** – Swallow tailed swift nest, bagworm case, jawfish, shark and the helix, brine shrimp nest

2.2.4 Flatness:

Advantages of being flat:

1. **Easy to walk on at any point** - An even floor, void of surface deformation, allows ease of circulation at any area on the surface

2. **Utility in a world dominated by gravity** - Gravity allows for rapid construction with regard to the creation of level surfaces as well as in material application where concrete, for example, has the tendency to level itself based on gravity.;

3. **Wall of minimal area that separates two compartments -** A straight wall between adjoining rooms or buildings has the least amount of area requiring surfacing.

Clockwise from top left. 21. Spittlebug cocoon. 22. Ant nest. 23. Weaverbird nest. 24. Spiderweb. 25. Termite tower.

4. **Materials pile smoothly on one another -** Flat and straight materials are efficient because they allow for a regular and maximized arrangement during transport to the site and subsequent storage until ready for use. In terms of construction, flat roofs are easy to build and handy to use. Beams and boards can be laid parallel on top of each other for ease of transportation. Shingling becomes a strictly two-dimensional operation. Simple instructions are required for their assembly.

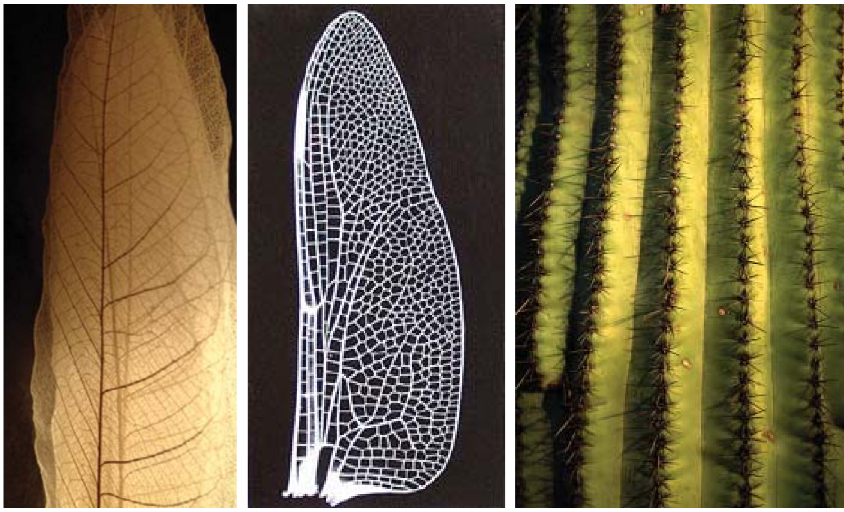
Disadvantages of being flat:

1. **Sag at the center of a horizontal element** – Depending on the size and span requirements of building elements a certain amount of gravitational sag will occur due both to dead and live loading. To prevent sag from occurring, a large amount of material may be required to provide adequate flexural resistance.

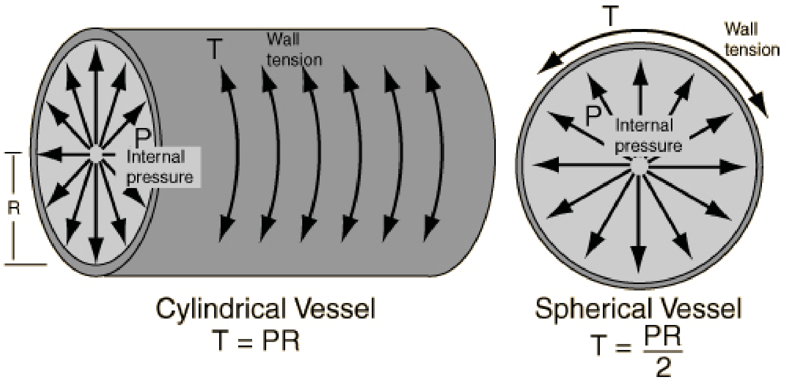
2. **The greater the loading the thicker must be the floor or the horizontal beams that support it -** When the requirement for loading increases in a typical slab and beam scenario it is necessary to increase the depth of either one or both to attain the required strength. This will result in greater floor to floor heights and subsequent material costs or reduced ceiling heights.

3. **Exacts a considerable price paid with regard to weight -** In flat roofs and high rise buildings weight is a major factor in design and the desire is to reduce the loading that occurs cumulatively on the supporting members. A small increase in weight on the top floors and roof of a building will result in a significant increase in loading that the structural members of the lower floors of the building must support. This results in additional material and building costs.

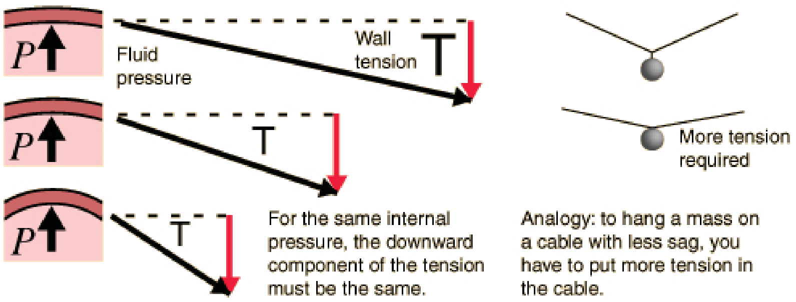
4. **Longer means weaker -** With the requirement for minimal surface deflection to prevent cracks from developing on surface finishes as well as to prevent flex from occurring a beam must meet the structural requirements imposed on it. A longer beam will deflect more and be able to resist less loading than a shorter one. As a result, an increase in span will require either an increase its beam depth or decrease the column to column distance. Both have the effect of increasing material weight and costs.



26. Plant leaf. 27. Dragonfly wing. 28. Cactus.



29. Surface Tension in Cylindrical and Spherical Vessels.



For the same internal pressure, the downward component of the tension must be the same.

Analogy: to hang a mass on a cable with less sag, you have to put more tension in the cable.

30. Relationship between radius and tension.

How nature deals with flatness:

1. **Veins -** Veins increase the functional thickness of leaves with only a little extra investment of material. See Figure 26 & 27.

2. **Curvature -** Without the need for veins, a flat surface can be effectively thickened and stiffened with the introduction of a small amount of curvature.

3. **Pleats -** The introduction of a set of pleats running in the direction in which bending is expected increases the effective thickness without going to the trouble of adding proper beams beneath the surface. See Figure 28.

The wings of an insect comprise only 1% of their body mass. Their structural integrity is derived from a combination of curvature, veins and lengthwise pleats. The key here is the fact that nature, as seen with the insect wing, often combines all three of these methods which can multiply their effects.

Automotive manufacturers discovered the benefits of curvature when the unibody replaced the traditional ladder frame. Pressing a piece of metal into a curved shape is much simpler and uses less material than spot welding stiffener plates to achieve strength. Essentially the central spine of the automobile was removed and replaced by a structural skin. (Vogel 1998, p57-60)

2.2.5 Surfaces:

Pressure and Curvature in a Sphere – When a pressure is exerted either externally or internally on a sphere, a tension is produced in the skin. The tension force is directly related to the size of the sphere. Laplace's Law, which relates internal pressure to surface tension, states that the tension force per unit length of the skin is equal to the pressure times ½ the radius of the sphere. A cylindrical vessel will experience twice the tension in its skin as a spherical vessel. See Figure 29.

A large sphere results in greater surface tension for a given pressure than a smaller sphere. As the radius increases, the curvature of the vessel wall decreases. When the vessel reaches an infinite radius the surface will have an infinite tension. See Figure 30. This fact essentially rules out making balloons, or any other internally pressurized structure, with flat walls. Living organisms usually maintain different internal and external pressures and as such must make efficient use of curvature in their bodily forms to reduce the requirement for their skin to withstand enormous tension forces. Nature avoids flat surfaces wherever possible and stiff domes are the preferred form with uses in eggshells, skulls, nutshells, clamshells, etc.

2.2.6 Angles and Corners:

Right Angles – Throughout human history the presence of right angles in society has been an unfailing signal of cultures with high technical complexity. Nature very rarely uses right

31. Human pelvis. 32. Rounded corners in tree branches. 33. Stress localization and corner cracking.

angles except in bacteria and certain protozoa and foraminifera. Round houses usually indicate a nomadic/semi-nomadic society where curvilinear buildings are more economical of material, less weight and easier to erect. Rectangular houses typify sedentary societies where it is possible to include more buildings in a specified area, the interiors can be partitioned more easily and subsequent additions become easier as well.

34. World Trade Towers.

35. Tree in hurricane conditions.

Corners and Cracks – Humans tend to prefer sharp corners while nature uses rounded corners. See Figures 31 & 32. There are a number of reasons why sharp corners are inefficient and impractical. We still prefer them for ease of construction, however. Cracks in a structure originate where the stresses are the greatest and this happens to take place in the corner of structures. See Figure 33. The problem is intensified when two materials are brought together by means of a fastener. The fastener is thus entrusted with handling both attachment of the materials and the resulting forces that are acting upon them. The relevance of this structural reality has been well recognized in other realms of construction and has been dealt with in an effort to prevent structural failure. Airplanes and ships must both deal with an enormous amount of stress throughout their fuselages and hulls without breaking apart. On the large scale the shape of their form is predominantly curvilinear so as to distribute forces evenly. The windows and portholes in each are also rounded to prevent crack propagation. This method of stress distribution and dissipation has been in use for millennia in many of nature's organisms, from the bones in our bodies to the forking of a branch in every tree.

2.2.7 Stiffness and Flexibility:

Stiffness – Predominates in architectural construction while nature prefers strong, flexible structures. Stiff materials like bricks and blocks are quite plentiful, easy to assemble and work quite well in compression but are quite susceptible to failure due to accidents or unusual loading. See Figure 34. Most sufficiently stiff structures are strong enough to resist collapse, however an adequately strong structure is not necessarily sufficiently stiff enough for occupancy comfort. In the search for our desired stiffness there is a proportionate increase in material that must accompany it. The stiffness encountered in natural products like bone, ceramics, coral and mollusks are made from compounds that exist abundantly in nature yet these compounds are used only in crucial locations rather than throughout the organism where other flexible materials may be substituted and possibly required.

Flexibility – With exception of the strategic use of stiff materials, the majority of an organism is constructed with relatively flexible materials. From an architectural standpoint, flexible materials are beneficial in that they can withstand extreme external conditions like the impacts of waves, wind and earthquakes without failing because they are able to flex and absorb their energy. See Figure 35. Flexibility allows a structure to alter its shape in response to the same uneven loading that can prove disastrous for stiff structures.

2.2.8 Increases in Scale:

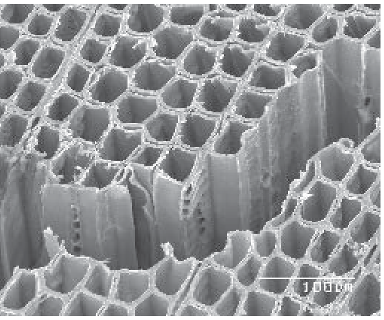
Size – When objects grow in size their volume increases more drastically than does their surface area. This can have a profound effect on the ability of the object to resist and respond to the internal and external forces acting on it for which it was originally designed. Simply scaling the size of an object does not necessarily mean that a corresponding increase in the magnitude of its structural components will prove adequate for structural integrity

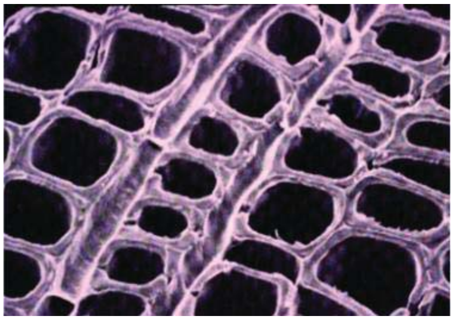
Heat – Heat is generated throughout an animal's insides but lost across its surface. One large and one small animal produce heat at the same rate. The larger volume rich, surface poor animal would be warmer. Keeping a large building heated is cheaper, relative to its volume than is a small house.

Columns – A structure may fail to support its load if a member in compression buckles, that is, moves laterally and shortens under a load it can no longer support. The critical force varies with the fourth of the column's diameter divided by the square of the column's height. Therefore, a column with a twofold increase in size (diameter and height) will experience a fourfold increase in resistance to buckling. However, being consistent with the properties of linear versus volumetric increases we end up increasing the weight of both the column and whatever it loads eight times. This results in a scenario where the dead load becomes twice what the column can support thus resulting in failure. As the scale of a building increases, it is possible to see that there is a four-fold relationship between the mass of the building and the structure required to support it. A small increase in the size of a building will result in a relatively large increase in the required building materials.

2.3 Resilience and Healing:

If an organism is subjected to an external force that causes damage a number of conditions must be met. First of all it must be resilient to the force or impact so as to reduce the initial damage experienced. This means utilizing a structural system that contains within it a redundancy of structure that distributes the force of impact and prevents a catastrophic structural failure. Subsequent to the damage the organism must be able to repair itself without a corresponding loss of function.





36. Cross-section of Douglas Fir Cells.
37. Cross section of vascular bundle in wood (xylem cells visible).

2.4 Materials as Systems:

Organisms and natural systems are often times composed of a number of interrelated components and materials that act on a continuous scale from the micro to macro structure. At each level of structural organization the cells within the organism perform a function that corresponds to a necessary requirement at that level.

The cells within a tree perform this hierarchy of functions at different scales. At the micro level the cells are responsible for the movement of water from the roots to the leaves. Based on weight, the tubular structures of the cells are also stronger than a solid structure that would not be able to act as a transport mechanism. When these cells are grouped together they provide the tree with a high strength lightweight structural system that resists both tensile and compressive forces as well as allowing for flexibility. See Figures 36 & 37.

2.5 Sensing and Responding:

2.5.1 Static and Dynamic Structures

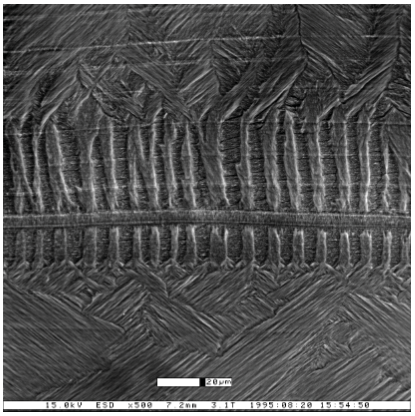
To exist and maintain itself throughout its life, an organism must possess the ability to both sense the external environmental forces acting on it and respond to these forces in a way that minimizes damage and eliminates the need for an investment of unnecessary material and structural reinforcement. The ability of biological organisms and structures to function in this regard can be categorized into two systems that are of interest.

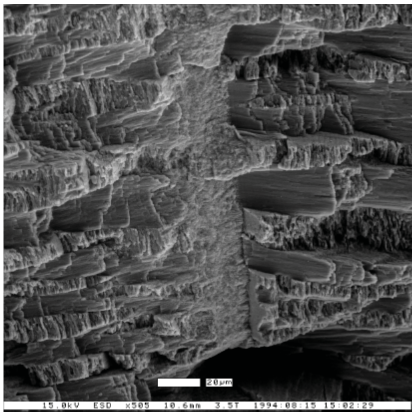
1. A closed loop system - The structure has an integrated dynamic ability to sense one or more variables (strain, temperature, etc.), process the variable, and act, sense, and reprocess to continue the performance required of the design.

Living bone is a material that is in a constant state of reformation to accommodate the changes in its loading. While these changes may occur over the course of many months, the cycle can begin within minutes of an external action.

Unlike the relatively slow and continuous process that bone undergoes, the leaves of a tree are able to realign and reconfigure themselves with quick deformation in response to wind.

2. An open loop system - This principle of design is aimed at enhancing toughness, which leads to a mechanical integrity of the system. There is no feedback mechanism but the static structural design is unique. Through evolutionary development organisms develop structural enhancements that prevent environmental damage to themselves rather than having the ability to repair themselves once damage has occurred.
Mollusks are strong and tough composites that have the ability to prevent structural failure due to their unique microstructure. Ceramic layers imbedded in a proteinaceous matrix are oriented at different angles to redirect crack propagation. (Srinivasan 1996, p19). See Figures 38 & 39.





38. Cross-section of shell matrix.
39. Detail of shell mollusk microstructure.

2.5.2 Natural Development of Form:

Natural forms are derived from their varying rates of growth and these three dimensional shapes are dependent on an irregular rate of growth throughout the organism. The form reached at the end of the growth cycle is determined both by the physical limitations of the construction material and its differential rate of growth with the latter responsible for the shape or curvature of its surface. From this it is possible to derive a relationship between the form of the object and the space it occupies. The external environment exerts a pressure on the developing object and its resulting form is a product of its response to the environment and the limits of the structural properties of the material used. It is a culmination of interacting internal and external forces. An organism in nature grows along the lines of greatest stress and it is this act of balancing the forces of stress and strain that give an object its inherent structural characteristics.

3.0 Biomimetic Principles of Form in Architecture

Architecture has long been inspired by and infused with natural forms, where a building may reference a particular organic form yet may exhibit none of the physical advantages that it could lend to an innovation or extension of architectural technology. Alternatively, a building may not allude to an individual organic form yet its function with regard to structure, mechanical or circulatory systems may be a direct result of investigations into natural principles of design and construction. This thesis concentrates on the latter, where the architecture develops from or utilizes the biological science that it derives inspiration from. The examples of built form outlined in the following section are presented here not because they are said to represent instances of organic or zoomorphic architecture, but because they are suitable examples of curvilinear forms whose definition is rooted in the natural geometric or organizational rules that define them.

40. Sagrada Familia.

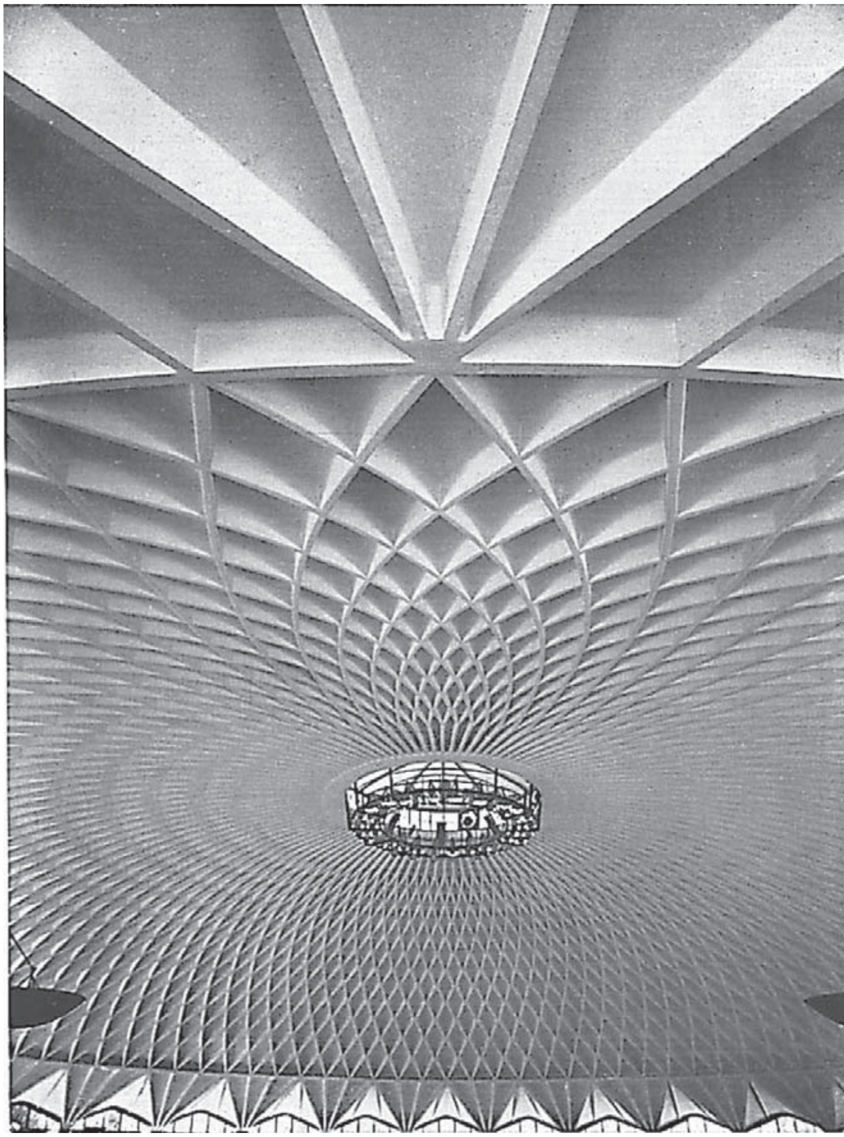
3.1 Built Examples:

Antoni Gaudi – Sagrada Familia – "Everything comes from the great book of nature." (Craven 2006) This 19th century architect closely observed natural forms and was a bold innovator of advanced structural systems. He designed 'equilibrated' structures (that stand like a tree, needing no internal bracing or external buttressing) with catenary, hyperbolic, and parabolic arches and vaults, and inclined columns and helicoidal (spiral cone) piers, first cleverly predicting complex structural forces via string models hung with weights (his results now confirmed by computer analysis). (Pearson 2001, p11) See Figure 40.

"The most important requirement for an object that is to be considered beautiful is that it fulfill the purpose for which it is destined, not as if it were a matter of gathering together problems solved individually and assembling them to produce a heterogeneous result, but rather with a tendency toward a unified solution where the material conditions, function, and character of the object are taken care of and synthesized, and once the good solutions are known it is a matter of taking that one which is most fitting to the object as deduced from the need to attend to its function, character, and physical conditions." (Martinelli 1967, p125)

Gaudi was an architect who believed that if one looks for functionality in a design then he will ultimately arrive at beauty. He thought that if it is beauty that is sought then it is only art theory, aesthetics, or philosophy that will be reached. Gaudi was able to recognize the endless variety of structural forms in nature and deduced that there is great wisdom in studying natural structures that are subjected to gravity, look for final solutions, and have evolved maximum function over millions of years. He sought to gain a knowledge of these structures and bring them into the architectural realm. Gaudi's design principles coalesced into a new theory that united three previously disparate areas of architecture where: "...the mechanical fact is geometrically demonstrated and is translated into three-dimensional material, making it structural. Mechanics, geometry and structure have been synthesized to produce a logical architecture in which each active element fulfills its function in an equilibrated way and with the least effort." (Martinelli p134)

"The helicoid is the form of a tree trunk, and Gaudi used this form in the columns of the Teresian School. The hyperboloid is the form of the femur, a form he used in the columns of the Sagrada Familia. The conoid is a form frequently found in the leaves of trees, and this form he used in the roofs of the Provisional Schools of the Sagrada Familia. The hyperbolic paraboloid is formed by the tendons between the fingers of the hand, and he built with this form the porch domes of the church crypt in the Guell Estate." (Nonell 2000)



41. Palazzetto dello Sport.

Pier Luigi Nervi – Palazetto dello Sport, Hangar – Italian architect/engineer responsible for a series of constructions based on the form of the equiangular spiral that appears with regu-

42. Tsui's Ecological House of the Future.

larity in the natural world. Nervi looked to nature as a teacher that seeks to achieve optimal results with minimal effort, while also creating harmony where beautiful proportions and relationships manifest themselves through mathematic principles. He experimented with these principles to establish a harmonious relationship between the internal reinforcement and the external skin that enveloped it (Portoghesi 2006). The ability to develop these delicate forms came when Nervi made a breakthrough in the field of reinforced concrete: the invention of ferro-cemento. This material was formed using steel mesh as a core with layers of cement mortar brushed on top of it. The steel mesh was thin, flexible, and elastic, and its addition to cement created material which could withstand great strains. Ferro-cemento enabled Nervi to design any form he wanted, giving him a way to address the problems of stress and static equilibrium with greater freedom from convention than was ever before possible. In order to reduce the cost of construction the material could be easily prefabricated in plaster molds. This approach allowed the building - skin and structure - to become one cohesive unit. (Leslie 2003, p45). See Figure 41.

Eugene Tsui – Tsui has designed and built a number of projects that have developed through his fascination with nature and the process of evolutionary biology that he is heavily involved. His works take their inspiration from a variety of organisms whose different structural and functional characteristics inform the individual projects to which they are associated. While his projects are expressly zoomorphic in character they are always infused with natural design principles that underlie the forms. Tsui has performed extensive structural testing on a number of natural forms and uses his results to develop his architecture.

"Dr. Tsui is not imitating nature's shapes. He is attempting to enter into the very "mind" of nature—the source which creates the forms and processes—and apply this knowledge to create a new architecture, a new attitude of our living environments. No other architect in history has looked deeply into nature, in a rigorous and scientific way, and then apply these discoveries to architecture." (Tsui 2006). See Figure 42.

3.2 Unbuilt Examples:

Ken Yeang – Bioclimatic Architecture – Yeang's designs follow the theme of 'urban ecosystem', a holistic design solution that deals actively with milieu for pedestrian flows, plant growth and the equilibrium of energy, waste and water. Yeang believes that all architecture ought to respond ecologically to the natural environment as a whole. His designs aspire to making a direct contribution to a sustainable ecological future. (Yeang 2002) See Figure 43.

Peter Testa – Carbon Tower – Helical structural system that puts a heavy reliance on tensile forces and the use of redundancy in material to prevent complete failure of the system if a localized failure occurs. All of the building components are constructed of the same material that is woven together and eliminates the structural inefficiency of joints. (Knecht 2006) See Figure 44.

EMERGENT Architecture – Radiant Hydronic House - A prototype house that was developed through a feedback of various building systems into one another in an effort to produce emergent effects, both quantitative and qualitative. The structure of the house is composed of a set of flexible bands which function at different levels of behavior from structural to mechanical to circulatory based on both the local environmental requirements as well as on the behavior of the adjacent members.

43. Yeang's bioclimatic skyscraper.

44. Testa's carbon tower.

A central spine satisfies the environmental requirements by unifying them into a monocoque structure. The ductwork also functions as structural support and circulation platform. The building systems of the house were conceived of not as singular entities that were individually optimized rather the design sought to optimize the function of the whole. (Emergent 2005a) See Figure 45.

EMERGENT Architecture – Lattice House - A design proposal for Vitra based on a monocoque structure that strives to integrate every level of building system from structural to electrical into one three-dimensional latticework that is generated by its spatial morphology. The Lattice House is a flexible array of space that contains in its genesis a diverse amount of morphological possibilities for its final form.

The project uses Inverse Kinematics 'bones' in order to generate a multidirectional array that maintains a dynamic coherence in the system. The framework functions simultaneously as primary structure and mechanical infrastructure. A whole structure heat-exchange system, essentially a 3D radiator, capable of heating and cooling the space is created without the use of forced air by filling the structural struts with water. Struts also evolve locally into stairs, bridges, and secondary propping elements.

The final design was derived through 'breeding' the structurally fit iterations of the design that were subjected to structural loading analysis. (Emergent 2005b) See Figure 46.

3.3 Use of Structural Form in Architecture:

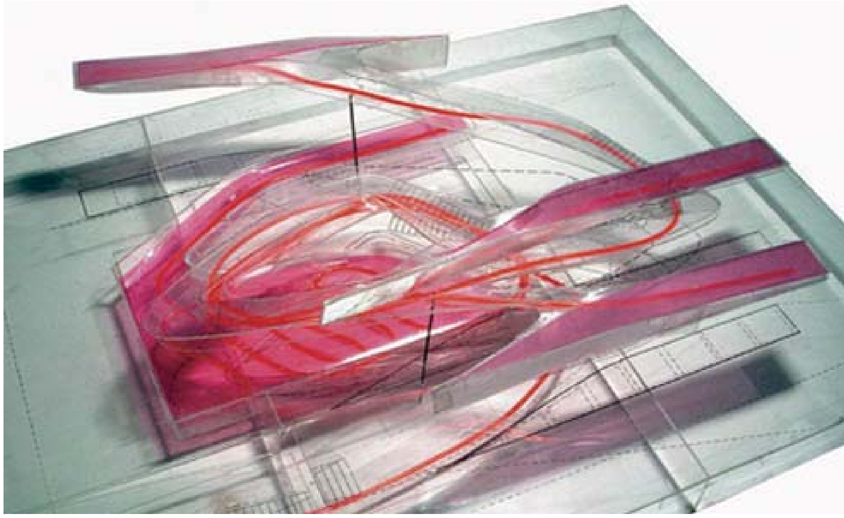
The architects and projects listed here are representative of a larger collection that have sought or are seeking to derive innovative structural solutions through an efficient use and understanding of geometry and its relevance in construction. The research and development techniques utilized span the spectrum from physical modeling to intensive digital development and analysis. While all of these designers may not pursue an explicitly biomimetic approach in their designs it is evident that many of their designs contain underlying geometry or principles that are found in nature. The implication here is that with a better understanding of nature's design and construction principles it becomes easier to produce complex forms that contain an elegant simplicity.

Designers with projects that invoke design languages that rely on complex geometries.

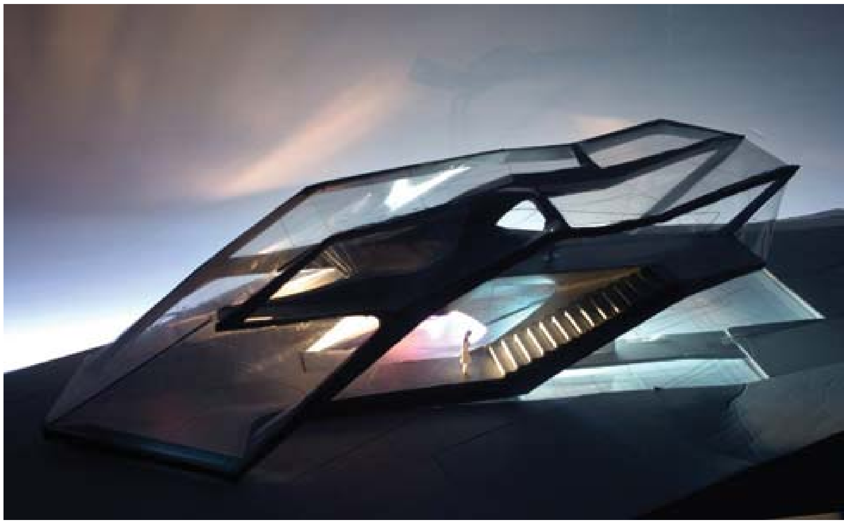
Antoni Gaudi
Victor Horta
Frei Otto
Felix Candela

Current designers utilizing complexly curved and nonlinear members and surfaces

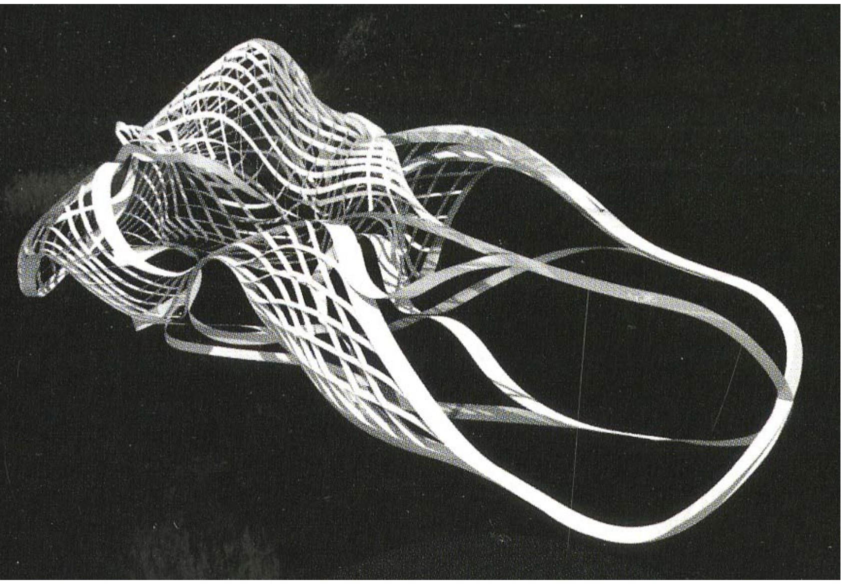
Morphosis
Santiago Calatrava
Norman Foster
Coop Himme(l)blau



45. EMERGENT Architecture's radiant hydronic house.



46. EMERGENT Architecture's lattice house.



47. NOX: A-life, an earlier version of Son-O-house.

48. NOX: Structural ribs defining a doubly-curved surface are clad in narrow woods strips the follow the curvature much like in shipbuilding.

NOX – Machining Architecture

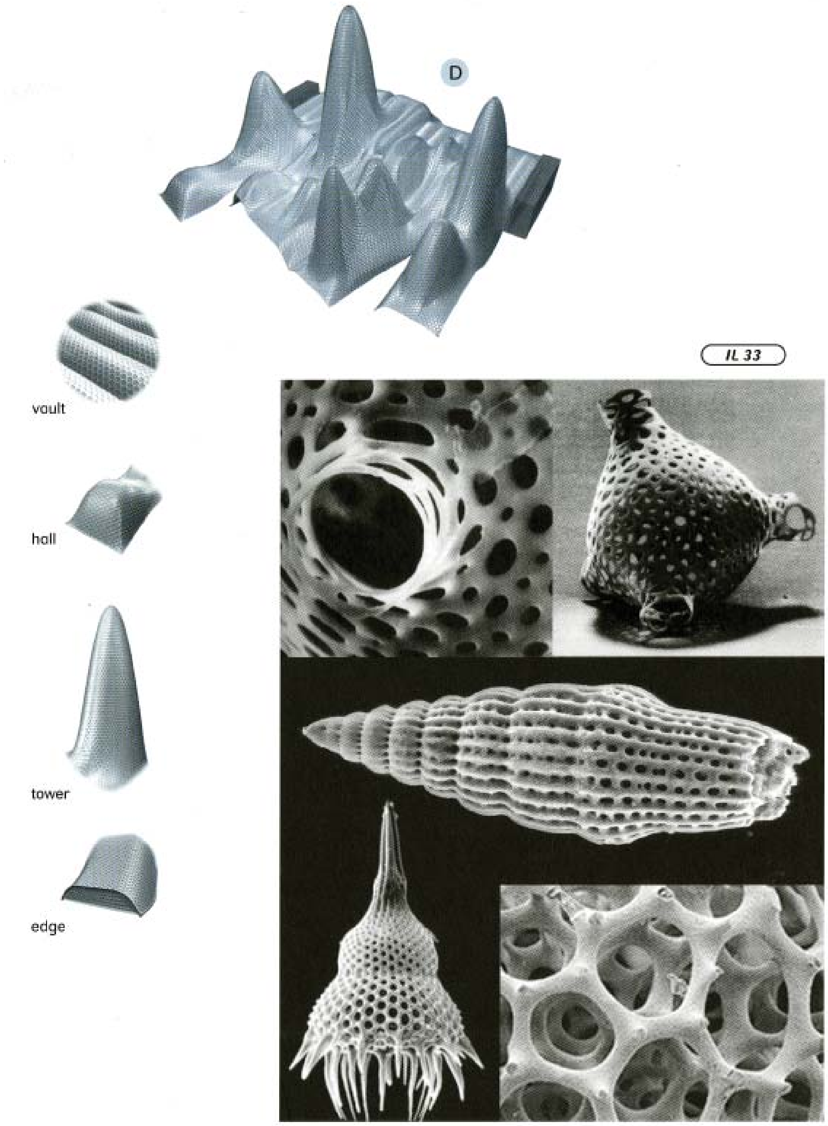
Pompidou Two - In an effort to reduce structural hierarchy and complexity of the exterior surface the project was conceived of as using geometries that transition from single curvature to double curvature. Long, linear elements acting as primary members where derived with straight rules or simple arcs. A bifurcating lattice branched from the primary elements to produce a doubly-curved lattice that much like the shell of an arthropod does not rely on a hierarchy of primary and secondary structure. See Figure 47.

Surface to line – Effectively covering a doubly-curved surface continues to be a challenge for designers. In Parc Guell, Gaudi had the idea of using waste pieces from regular square tiles that had broken on the factory floor. The polygonal elements created a pattern of cracks on the benches that occurs in craquelure and Voronoi diagrams. Spuybroek's thoughts on surfacing then shifted from thinking in joints to thinking in cracks. His idea was to segment the surface during geometrical formation instead of beforehand. The desire is to develop the geometric form, structural form and panelization in a concurrent manner rather than sequentially. This type of process leads to the feedback scenarios associated with natural constructions.

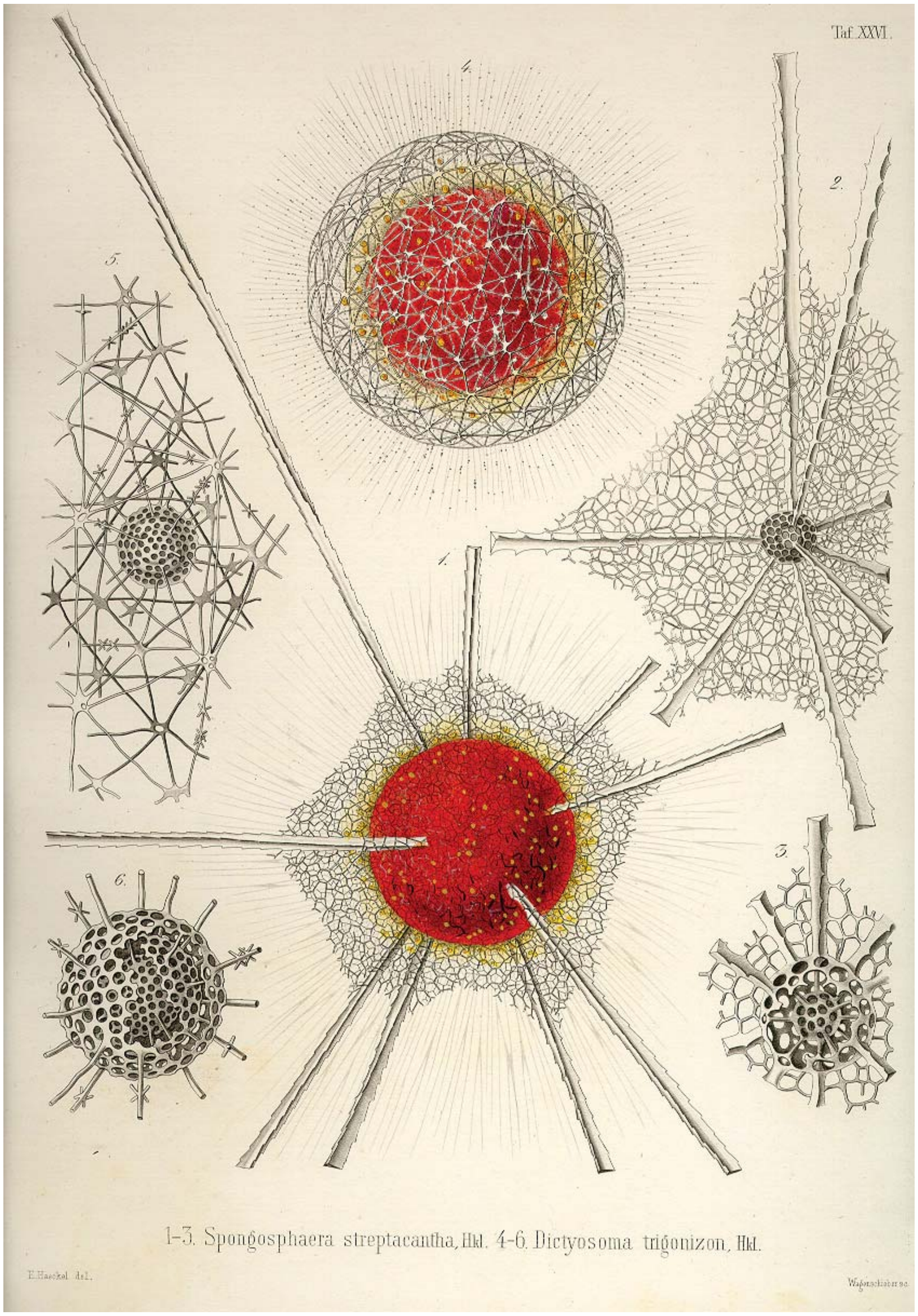
Line to surface – Typical surfacing procedures consist of breaking the developed surface into lines. Spuybroek outlines a fascination with a Gothic type of logic where lines bifurcate and weave themselves into surfaces. The simple curves begin to develop patterns of interlacing that evolve into larger and more complex configurations that satisfy not only aesthetic but structural requirements. The Gothic builders were able to develop and use arabesque patterns that transcended a strict ornamentality. (Spuybroek 2004e)

Son-O-House - Once again the issue of panelization of doubly-curved surfaces arises where Spuybroek regards tessellation as the subdivision into or addition of tile modules to a surface. The least interesting yet often most cost effective method of tessellation is triangulation, where the surface is partitioned into triangular facets each of which is planar. A variable approach based on textiles was used here where flexible bands are able to create a substrate for the hardened tile. (Spuybroek 2004g)

ECB - In this design for the European Central Bank, Spuybroek looked to Radiolaria (micro-organisms around 0.1 mm in size) for inspiration. See Figure 49. "The amazingly beautiful drawings of Ernst Haeckel from the early 1900s and the research of Helmcke and Otto throughout the second half of the twentieth century show that Radiolaria are of a highly architectural nature. See Figure 50. For these German bioconstructivists this is another argument in favor of the idea that a substantial part of the living form is non-genetic in origin. What makes the study of Radiolaria so relevant is that it teaches us that variation is a product of uniformity or, better, isomorphism; and second, that isomorphism is not fatally attracted to the Sphere but is the generator of ribs, spikes, creases, tubes, and the like. Variation within the system can produce variation of the system." (Spuybroek 2004b)



49. NOX: Design for the European Central Bank based on Radiolaria morphology.



50. Ernst Haeckel's drawing of Radiolaria from the Family Spongurida.

4.0 Investigation Into Surfaces and Manufacturing

While it is possible to derive efficient structural forms from a biomimetic investigation into natural designs, their logical development and efficient translation in built form must occur with knowledge of the geometric principles inherent in them. A mathematical analysis of surface and curve definition serves to allow for a reliable and informed translation from physical observation into digital generation. The methods for physical construction of a design are outlined in an attempt to align the biomimetic investigations with the realities of current construction technologies. While some natural design and construction methods may be highly efficient and ideal for architecture, their realization as manmade constructions may not be possible until current technologies evolve further or new ones are developed.

4.1 Curved Surfaces – Definition, Generation and Analysis

Perhaps the most obvious way in which designers have benefited from the advancement of digital design software is in the realm of curved and complex surfaces. However, there are trade-offs that frequently arise with various programs and their effective utilization at certain points in the design and construction process. The starting point for many architects is to create a surface model that closely approximates the shape and form that is desired. This process can occur rapidly and changes are also readily accomplished. Once the surface model has been obtained it is then necessary to create a solid model that is derived from those surfaces. A solid model is essentially a volumetric representation where complex surfaces that define the morphology of the model are numerically exact for proper manufacturing and construction. Often times a program that excels at surface modeling is hindered when performing solid modeling and vice versa. The development of solid models from surfaces can be accomplished through a number of techniques which can have resounding effects when it comes to manufacturing and construction. (Schodek, 2005, p6)

4.1.1 Surface Curvature

A curve can be mathematically described whereby at any point the shape of the curve will have an *instantaneous radius* (R) and an associated *curvature* (1/R). The instantaneous radii can be thought of as defining a circle that most closely traces and passes through the curve at that point and has a center point tangent to that point. The curvature is essentially the reciprocal of this instantaneous value. The smaller the radius of the curve is, the larger the associated curvature will be and vice versa.

The parabola is composed of a constantly changing curvature gradient whose instantaneous radius at its apex will be quite smaller than that at its end. This characteristic of a varying curvature from point to point can be seen in most other curves between the straight line and circle. Like the values for the instantaneous radius which exist at an individual point, so too does the *instantaneous curvature* rely on individual points. By selecting a point (A) on a surface it is possible to derive a line that is normal to the surface at the point (A). It is now possible to obtain a surface plane which passes through point (A) and its normal line. This *normal plane* if extended to intersect the surface will create an intersection curve called the *normal section*. Additionally, the instantaneous curvature at point (A) is referred to as the *normal section curvature.*

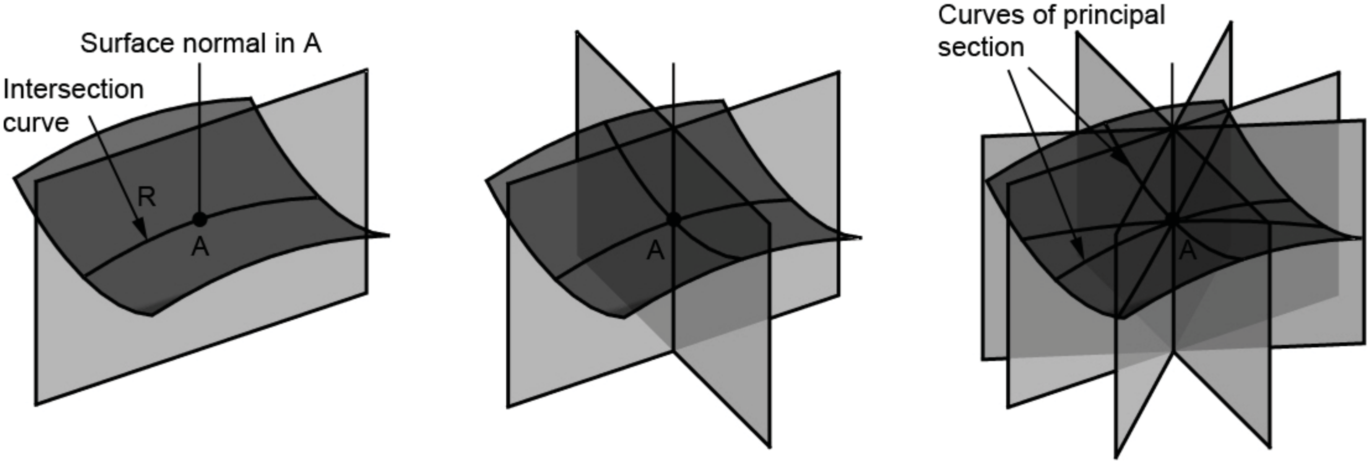
From Figure 51 it can be seen that the normal plane can be rotated in any increment around the normal line which would lead to an infinite number of normal sections each with its own unique normal section curvature. From this it can be stated that throughout the number of normal sections there will be one maximum value (k_{max}) and one minimum value (k_{min}). These two *principal curvature* values can be found by rotating the normal section plane until these values are found. (Schodek 2005, p195)

4.1.2 Gaussian and Mean Curvature

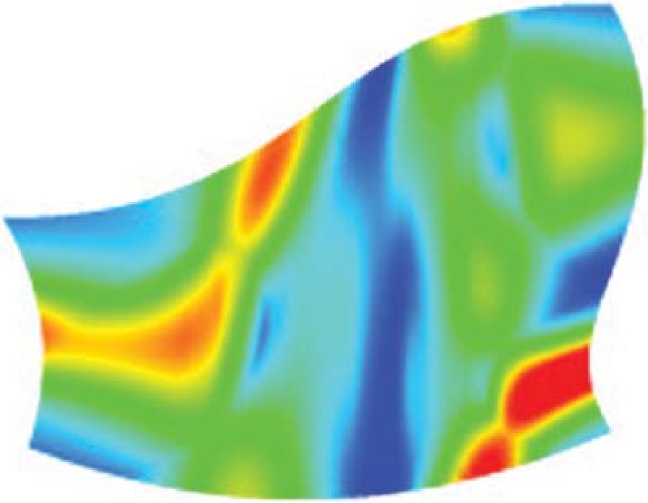
Gaussian curvature can be thought of as being the product of the two principal *normal section curvatures* at a point where $k_g = k_{max} \times k_{min}$. The *mean curvature* k_m is the average of k_{max} and k_{min}. A surface with a positive Gaussian curvature can be referred to as *synclastic* where the normal section curves have the same sign in all directions. These surfaces belong to all concave and convex shapes and are nondevelopable whereby the surface cannot be flattened without material distortion. A negative Gaussian curvature in a surface is called *anticlastic* where the principal curvatures are of opposite signs. These surfaces are not developable either even though some are classified as ruled surfaces. If the Gaussian curvature is equal to zero everywhere on the surface then it can be fully developed into a flat plane without any material distortion. In this case one of the principal curvatures must equal zero which in effect creates a straight line. (Schodek 2005, p196)

4.1.3 Curvature Investigation and Representation

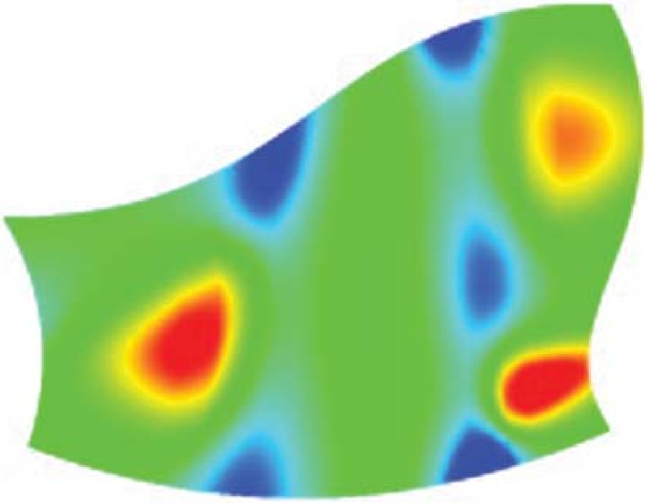
Many advanced modeling programs today have provision for analyzing surface curvature. These curvature values can be displayed numerically or visually depending on preference. Colors or hues can be set to correspond to varying degrees of curvature as well as positive and negative values. With this technique the designer can quickly visualize the surface to determine whether it meets the desired shape and is free from unwanted deformities. A complex surface form composed of a number of different surface curvatures can be also be quantified with regard to the degree and type of curvature with respect to cost implications. On a monetary scale the expense of cladding panels will increase from planar to



51. Curvature of surfaces: normal curvature and related principal values of a synclastic surface.



Mean Surface Curvature Display



Gaussian Surface Curvature Display

52. Curvature analysis diagram.

doubly curved. By visually defining the surface condition for the panels it is possible to get a graphical representation as to the proportion or areas of the façade that may be too expensive and therefore require adjustment. (Schodek 2005, p196) See Figure 52.

4.1.4 Conical sections and surfaces derived from them

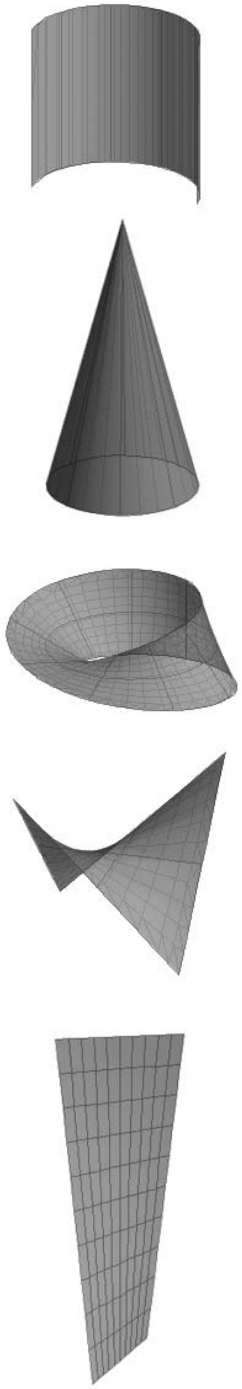
Many complex surfaces if created with some comprehension of basic curves can be created by combining a number of these curves. Conical sections for example are readily used to create curved surfaces that can be easily calculated mathematically. Through a number of different operations such as revolving, lofting, sweeping or any combination of the same it is possible to create domes, parabolic surfaces, barrel vaults, and hyperbolic paraboloids. Of note here is the fact that these surfaces can be understood relatively intuitively and have the benefit of being more easily created and manufactured with less digital computation than more complex surfaces.

4.1.5 Ruled and Developable Surfaces

A ruled surface is any surface that can be derived from a translational sweeping, with optional rotation, of straight lines. See Figure 53. The surfaces derived from these manipulations can take the form of cylinders, cones, and conoids in one group, and hyperbolic paraboloids and hyperboloids in another. However, while all of these shapes are deemed as ruled surfaces, there are two significant differences that separate these two groups where the first group consists of developable surfaces and the second group nondevelopable. Developable surfaces have the ability to be unrolled or flattened into a sheet without deformation. Nondevelopable surfaces must be cut or deformed in order to be constructed from a flat sheet of material.

4.1.6 Complex Surfaces

The designs seen today in architecture quite often take the form of surfaces whose defining layout curves are becoming increasingly more complex and not as easily defined as those of the ruled and developable surfaces. While the creation of models with curves such as B-splines and NURBS can be carried out with similar modeling techniques as to those mentioned above, their mathematical derivation and visual comprehension can far exceed many simpler surfaces. Added manufacturing complexity also arises in these cases due to the inherent inability of a planar surface to be formed into a complex surface without either extensive material working and deformation or a much more elaborate method of faceting to arrive at the desired configuration.



53. Ruled surfaces

4.2 Primary Structural and Construction Specific Considerations

4.2.1 Construction Considerations

Historically speaking, when geometrically complex building forms were built, as with the works of Victor Horta for example, they respected the limitations of the current construction technology. The designer recognized their responsibility for expressing their design intent through precise and comprehensible representations that could be understood by all of the parties involved in the project. Even designers seeking to create apparently non-definable forms began to develop new ways in which to manufacture the complex geometrical forms in line with the appropriate construction techniques.

Between the period of 1914 to 1926 when Antoni Gaudi worked on the Sagrada Familia, he developed a set of construction rules that the masons were able to follow. His generation of the principal architectural elements was based on "ruled surfaces" which included the hyperbolic paraboloid and the hyperboloid of revolution, both of which are doubly curved and non-developable.

While different in their architectural expression, the later works of Felix Candela and Pier Luigi Nervi used the same conceptual approach as Gaudi. These men made extensive use of those kind of surfaces in the reinforced concrete structures that they designed. In this manner the wooden formwork could be easily erected out of flat wood planks. (Schodek 2005, p49)

4.2.2 Structural Considerations

Structural efficiency is an aspect of design that may or may not be explicitly considered when generating complex building forms. While many civil engineering structures that utilize complex geometries (dams) are responsive to both structural and technical efficiency, this is often not the case with regard to architectural constructions. The simple act of forming a curved surface does not automatically infuse it with the positive structural benefits that are possible with certain curved surfaces. The classic doubly curved shapes such as portions of spheres or the hyperbolic paraboloid shapes used by architects in the late 19th and early 20th century have been widely proven to demonstrate "*membrane action*" where internal forces are efficiently transmitted through the surface of the shell in an in-plane manner. See Figure 54. When this scenario exists, the stresses acting out of plane within the surface are quite low and thus the shell can be made quite thin. Membrane action does not exist in all curved surfaces and its presence in a surface depends on the existence of particular combinations of surface shapes and types of loading conditions. It is important to note that with a corresponding decrease in the amount of material associated with the proper development of a structural skin that exhibits membrane action the skin will also be more susceptible to deformation due to local or point loads. A proper balance between these must be met or the design of the membrane must act on a variety of levels to redistribute stresses imposed on it.

The misconception that curvature automatically translates into structural efficiency is quite prevalent in construction today. Complexly curved surfaces and their widespread use can often be immature versions of properly designed surfaces that could potentially exhibit the desired characteristics of membrane action. It is only through careful examination of the design, functional criteria and intent along with structural analysis can the final product exhibit the structural advantages associated with a curved surface. (Schodek 2005, p48)

4.3 Defining Surface Shapes

4.3.1 Digital Form-Generation Techniques and Shape Generation

Many of today's computationally based design approaches to complex geometric forms focus on arbitrary form generation, with minimal attention paid to manufacturing, construction and structural efficiency.

Common vs. Uncommon Approaches

Common – The designs are envisioned by the user and the digital tools act to develop and represent these ideas. The inspiration for complex and unique shapes is derived from many different sources, ranging from direct responses to programmatic requirements.

54. Roof of Nervi's Palazzetto dello Sport which exhibits membrane action

Uncommon – The designers develop computational environments whereby the design is developed by the program through pre-specified rule structures or other principles.

The most widely used approach for shape generation used by designers is the direct use and manipulation of computational tools (points, lines, splines, lofts, sweeps, etc.) commonly found in a variety of digital modeling environments (form-Z, Rhinoceros, MicroStation, etc.).

Computational tools that are visually oriented and based on descriptive geometry or on other mathematical means of describing lines, curves, and surfaces can also be used in a more direct manipulation process to generate forms. Software technologies associated with this type of shape derivation are uncommon in the architectural design environment but are found in broad based mathematical tools (MathCAD, Mathematica, Maple).

In an effort to derive forms based on a set of external influences be they real or metaphorical, some designers have adopted the use of software (Maya) that allows for an influence of form based on force functions of on type or another. Objects or functions within an environment can be given a defined set of controllable parameters that afford them the ability to influence and interact with other objects that can in turn push, pull, deform and essentially drive shape generation for the resultant form.

Parametrically driven shape derivation is also being used in a more controlled manner, whereby the forms are generated according to sets of predefined rule structures and component parts. The design approach within these software applications can vary from one to another where priority can be placed on having a strong construction rationale or through different programmatic or conceptual intents (Generative Components, CATIA, SolidWorks, Unigraphics, CADDS5). A commonly used approach here is to define a set of parameters for a structural element whose form drives the formation of the building envelope. The parameters defined can be related to the physical dimensioning of a component or any number of relevant values or relationships. Through direct manipulation of these control parameters the changes will propagate throughout the model to instantaneously update it.

A recent trend is based on an approach that seeks to derive form through the implementation of genetic growth or repetition algorithms. Patterns seen in nature such as fractals and tessellations can be broken down into complex rule structures that can be in turn modified and used for shape generation.

The idea of time and temporality in architecture is often overlooked and it is in this regard that some architects (Kas Oosterhuis and Ole Bauman) have sought to develop buildings that effectively change throughout time and to various external forces. Here, architects are not designing static structures that maintain their structural form but ones that are capable of adapting to new uses or needs. Just as cultural changes occur over time, these buildings would modify their layout and organization to best serve the immediate needs of the user with the possibility to serve future uses equally well. Digital environments that support animation and motion (Maya) are useful here.

4.3.2 Physical Model to Digital Model

While the digital environment can be invaluable when deriving, representing and promoting designs to construction, a great number of architects still rely on physical modeling techniques as a rapid and tactile way in which to arrive upon a desired formal scenario. The models of churches, cathedrals and other buildings that remain from centuries ago are incredible reminders of how valuable physical modeling can be both in design and preliminary structural analysis. Digital scanning techniques and computationally based programming software now allow architects to scan a physical model for promotion into a digital model which in turn allows for the production of a physical model for further physical manipulation. Once the physical model has reached its desired configuration then the project can progress for subsequent development in the translated digital form. The process of digital scanning is still relatively raw in practice because the scanner will create a set of surfaces derived from the physical model that the program must then be manually guided to stitch together. This surface model must then be translated into a solid model through the appropriate program. (Schodek 2005, p52)

4.3.3 Form Finding Through Structural Viability

The digital techniques of form generation illustrated up to this point are all methods in which to conceptualize and generate complex surfaces. The forms derived from these however, do not necessarily translate into viable structural systems with efficient methods for production and construction.

Previous to digital computation software it was through accurate physical models (hanging chains, minimal surface experiments with soap or stretch fabric) that structural form finding was carried out. These approaches are still effective today with the possibility for their promotion into the digital environment through 3D scanning techniques. The computational approaches outlined above should not be confused with the computational systems described here which include the *force-density method* and the *dynamic relaxation technique*. Both of these are designed to minimize the embodied potential energy and balance the forces in the system through the optimization of the building form itself. The optimal shape is one that maintains equilibrium between the external loads applied to it and the internal forces that resist these loads with a subsequent minimization of material. Whether it be through physical or digital form finding techniques, the manipulation of form is only possible through changes in loading of the structure or to the support and boundary conditions with each resulting in a unique shape.

4.3.4 Structure and Enclosure

When designing a surface enclosure that is composed of compound curves there are many considerations that need to be addressed early within its development. Included in these is the question of whether the surface will be required to be structural or not. If the surface is intended to be structural then there must be the associated investigations into whether the surface is also load bearing with regard to live and dead loads as well as natural forces such as wind and earthquake. If the surface is not intended to be structural then its relation to a primary structure must be developed. In line with structural considerations are the requirements for glazing/transparency, energy requirements, material viability, ease of construction, maintenance and other factors involved in the design of any enclosure.

Another question is whether the exterior surface relates to the interior surface whereby there is a single defining surface. If so then both the enclosure and structure must be combined into one system. If the exterior and interior spaces are unrelated then the structural system has the possibility to occupy the interstitial spaces between them which invariably allows for a greater degree of design choices.

4.3.5 Approaches to Building a Large Compound Curved Surface

Subdivide the surface – Lines of structural framing are placed to correspond with the surface division. Smaller, lightweight enclosure panels then span between the primary structural elements. In this scenario the primary structural elements would often be composed of compound curves and the associated enclosure panels would be doubly curved. In an effort to reduce the complexity of this system it is possible to compose the structure of planar facets that are connected to linear structural members. (Schodek 2005, p200)

Sectional planes at regular intervals – By dividing the structure into a set of repeating sectional planes it is possible to design structural members that although curvilinear remain planar with the surface and as such avoid compound curves. An egg crate pattern begins to develop when horizontal sections are passed through the structure as well which allows for smaller enclosure panel sizes. See Figure 55.

Slab support system – On a larger structural scale and in a project with multiple floors it is the floor plates themselves that can become the horizontal sectional planes with the exterior panels spanning between them. See Figure 55.

The creation of a smooth doubly curved surface will usually require the integration of surface and structure together as in a structural shell or where the structural elements and the surface enclosure are curved. When the structural scale with regard to the surface size is increased then the surface will have a tendency to become more faceted and conform less to the desired shape. This has the practical implication of reducing build complexity and cost. (Schodek 2005, p54)

4.4 Structural Surfaces – Translation from Digital Design to Physical Fabrication

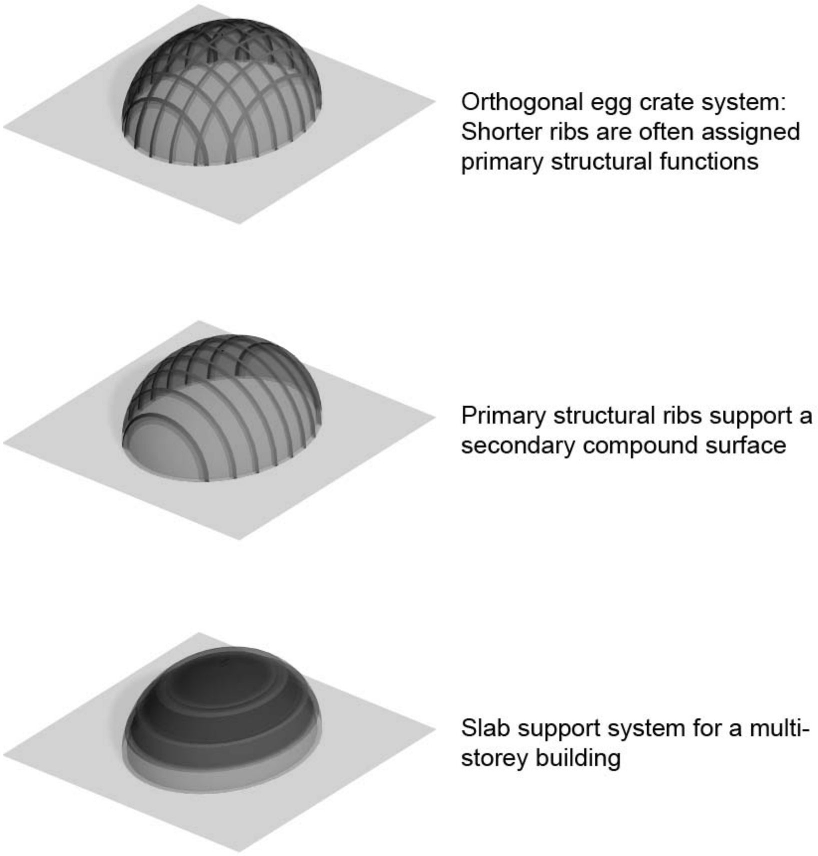
When designing a building in a relatively unrestrictive digital environment it is often useful to have an idea of the type of building material to be used and the construction techniques involved with the use of that material or system. With an idea of the possibilities and limitations inherent with use of a particular material and construction approach the designer can avoid spending time on creating forms that are unrealistic with regard to their development and manufacture.

4.4.1 Large Continuous Surfaces

There is a wide range of material possibilities for manufacturing curved surfaces, from reinforced concrete all of the way to pre-stressed structural fabrics. The techniques associated with their construction vary widely as well. In the case of reinforced concrete and classic masonry construction there is often an intricate system of formwork involved to achieve the final form. This approach has been aided with the use of CAD/CAM technology where the formwork can be CNC machined to provide the proper curvature. It is the incredible surface fluidity that is achievable with poured concrete that continues to attract architects today.

Where the structure itself is composed of intricately carved stone there has been a tradition of manual carving which is labor intensive and costly in today's market. While this approach has been updated with the use of CNC cutting, milling and routing machines as in the new work being done on the Sagrada Familia in Spain, it still remains an issue of cost for many. In an effort to reduce material costs this scenario has been reduced to affixing a thin stone veneer to a distinct structural core.

Wood has a history most notably in shipbuilding for being shaped into curvilinear forms. The relatively recent technology of glue-laminated lumber has added another dimension to the structural possibilities of wood in addition to the ability of CAD/CAM technology to both provide data for the construction of the



55. Strategies to support complexly shaped surfaces.

required jigs as well as making viable the creation of complexly curved surfaces.

The panelized unit which is usually constructed of thin sheet metallic, polymeric or composite materials has typically been difficult to develop into a system that in itself works as a structural system. It is often necessary to provide a secondary stiffening system. In the same way, surfaces consisting of woven or layered strips cannot function efficiently unless multiple cross bonded layers are used to achieve the required cross-sectional structural depth.

4.4.2 Small Continuous Surfaces

Advances in material forming have allowed the production of complex surfaces that exhibit structural capabilities and are well suited for relatively small structures. As the forces begin to multiply for larger structures, the structural possibilities associated with these materials begin to diminish and are usually inadequate to serve for these larger structures.

Fiberglass has historically been used in a wide variety of applications to create large, smooth, and stiff surfaces. Within the automotive, aerospace and naval industries, the use of fiberglass has essentially involved laying multiple resin-impregnated strips or sheets of fiberglass over a curved framework for curing. Advancements in the composites industry have produced materials (carbon fiber, kevlar) that offer incredible structural properties with a drastic reduction in the amount of material necessary and as a result a reduction in the dead weight of the structure.

With the use of CAD directed finite-element analysis of a proposed structure in its digital form, it is possible to develop strategies for built up and layered composite systems that derive their strength or additional strength from the directional placement of individual strips along the lines of force contained within the surface. By applying material along the direction of the forces involved there is a reduction in the amount of material necessary to resist those localized forces. See Figure 56.

Doubly curved metal panels have continued to remain of interest to architects that desire a curved surface that can be structurally supportive and weather resistant with the desired fluid and monolithic aesthetic. Smaller units can be molded or stamped while larger panels which are inherently nondevelopable must undergo extensive deformation or slicing with subsequent rejoining to achieve a compound surface. Numerous cold forming techniques are available to the designer including rolling, stamping and planishing. These techniques, with the exception of rolling, require a considerable investment in either time or tooling which can become cost prohibitive if there are a large number of unique pieces to be made. (Schodek 2005, p55-58)

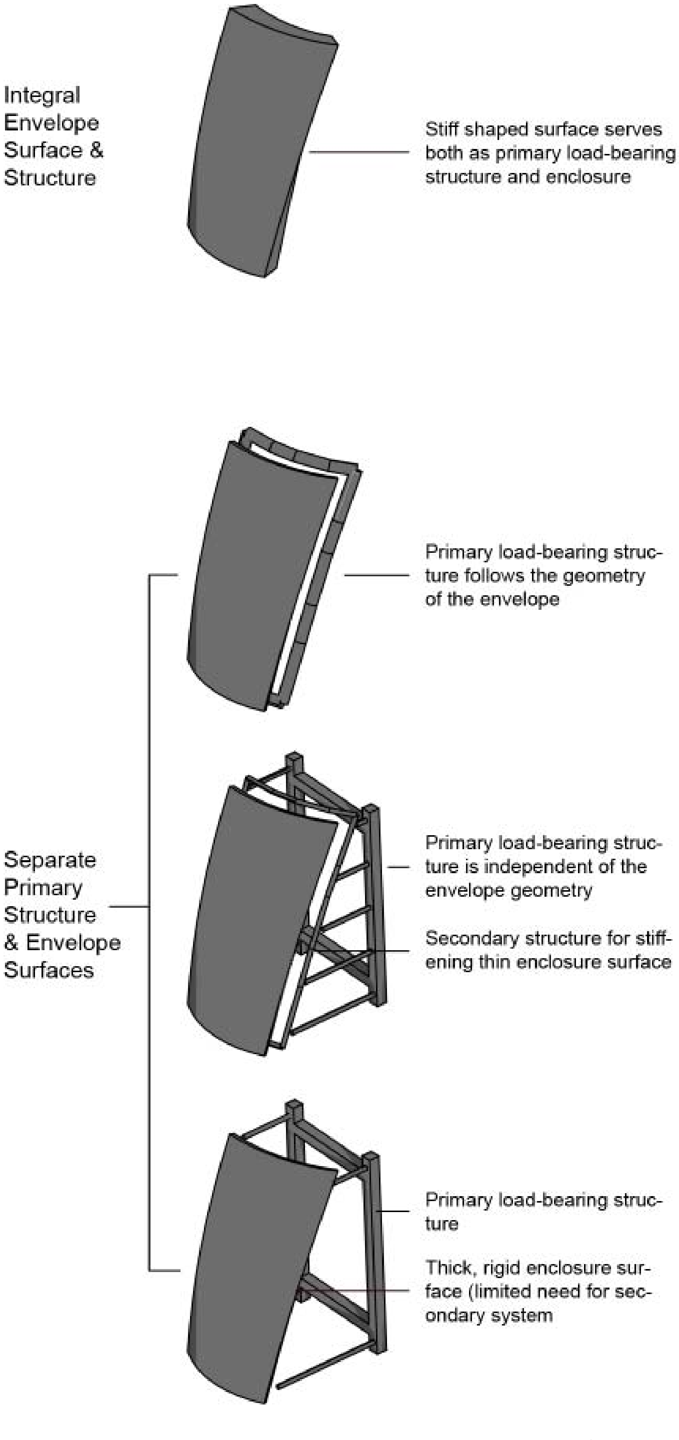
4.4.3 Surface Enclosure

If the surface itself is not capable of handling the intrinsic structural forces that must be resisted then it is necessary to introduce a primary structural system that can. The outer surface of the building then becomes predominantly non-load bearing with the only structural requirement being that of resisting local loading. This approach typically sees the primary structure designed according to a less complicated method of manufacture and construction. If the interior and exterior forms differ drastically it may be necessary to introduce a secondary structural system that is a means of connection between the primary structure and the façade. The most complicated problem with this technique is the derivation of the correct offsets and positioning of the secondary members and their corresponding attachment points to both the primary structure and the surface as well. This process is simplified with the use of advanced CAD technology, however the suitable programs are quite difficult to learn/use and may be cost prohibitive for many designers. See Figure 57.

56. Directional layers of fiberglass laminated to a formed balsa core

4.4.4 Thin Sheet Surfaces

On a small scale it is possible to manufacture complex surfaces through the use of CNC produced forms where the chosen surface material is subsequently formed or stamped directly on it. Metal panels can be produced in this way but they are often limited to thin wall sizes and small bounding dimensions. As the size and thickness of the metal sheets



57. Relationships between skin and structure for complex surfaces.

increase they become increasingly difficult to deform and produce the desired complex shapes. Due to the limited thickness possibilities for stamping the use of metal panels here is limited to a surface condition that prohibits them from performing in a load-bearing capacity without deformation. Curvature in one dimension however can be easily accomplished through rolling and as such allows for panels with greater size and thickness. This enables the designer to reduce the secondary system required for attachment to the primary. Depending on the complexity of the skin configuration a balance must be met between the formability of the individual steel panel and the complexity of the secondary system.

The evolution of a traditional method for steel fabrication is in development by the Navy Joining Center (NJC) along with a number of other partners. The technique called Automated Thermal Plate Forming (ATPF) is a process whereby numerical modeling, digital measurement and intelligent computer feedback programs will work in concert to produce repeatable, high accuracy formed steel plates. This process of thermal formation is currently performed by skilled operators using oxy-fuel torches and manual quenching with water. While both approaches allow for the formation of simple and compound curvatures the manual approach is quite labor intensive and limited by the experience of the operator. The automated system is composed of four mechanisms including path planning software (PPS), an induction heat source (laser), a manipulation and plate holding device, and an automated measurement system (AMS). The PPS will produce a required set of heating paths and parameter sets based on the desired 3D conformation (CAD derived) and the initial plate shape which incidentally is not limited to a planar configuration. The PPS will output data to the manipulation system that will direct both the movement of the heating unit as well as the plate itself. Once the forming has occurred the AMS will measure the final plate shape and compare these values to the desired shape. If necessary the PPS will automatically derive any new heating paths required to achieve the final form. This new technology has the ability to increase quality, decrease costs and reduction production times. The Navy expects that with regard to its DD(X) advanced multi-mission destroyer they will see a 100% increase in throughput, 80% reduction in rework, 50% reduction in direct labor costs, and 75% reduction in support labor costs. As can be imagined the potential applications with regard to architecture are widespread and the associated cost reductions over conventional forming methods will allow for its use on a greater number of projects. (Coffey 2006) See Figure 58.

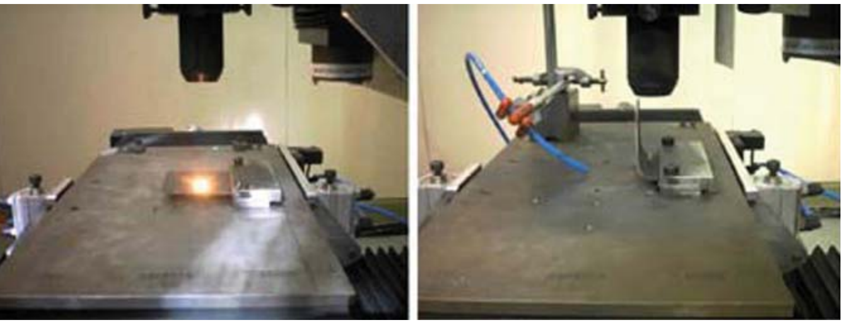
4.4.5 Bendable Strips

Long used in the shipbuilding industry, the application of thin strips of material over a more complex rib system has proved quite successful in producing complex forms that exhibit a smooth and flowing surface. It is of interest to note that spline curves so readily used in digital modeling today stem from the naval arena where thin strips of material will bend into a defined shape when attached at the ends and specific points in between. The bendability of thin strip materials often requires that the surface be composed of broad flowing forms without abrupt surface deviations which coincidentally prove appropriate for large surfaces from ship hulls to facades of buildings. Digital models that utilize finite element analysis are useful here in that they can produce visualizations of primary stresses within the model which in turn can direct the placement of strips in an optimal manner. See Figure 59.

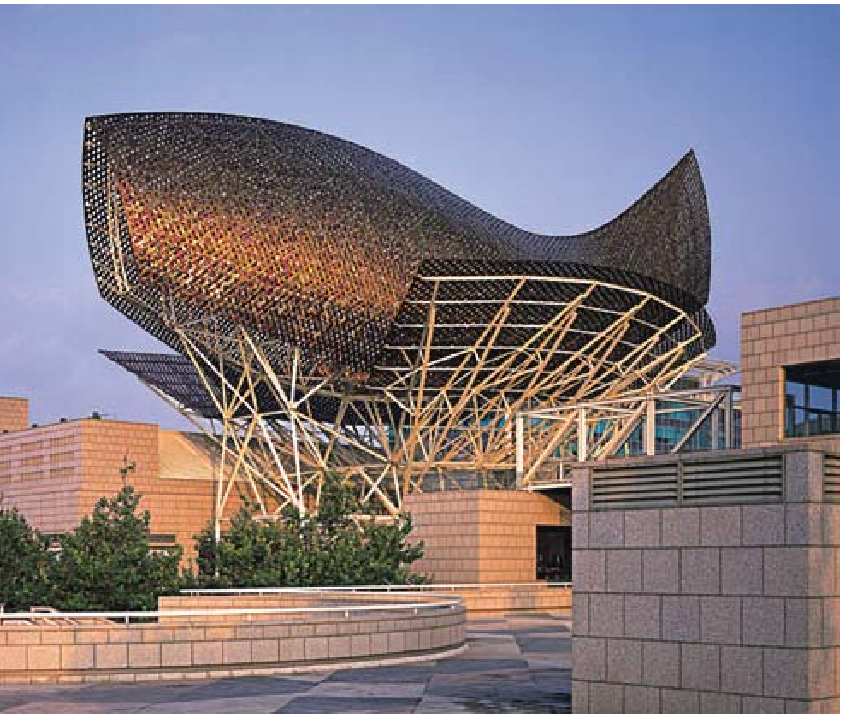
4.4.6 Aggregated Faceted Panels

To avoid the associated difficulties inherent in creating complex surfaces from non-developable flat sheets, architects have resorted to dividing the surface into a number of smaller units that consist of planar surfaces. These facets may take the form of triangles or various other shapes, but the key here is that their edge conditions are straight and as such both manufacturing and constructability are made easier. As the facets within the surface become smaller it is possible to produce a smoother finished product but this can come at the result of increased complexity, manufacturing and material usage. See Figure 60.

Digital modeling in this approach requires that a grid be applied over the model and suitable panel sizes are derived from the resultant of the intersection between grid and surface. Projection and mapping are two methods possible for defining the surface grid. Projection implies



58. Thermal Plate Forming.



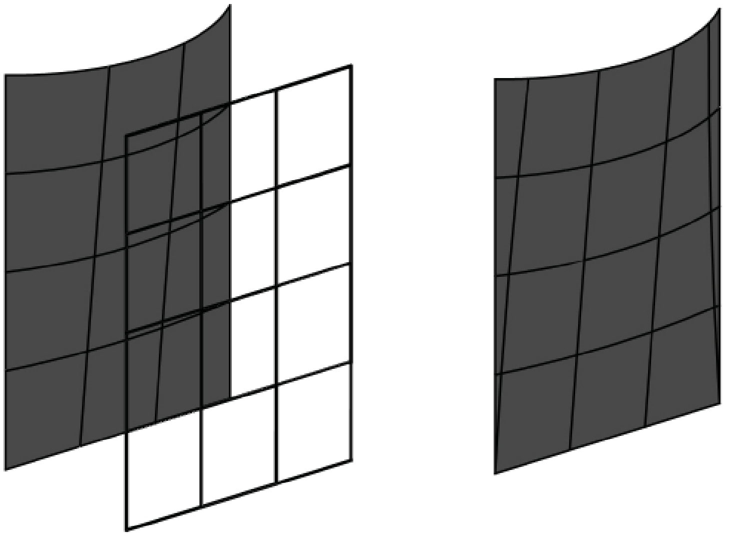
59. Fish Sculpture, Barcelona.

simply that, a planar grid is projected directly onto the surface. This produces panels that while looking identical in elevation are actually distorted in order to compensate for the surface curvature. Mapping essentially wraps the surface with the predefined grid arrangement. This technique has the advantage of maintaining the desired panel shape for ease of manufacturing however it may be necessary to modify the surface shape to accommodate the limitations of the panels in producing the desired complex surface. See Figure 61.

4.4.7 Shaped Primary Structural Elements

To maintain an architectural purity within a building that maintains a connection between inner and outer surfaces, it is desirable to produce a primary structural system that follows the shape of the exterior surface if not exactly then to a degree that minimizes the requirement of an elaborate secondary structural system. While it is relatively easy to accomplish these complexly shapes structural members in small scale applications such as in the automotive and naval sector it becomes much more complicated in a large scale building where the structural elements can be quite massive and difficult to form. Select rolling mills have the capacity to bend large steel sections in one direction but their capacity for out of plane twisting is quite limited. The bending machines suitable for circular sections have the ability to produce complex shapes although in practice the sections lack the required strength and stiffness to act as primary structural members. (Schodek 2005, p59-61) See Figure 62.

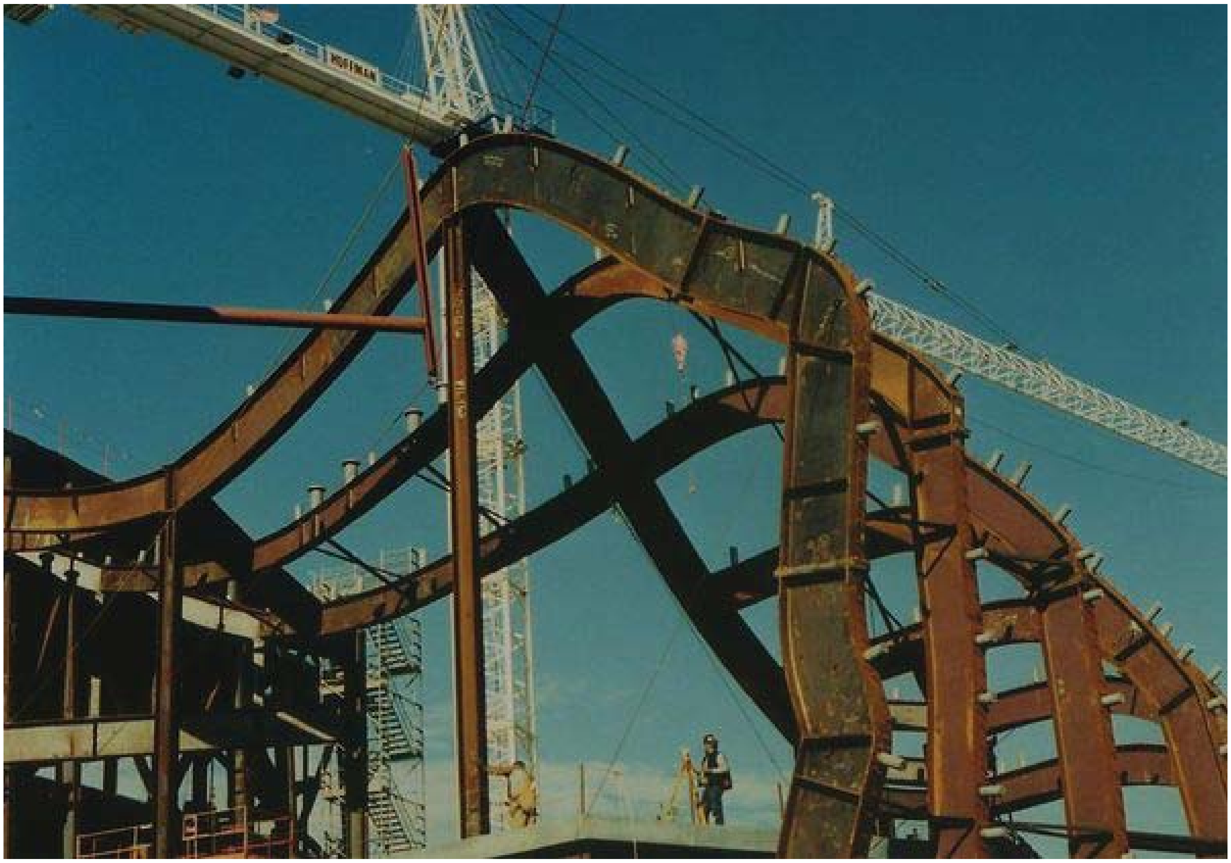
60. Swiss Re Headquarters, London.



A. Projecting a regular grid onto a curved surface results in panels of varying size

B. A surface is subdivided to generate surface panels of similar size

61. Surface subdivisions.



62. Experience Music Project, Seattle.

5.0 Design Proposal

The aim of this thesis, while attempting to develop an innovative way in which to create curvilinear structurally supportive building skins, strives to provide a method of design that encapsulates the iterative design process from schematic design to final construction. This means providing a novel way in which to design, document and build.

5.1 Design Approach

The design of a building requires the thoughtful integration of a rapidly expanding palette of structures, systems and construction approaches that if not considered early within and throughout the project can have deleterious effects when design changes occur downstream. Current design practices treat many systems, such as mechanical, electrical or structure to name a few, as separate entities that are designed independent of one another and occupy their own partitioned space. While this approach may be useful in relatively uncomplicated spaces, its appropriateness begins to diminish when the complexity of building structure and layout begins to intensify. At this point, a minor adjustment in one system may have dramatic effects on a neighboring system. Additionally, when using drafting programs that do not support a method for automatic updating of documentation then all changes require manual correction and update of relevant drawings which again, with complex buildings, can result in mistakes, omissions and an increase in man hours.

Nature's design process as stated in previous chapters utilizes a number of feedback systems to direct the growth and formation of an organism based on the internal and external forces acting on and within it. All systems are continually updated and act in concert with each other to provide optimum functionality at all levels of development. If this is applied to architecture there arise possibilities to streamline the design process in that multiple design concepts could be rapidly tested with minimal investment of time while allowing downstream changes in the selected model to be incorporated in a rapid and concise manner. This type of design is a partial possibility with building information modeling (BIM); however its capacity is limited with regard to the rapid changing of elements that are related to each other. In other words, necessary changes must be done on an element by element basis which, although translated into all of the relevant drawings, fails to allow for rapid building scale changes. Parametric design allows for this element relationship whereby changes to specific pre-defined parameters can influence any number of output variables.

The design component of this thesis utilizes an innovative program called Generative Components from Bentley Systems which is a powerful parametric, constraint-based modeler capable of designing in the aforementioned manner. While the program performs many necessary functions and is able to generate a variety of thesis objectives it is still under development and there are a number of additional requirements that are as of yet unavailable in the program but which will be addressed for further research and development. The key to success of the thesis will be an adherence to the philosophy of developing designs that are not based solely on visually driven designs but rather ones that include or are informed by intended modes of construction, the physical characteristics of the materials to be used, along with a biomimetic approach to spatial and structural coherence. This 'bottom up' development of architecture can be observed in the attempt to create forms that are derived from higher-dimensional geometry, where surfaces are defined in a strict mathematical sense and contain the prerequisite of material compatibility during the manufacturing process. (Lalvani 1999, p32)

5.2 Design Objectives

Before delving into designs it is necessary to define some objectives for those designs and establish what it is that will be accomplished in their generation. It is not a question of what is to be designed but rather what the design is to do and what can be derived from the design process that is of primary importance. The significance of this differentiation focuses on design approach rather than design outcome where the final solutions have the ability to affect multiple design scenarios instead of a singular example.

The two major objectives that form the basis for this thesis investigation are:

1. Develop a design process and documentation system that allows the AEC (Architecture, Engineering, Construction) community to work more effectively as a cohesive unit with regard to the digital design and physical construction of architectural projects.
2. Create a variable structural prototype unit that is able to conform to a variety of complex surfaces and whose form is derived from natural spatial and structural morphologies, the physical limitations and benefits of the intended construction materials, and the desired construction methods.

At this point a number of questions are raised in order to arrive at the key products to be realized at the end of the research. These questions evolved from a critique of current design approaches in a manner that elicits the possibilities for new outcomes.

1. **Why are current methods of building design and documentation inefficient?**

    a. The relationship between element, system and building are often disparate and multiple drawings are required to illustrate them.
    b. Changes in the design are not easily propagated through the drawing set which results in additional time and possibilities for error.
    c. The shift from sketch design to CAD development is a hard-edged threshold in which abstracted and generalized spatial and geometric ideas and relationships are rigidized into a one path directive.
    d. Initial measurements must be approximated which a priori necessitates later dimensional modification and ensures a built in time expenditure.

2. **Why are current methods for design and construction of non-orthogonal surfaces and structures so much more difficult to get built than linear surfaces and structures?**

    a. Complex surfaces and structures can require many uniquely shaped elements to attain their three dimensional conformation, therefore development time and manufacturing costs are elevated.
    b. The construction documents and actual process of construction can be very complicated which requires a highly skilled and knowledgeable workforce along with unique construction methods.
    c. Manufacturers are slow to adopt new production methods that would facilitate easier construction due to the requirement that new production and assembly methods as well as the logistical systems would require investment into new facilities and their associated cost implications and risk.

5.3 Design Requirements

From the above line of questioning it is possible to arrive at a number of conclusions as to what schemes need to be developed and how they can be adapted to the design of complex structural surfaces.

1. Revise the current method of design documentation and explore ways in which to more effectively organize the visual information conveyed.
2. Tailor the design and documentation phase as more of a feedback oriented method where minimal manual revisions are required to documentation when design changes occur.
3. Devise methods of generating complex surfaces that allow for elements that can be more easily designed and manufactured.
4. Create a system where the three dimensional form of an element will specify its location in the building with a minimal amount of measurement, positioning and labor.
5. Select ideas that maintain the quality and intent of the design while reducing the final cost of the project.

5.4 Design Methodology

A structured approach to the genesis and development of the desired thesis objectives is necessary to allow for their broad relevance to architectural constructions rather than their singular appropriateness for a given scenario. While this thesis seeks to provide exploratory physical manifestations of the design objectives it will also focus on developing an approach and method to design, manufacture and construction of architecture that will aid in producing more efficient and cost effective buildings.

Due to the nature of the investigations and their development from natural systems it is difficult and indeed undesired to separate their direction into discrete streams. As a result, an overlapping of conditions will occur where the same biological influences will aid in the advancement of multiple design products.

5.5 Design Drivers

With an idea of what is to be accomplished it is possible to look at natural systems that could begin to inform the design process. The selected principles of biomimetics chosen in *Section 3.2* are to be used as both inspiration for development of the thesis objectives as well as a yardstick by which to measure the appropriateness of the designs created.

1. Self Assembly
2. The Power of Shape
3. Resilience and Healing
4. Materials as Systems
5. Sensing and Responding

6.0 Thesis Resolution

The advancement of the thesis takes place on a number of levels that build upon one another where conceptual design, development and construction strategies provide a base for the creation of structural building skin prototypes. The first design concept will focus on outlining a design process that covers the entire range of an architectural project from schematic design to construction. This process will be developed and rely upon nature's methods of organization, instruction, and construction to provide a framework that will help to streamline the efforts in the Architecture, Engineering and Construction (AEC) community. The second and third design concepts will utilize knowledge gained both in the biomimetic design principles explored in *Chapter 2* as well as the organizational principles put forth in the first design to create prototype scenarios for adaptive, curvilinear, structurally-supportive building skins.

6.1 Design Concept #1 - Design Methodology

The process of design proceeds along a path from conceptual idea to final physical form. Although this is the preferred path for all parties involved this is often not the case. It is inevitable that during the development process a number of issues will arise that result in changes to anything from minor details to overall conceptual considerations. So then, it would be beneficial if the tools available for design were able to follow the lead of the designer in that they allow for a freedom of controlled design exploration as well as the ability to effectively document and describe the final form all the while utilizing a form or representation that can serve both equally.

The conception and development of the design itself where a dynamic digital model that can adapt to specific environmental conditions is favored over a static, unchangeable one that suits only the context into which it is placed and loses its adaptability in subsequent projects. While it is not expected that one design model will suffice for all subsequent design explorations it is desired that a design scenario will arise in which discrete portions of a design may be brought together in different configurations to produce new and varied morphologies without starting from a blank slate each time.

Some of the most technically and structurally intricate and emotionally evocative forms originate in nature from a relatively simple set of instructions. This scenario arises from physical limitations that exist in the natural environment. Organisms must constantly compete for natural resources which can occur in limited supply within an ecosystem and as such there arises and in-built need for both material and energy conservation. This requirement exists not only for the formation of the organism but for its continued survival. The simplest set of instructions required to produce a viable organism is a necessity in that it reduces the physical size of the molecules that contain them. Additionally, a reduction in the number of instructions automatically reduces the number of possible errors that can arise as well as the investment of energy required to correct them. So then, it can be said that natural organisms have through their development evolved informational and constructional scenarios that create maximal functionality from a minimal investment of energy.

Any attempt to reduce the complexity required for the realization of man made constructions can benefit from an investigation into how nature deals with its own architectural documentation and process of design. To this end, it was at the molecular scale where the necessary directives were found. The process whereby segments of DNA, which cells transcribe into RNA and translate, at least in part, into proteins is able to contribute a number of ideas directly related to the way in which architectural documentation can be more effectively prepared and related to the design of a structure.

6.1.1 A Natural Order

'Cells are inventive architects…To build these elaborate structures…one can find examples of any engineering principle in use today. Fences are built, railways are laid, reservoirs are filled, and houses are constructed complete with rooms, doors, windows, and even decorated in attractive colors. Lap joints, buttresses, waterproofing, reinforcing rods, valves, concrete, adhesives – each has a molecular counterpart.' (Goodsell 1996, p81)

Organisms carry within their genetic makeup the instructions for complete self assembly. The process of self assembly does not occur in a vacuum however and the growth and final form of the organism is based on the static genetic sequence as well as the dynamic forces both internal and external which impose themselves on the organism.

Section 2.1.1 DNA and Genetic Coding explained how the genetic code is relatively deficient in the full complement of instructions that appear necessary to build complicated organisms. From this it was concluded that rather than encoding for each cell separately there are a number of design principles that allow for development based on a set of growth parameters and strategies that reduce the complexity of organic formation. If this is the case, then it follows that there is some innate flexibility in the design outcome whereby the instructions in the set define the parameters for development rather than defining a rigid model for growth. In other words, while the instructions for full, functional development of an organism are contained in its genetic code, the final form of the organism is directly influenced by the internal and external factors acting on and within it. Diet, environment, physical stresses and a host of other factors influence the direction of growth and ultimately the final outcome. Architecture and its creations are similarly influenced by a set of developmental factors such as program, budget, siting, etc., that must all coalesce into a final built form. There is no absolute resolution to these factors, only an attempt to best balance the necessities of each so that the product approaches the ideal or desired outcome. Often times a variation in one of the factors influencing the design will have implications whether positive or negative for the entire collection. A decrease in budget, for example, may require the reduction or elimination of certain elements that are deemed non-essential.

If we are to envision the design process for a building developing in this manner then it will be beneficial for reasons outlined above to reduce the number of instructions necessary for it to be designed and built. This can be accomplished in both an informational and physical manner. The key here is to reduce the number of instructions required to define the building so that necessary changes or alternative design scenarios can be executed with a minimal investment of time. The physical counterpoint to this is the utilization of natural design cues where the actual building elements are derived in such a manner that their three dimensional form helps to define their location and connection within the building thus reducing the number of instructions required for its proper construction. The method for natural development and assembly outlined below will help in creating a framework for man-made design, manufacture and construction techniques in line with a design process utilizing a minimum number of instructions.

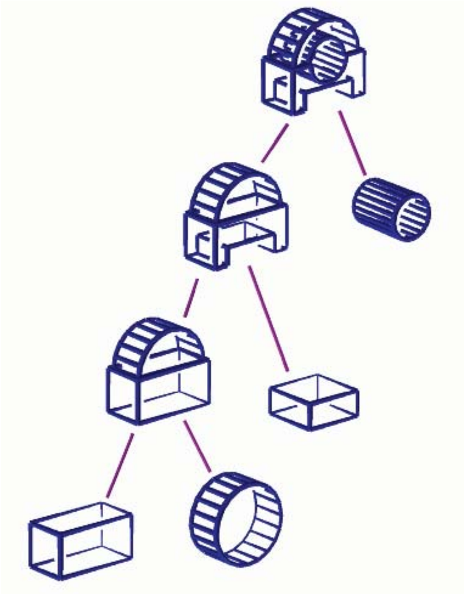
Section 2.1.4 Hierarchy of Structure illustrated how patterns are intrinsic to natural systems in that every component must not be looked at as an individual unit but as part of a collective whole. While treating the entire building as a complete unit may be a difficult task, the idea begins to clarify itself when we start to examine the various ways in which this may be possible.

A benefit with regard to design development or alteration that can be derived from *Section 2.1.4 Patterns* arises if the design approach is looked at as a hierarchical organization. Typical tree diagrams representing informational hierarchies proceed in a strict additive or reductive manner where one parent node will specify many children nodes or vice versa. See Figure 63. While these methods of organization are useful in their respective contexts such as hierarchical transforms or feature trees in solid modeling applications, their effectiveness diminishes when applied to the process of design itself. In real world design scenarios there may be instances where a node or particular design element will require input from a variety of upstream sources for its definition and it in turn may influence the definition of multiple elements. Here, the graph is still directed in that the relationships proceed from independent upstream nodes to dependent downstream nodes yet it provides a much freer approach to the relationships established between components. See Figure 64.

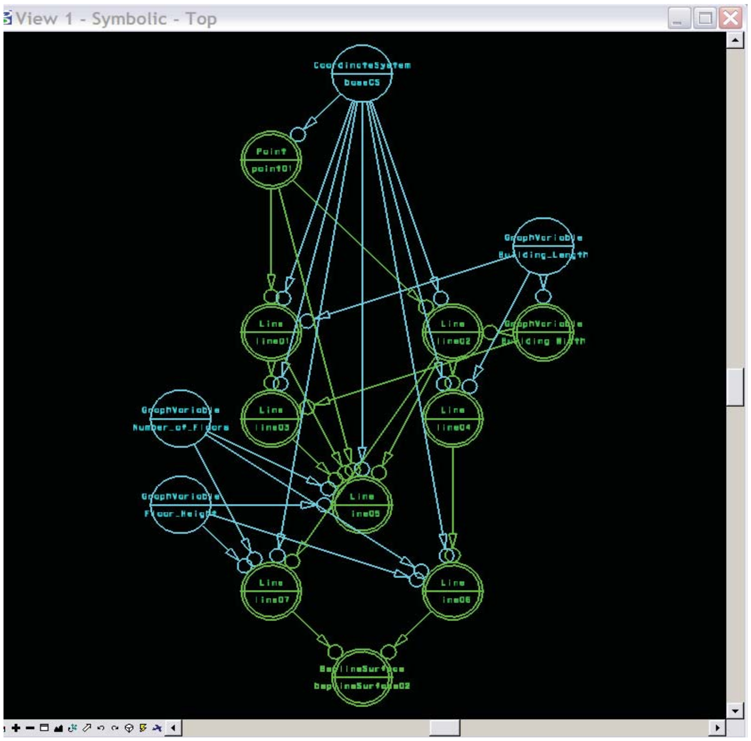
The design process as it relates to use, layout, structure and construction is often quite complex and requires a number of iterations to arrive at a viable final design. Often, the progressive development of these design iterations will occur with digital models that have been translated into physical models for hands on manipulation and then digitized back into the computer for further development. While this process does work quite well it has the drawback of not being backwards compatible, that is, once the design is changed in the physical model and digitized back into the computer, the previous digital model becomes redundant. By infusing the project with an approach that parameterizes the relevant design variables, changes that may be necessary, whether they be structural or aesthetic, have the ability to be changed within the digital model. A model with parameterized design variables has the benefit of reducing the amount of remodeling that is necessary for each design iteration. In fact, each modeling instruction or set of instructions can, like a gene in natural organisms, be turned of or on to express or hide its function. Changes to the design parameters are thus reversible and time is not lost if a previous design direction is to be revisited. It should be noted however, that the model must be properly developed so that any modeling instruction that is turned off will either have a corollary to takes its place, or that its absence will not result in downstream errors. Any results obtained from analysis of the model by other related design disciplines that require a change in the design would be quickly expressed and tracked in the program code.

6.1.2 The Relevance of Parametric Design

CAPD (Computer Aided Parametric Design), as it is referred to for the purpose of this discussion, can begin to emulate the natural process of growth and development by allowing relationships between design variables to be created so that they can influence each other according to prescribed methods of interaction. In this way the design is able to respond to manipulation of parameters that coincide with developmental forces driving the design. A closed feedback loop is created for model generation, sequencing, alteration, visualization and construction that effectively overcomes the inherent inability in the majority of CAD software to do the same. This feedback enables the designer to reduce time in varying and in turn manually revising changes in the design. Additionally, and in keeping with evolutionary theory, albeit on a condensed timeframe, CAPD allows for the simultaneous development of multiple designs within the same model with the possibility for selection of the most appropriate once they have all been examined. This type of parametric design enables the designer to create dependencies (relationships) anywhere within the model and between design



63. Tree diagram showing typical hierarchical relationship. for solid modeling operations.



64. Tree diagram showing a composite hierarchical approach.

components. The size of a duct shaft may be dependent on the area of the first floor which is in turn dependent on the number of floors that are proposed for the building. Alternative parametric approaches exist albeit on a more simplified level where relationships exist between components that physically interact with each other as with walls and windows for example. If the wall is moved the window will move with it. An ideal parametric design system would effectively encapsulate both the broader project sized parametric associations and the more specific building component relationship methods.

6.1.3 Parametric Correlation

With a parametric digital design system an issue arises between bottom-up and top-down design styles. The bottom-up method contains within it some vision of the overall project design and seeks to resolve this design through a gradual development and integration of building elements into a larger whole. The top-down method approaches the design in a different light where there is an initial development of the whole scheme with subsequent subdivision into its appropriate subcomponents. A composite approach to design would most likely be required in that to effectively establish a set of hierarchical component relationships it is necessary to have an idea of the final product. However, it is difficult to model an approximate final form without first defining the parameters that allow for sequential variation and the building of components from the bottom up. The usefulness of a parametric design system quickly becomes apparent when it is realized that both the final form and the subcomponents are variable.

6.1.4 Generative Components

6.1.4.1 An Outline

This thesis makes use of a parametric digital design system called GenerativeComponents (GC) by Bentley Systems Incorporated that runs in their Microstation design environment. The unique character of GC arises from its ability to allow for and promote extremely customizable parametric and associative design solutions. Parametric design in this case refers to a method of design that establishes dependencies or associations between design elements. This means that the behavior of specific components of a design whether they are walls, cladding panels or structural columns, are defined such that changes that occur in the design influence not only the element that is altered but all of the elements that are associated with that element. While the individual design components may range from a simply defined layout point based on Cartesian coordinates to a complex array of trusses that adapt to localized roof conditions, it is in their user defined associations to one another that makes GC parametric design so powerful. The designs created in GC are dynamic instruction sets that are developed with an understanding of what the end result is to be without the need to have this vision fully realized. The parameters and associations that are defined within and between components allows for a variability of design scenarios based on the conscious implementation of these by the designer. In contrast to standard 2D and 3D design programs that create static models and require a large input of time to explore and implement variations, GC is able to rapidly incorporate these changes into the existing model being used while still maintaining the full functionality of the previous iteration if it is to be revisited in the future. Additionally, GC allows for a scalability of complexity with regard to the clarity of the design at any point within the process. Early on in a project when many variables are unknown GC is able to create a framework that allows for an exploration of design intentions without defining these intentions in a rigid manner. If one or any number of the design parameters need to be revised then they will be instantly updated and these changes will propagate through the model to align it accordingly. When the project has developed to a point where an increased desire for geometric accuracy is required, then it is possible to do so with minimal input. While GC allows for a high degree of freedom with regard to design exploration and final solutions it should be noted that the amount of flexibility inherent within the design is a function of the way in which the designer has created the model. The program itself becomes most useful when the designer is able to logically establish a design hierarchy that is variable based on their intuition and the requirements or restrictions imposed by the chosen method of manufacture and construction. GC is able to play a key role in each step of current design methodology from concept genesis to design development to rapid prototyping and digital fabrication to the final export and management of construction documentation all of which are instantaneously variable and updateable.

6.1.4.2 Programmatic Description

In order to fully understand the usefulness and applicability of GC with regard to this thesis it is necessary to outline the way in which the GC environment is organized and used.

GC is based on the creation of dependency relationships between individual design components where the output variable for one is related to the input definition of another and any changes that occur in the former will propagate to all of its associated downstream dependent components. The hierarchical structure that develops from these relationships forms what is known as a directed graph. The graph contains within it all of the dependencies between the associated components. GC displays this graph in a symbolic model view which is very useful for allowing the designer to see a graphical representation of typically non-visual relationships as well as providing a tool that allows for others to quickly become familiar with the design intent and relationships. See Figure 65.

The components used in GC are able to exhibit multiple behaviors in that their input definition can vary depending on the desired function of the component. In this case a single point may define the preliminary layout position for the excavation of a building and may be based on the input of specific Cartesian coordinates while another point may represent the starting position for a cladding panel on a curvilinear surface whose position is defined by the intersection of structural elements. It is important to note that the designer can effectively change the input variables by which the point is derived without altering or influencing the downstream dependency structure of the components that are associated with it. See Figure 66.

Both the directed graph and the symbolic view are generated through actions initiated by the designer. These actions are performed through the definition of new features or design steps. New features may contain the addition or variation of one or many individual components. Once the desired amount of modification to the model has been added then the new steps are recorded as transactions. The sequence of transactions is recorded in a transaction file as program code and in a transaction view that graphically displays them. The importance of the transaction view is that it effectively displays for the designer a historical visual representation of the design progression as well as containing within it the necessary information to allow the program to build the model. See Figure 67. The user can step backward and forward sequentially through the design to revisit any feature that was created to determine its effectiveness, relevance or any other number of design questions. The transaction view is directly linked to the transaction file so that a user is able to open, view and edit in programming language (which is automatically generated from the transactions) any part of the file from the addition of new features to the rearrangement or consolidation of specific features. This ability allows the designer to move between conventional graphically based design into the realm of scripting and programming. The benefit of this flexibility is that it allows for the development and implementation of new components over and above the current palette of features contained within the base program.

6.1.4.3 Terms

In this section a number of the key terms used throughout the GC design system will be defined in order to aid in the understanding of subsequent writings. (Aish 2004)

Component Type – Refers to the collection of input and output properties and their associated update methods (explained below) as they relate to a specific geometric element or collection of elements that comprise a building component.

GC already includes a large collection of predefined components that include but are not limited to; Point, Line, Arc, BsplineCurve, BsplineSurface, Solid and modeling operations that allow for the creation of additional components.

Component Instance – The component instance refers to the actual usage of a specific component type in a particular feature of the model. The component instance is assigned a unique user defined name.

It is possible for the model to include a number of instances of a Point that are distributed throughout a number of transactions and are unique in their definition. Each instance of the Point could be assigned names such as mypoint, point01, yourpoint, etc.

Update Method – An update method refers to the way in which a component instance recalculates its output characteristics based on its input definitions.

For example, a Point can be defined by a number of update methods such as;

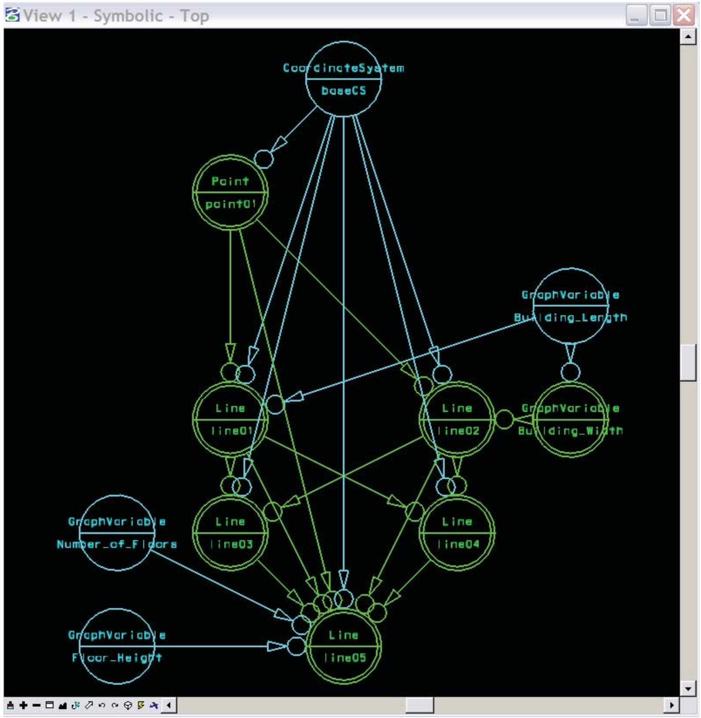
- AtCurveCurveIntersection
- ByCartesianCoordinates
- ByCylindricalCoordinates

The Point component has one update method for each point definition.

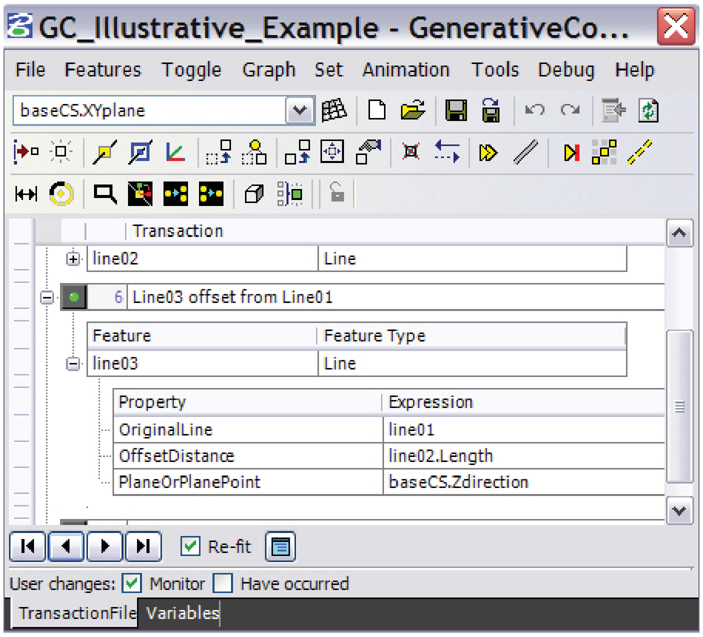
Property – Refers to the attributes of a component that combine to produce its current state. These attributes act as inputs for the update methods above.

A Point ByCartesianCoordinates will be defined by the following properties;

- CoordinateSystem
- Xtranslation
- Ytranslation
- Ztranslation



65. GC Symbolic View



baseCS.XYplane

Transaction

line02	Line

6 Line03 offset from Line01

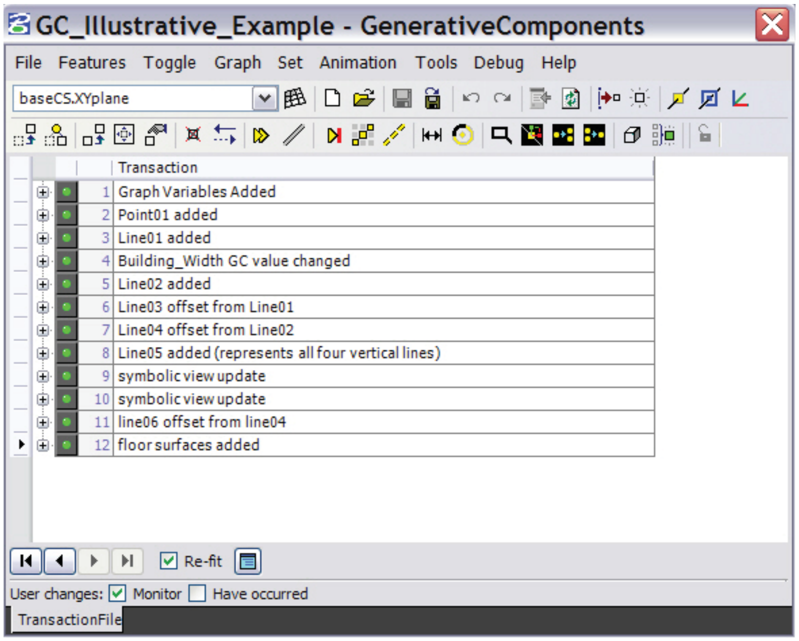
Feature	Feature Type
line03	Line

Property	Expression
OriginalLine	line01
OffsetDistance	line02.Length
PlaneOrPlanePoint	baseCS.Zdirection

☑ Re-fit

User changes: ☑ Monitor ☐ Have occurred

66. GC Line component and associated properties



	Transaction
1	Graph Variables Added
2	Point01 added
3	Line01 added
4	Building_Width GC value changed
5	Line02 added
6	Line03 offset from Line01
7	Line04 offset from Line02
8	Line05 added (represents all four vertical lines)
9	symbolic view update
10	symbolic view update
11	line06 offset from line04
12	floor surfaces added

☑ Re-fit

User changes: ☑ Monitor ☐ Have occurred

67. GC transactionFile view

The values for these properties are defined by an expression that satisfies the requirements for their input.

Property expression – This is the form of the input for the update method by which a property value is arrived at. GC is able to accept a variety of property expressions from something as simple as a single integer input to something more complex like a mathematical formula derived from the interaction of the property values from other component instances.

For example, a circle whose radius is defined by the property Circle01.Radius has the ability to contain a variety of expressions such as

Circle01.Radius = 5
Circle01.Radius = Line01.Length*5

Property Value - The property value represents the result of the latest recalculation of the property expression.

Graph Variable – A graph variable can be created that defines a value for use within the property expression of a component or any number of components. By changing the value of the graph variable all of the components associated with it will recalculate their values.

For example, a graph variable called line_length can be created that defines the length of Line01 from the previous example. The value given to the line_length variable can be an integer, a real number, a conditional statement, or a string. If the value of the graph variable was set to 5, then the Circle01.Radius = Line01.Length*5 expression would result in a value of 25.

Dependencies (Associations) – GC maintains dependencies between features within the drawing. Simply stated this means that when defining a new feature the user has the ability to associate its position or any number of characteristics with any other feature or set of features in the drawing. If the parent feature is updated then any children features that are associated to it will automatically update themselves based on the user defined dependencies. We can use the length of a line as an example here where the line represents the length of a wall. We are able to define a number of points along this line that represent the position of potential vertical structural members. If the length of the wall is to be lengthened then GC will automatically change the position of the vertical members to satisfy the relationship to the line that the user predefined. At any point however, the user has the ability to change the dependencies if they require alteration. At this point the file will recognize the changes and alter the form of the model accordingly.

6.1.4.4 An Illustrative Example of the Generative Components System

This relatively simple example will help to demonstrate the visual and programmatic platform of GC. In this case a building will be developed with a variable footprint, number of floors, and floor height.

When the initial design of a building is taking place there are often a large number of variables that are unfixed and changeable. By carefully planning the strategy for the development of the building concept it becomes possible for the model to develop in a way that allows for relative freedom with regard to dimensioning. As the building develops the dimensions can be updated to reflect the final requirements.

When the GC program loads it runs within the Bentley Structural design program. The GC Graphical User Interface (GUI) appears as a floating window that can be repositioned as desired. In it are contained all of the functions provided by GC. Running behind the GUI is a palette of user defined windows that are able to display both the symbolic view as well as multiple graphic views of the 3D model. See Figure 68 & 69.

The premise for the symbolic view is to represent the computer model in a way that illustrates the dependencies that can exist between different features. Each feature is represented by a circle with a defining tag within it. Connectors join features that have relationships to each other. In a traditional CAD program an element, such as a line, is drawn from point to point but the line and points do not maintain a relationship to each other. The points or line may be moved while leaving the others unchanged. It is the coordinates of the elements that are recorded in these "non-associative" CAD programs not their relationships to one another. In a project where design changes can affect multiple drawings, traditional CAD programs are unable to update them automatically because they elements within them are not associated with each other. At this point the user must use a great deal of time in checking and cross-referencing drawings for accuracy. If changes occur frequently then it is possible to see where a great deal of time can be lost. The drawings produced from a GC model are associated and thus any changes that occur will instantly be propagated to all relevant drawings.

1) Defining the graph variables

A graph variable is created by selecting add in the GV view, defining the name of the new GV then inputting the desired output value and value limits if required. See Figure 70.

2) Developing the model

Once the GVs have been defined it is possible to begin creating features that will visually represent the building design. A base Point01 is defined that corresponds to the primary layout point of the building. This point is defined choosing from a number of Point instances, in this case a point ByCartesianCoordinates that uses the base coordinate system baseCS as its input coordinate system and X,Y,Z values of 0 (null) to place the point within the baseCS. See Figure 71.

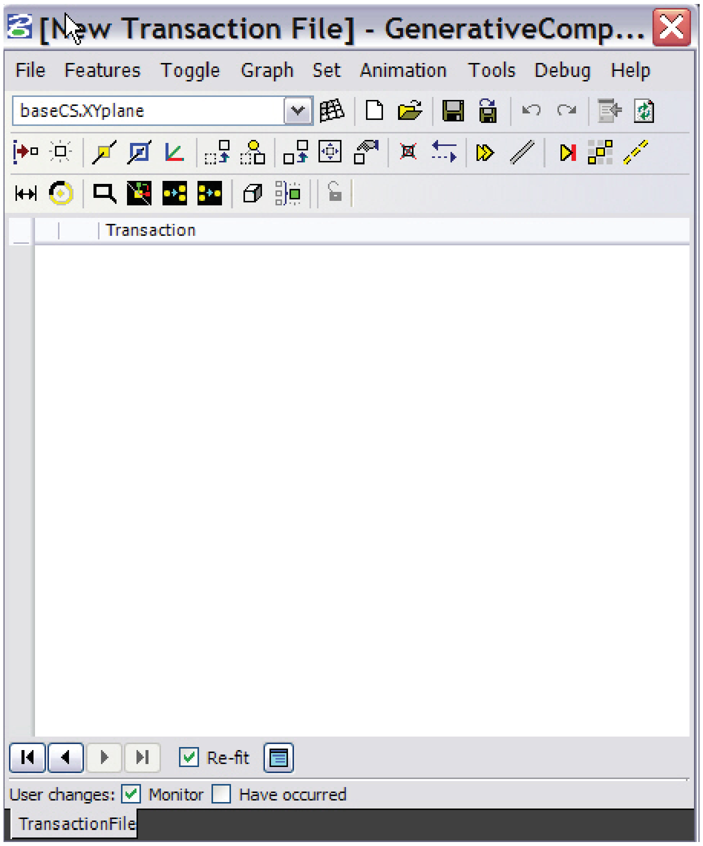
Point01 is now defined in a number of areas within GC. It appears in the graphic view as a graphic representation, in the symbolic view as a representation of its associativity to other components in the file, and in the GUI transaction view as steps in the transaction list which represent the design history. See Figure 72. Lines representing the length and width of the building can be constructed next. The lines will be dependent upon Point01 and the baseCS. The first Line01 is a line ByStartPointDirectionAndLength which uses the GV Building_Length as the property expression for its execution. See Figure 73. The length of the building is now parametrically dependent on the value contained within the GV. Any time the building length needs to be changed it can be done quickly by sliding or manually inputting a new value into the Building_Length GV.

Consider, for sake of proportion, that the width of the building is desired to be one half its length. It is possible then to define the value for the Building_Width as Building_Length*0.5. Having originally set the value for the Building_Width as a default value of 10 the change that is made to it will add another transaction statement. Each transaction statement is given a default name of *Graph Changed By User* which is editable for the user to define the actions taken in that transaction. If for some reason the user wishes to unlink the building length and width then it is possible to suppress the change by right-clicking on it and selecting suppress. This will change the GV value back to its original state. See Figure 74.

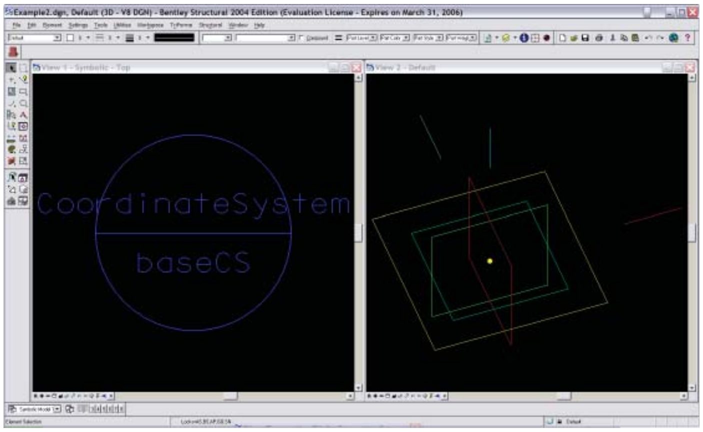
Line02 will be defined in the same manner as Line01 however it will use the newly edited Building_Width GV as its property expression. Now that both lines have been defined they can be played in the transaction file and they will now appear in both the symbolic and graphic views. In the symbolic view it is possible to see in graphic form the logical associativity of the developing model. The baseCS is situated at the top with Point01 and Line01 and Line02 directly associated with them. The GV Building_Length is associated with Line01 and Building_Width. The Building_Width is associated only with Line02. See Figure 75.

As the model and transaction file develop the symbolic view will develop alongside them to aid the user in keeping track of the logical order in which the design is progressing. The next step is to define the opposing lines defining the length and width. This is done by offsetting a new child line that is associated with the values of the parent. At this point all of the lines are dependent on the Building_Length GV for their definition. See Figure 76.

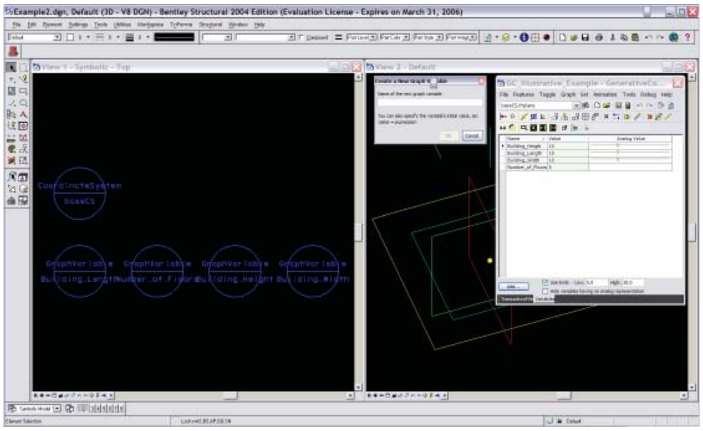
To add the lines representing the four vertical corners of the building it is possible to do so by defining their origin points as the end points of the plan lines. This will allow the vertical lines to realign themselves if a plan change is made. The feature used is a line ByStartPointDirectionAndLength but the uniqueness here lies in the definition of the origin point which is not a single point but three of the planar end points and Point01 thus creating four lines. This allows one feature to create four lines all editable with one variable. In this case the length expression is defined by the Floor_Height GV multiplied by the Number_



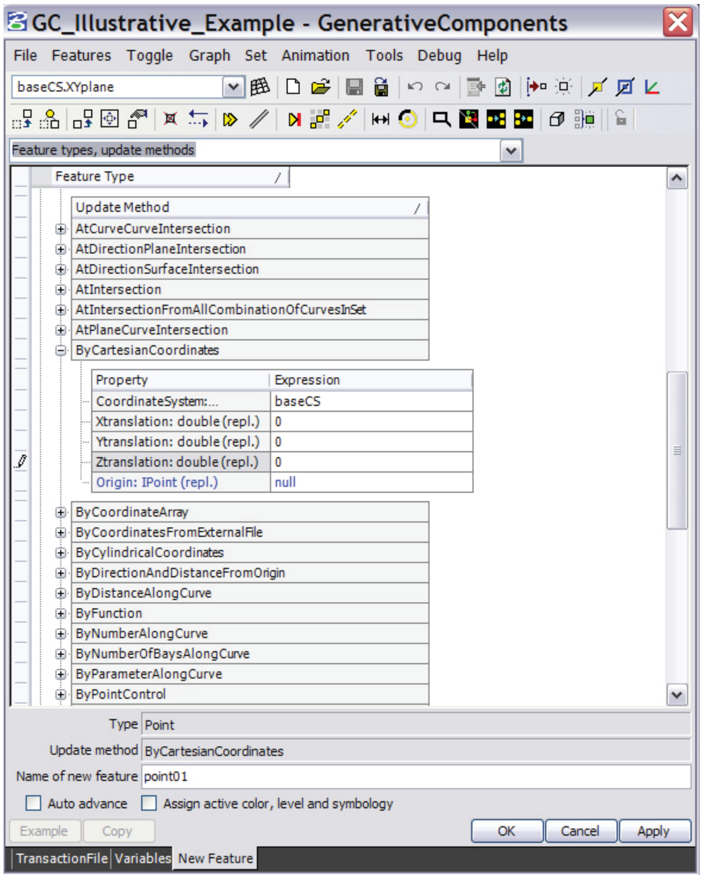
68. GC Graphical User Interface (GUI)



69. GC Symbolic view and Model view



70. Definition of Graph Variables.



Feature Type

Update Method

- AtCurveCurveIntersection
- AtDirectionPlaneIntersection
- AtDirectionSurfaceIntersection
- AtIntersection
- AtIntersectionFromAllCombinationOfCurvesInSet
- AtPlaneCurveIntersection
- ByCartesianCoordinates

Property	Expression
CoordinateSystem:...	baseCS
Xtranslation: double (repl.)	0
Ytranslation: double (repl.)	0
Ztranslation: double (repl.)	0
Origin: IPoint (repl.)	null

- ByCoordinateArray
- ByCoordinatesFromExternalFile
- ByCylindricalCoordinates
- ByDirectionAndDistanceFromOrigin
- ByDistanceAlongCurve
- ByFunction
- ByNumberAlongCurve
- ByNumberOfBaysAlongCurve
- ByParameterAlongCurve
- ByPointControl

Type Point

Update method ByCartesianCoordinates

Name of new feature point01

☐ Auto advance ☐ Assign active color, level and symbology

71. Definition of Point01.

of_Floors GV. It is possible to see here from the symbolic view how Line05 is directly associated with a number of other components and that the definition of Line05 which represents four physical lines in the building model is defined by the property expressions of those other components. See Figure 77

The final portion of the exercise is to define the individual floors and the roof which is completed in two steps. The lines defining the building width are created by a Line ByOffset from the ground plane by a distance equal to the Floor_Height GV and the number of offset lines describing the floors and roof is generated by the Number_of_Floors GV. These operations can be seen below in the GC Script Editor which allows one to view the programming code that GC creates as the user develops the model in the transaction view. See Figure 78. The series property expression allows for a number of sequential values to be obtained through defining a lower and upper value that is divisible by a third value. For example, the following Series(0,5,1) would result in output values of 0, 1, 2, 3, 4, 5.

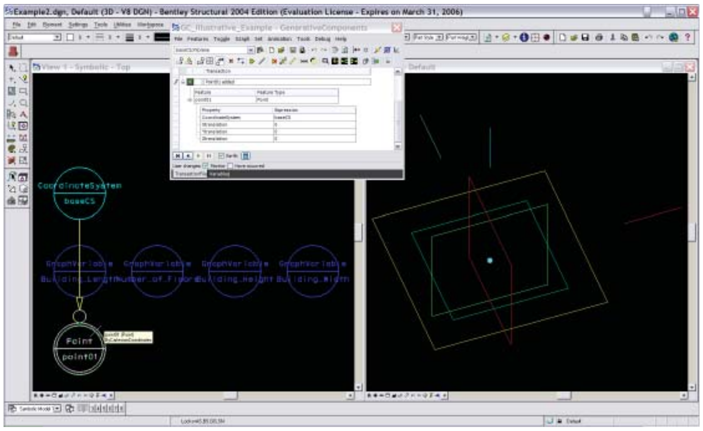
3) Refining the model

Once the GC script has been played through the final result can be viewed in a number of different ways according to the desired interpretation. The model view demonstrates the physical condition, the symbolic view displays the hierarchy of relationships and associations between building elements, the transaction view lists the historical order of operations used to obtain the product and the GCScript Editor shows the source code that can be further manipulated by the user. See Figures 79-82.

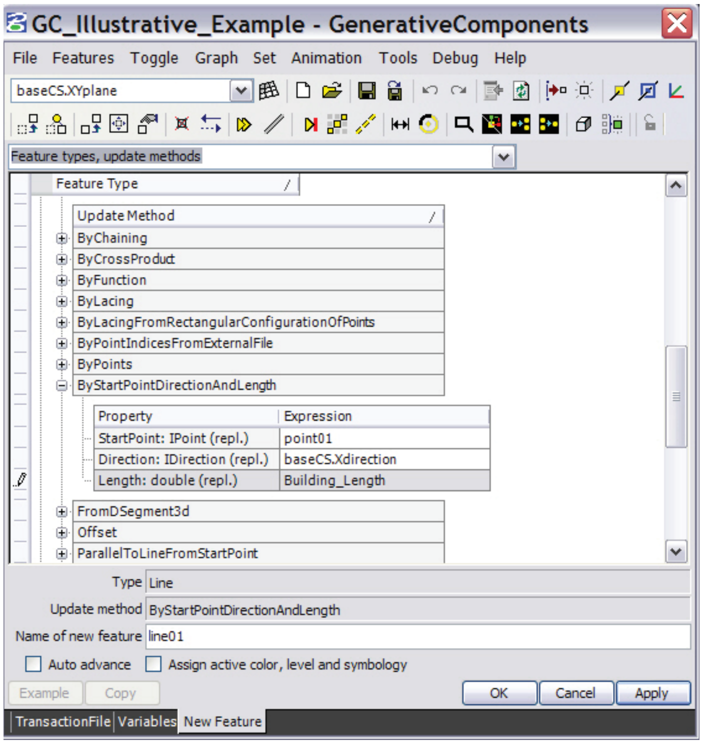
The final model produced here although simple in its geometric layout is very robust with regard to its instantaneous variability with relatively minor user input. With manipulation of just three numbers it is possible to vary the length, width, floor height, and number of floors within the building. The different model configurations realized in the following images were all created in less than one minute total time. See Figure 83.

4) Management and Export of Model for Construction

From this model a number of additional operations can be performed that streamline the AEC process. These can include fabrication planning for export to Computer Numerical Control (CNC) manufacturing, model prototyping, drawing extraction for setup of construction drawings, among others. Depending on the values assigned to the model the export products can be similarly used for physical models or full scale production. At the writing of this thesis however, not all of these additional operations are functional in GC.



72. Point01 in the Symbolic, TransactionFile and Model



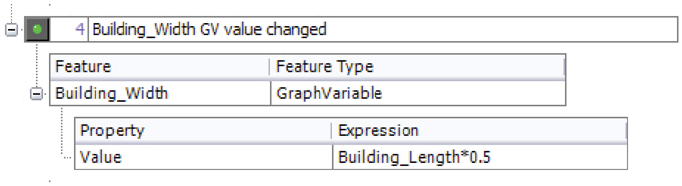
baseCS.XYplane

Feature types, update methods

Property	Expression
StartPoint: IPoint (repl.)	point01
Direction: IDirection (repl.)	baseCS.Xdirection
Length: double (repl.)	Building_Length

Type Line
Update method ByStartPointDirectionAndLength
Name of new feature line01
☐ Auto advance ☐ Assign active color, level and symbology

73. Definition and property expression for Line01.

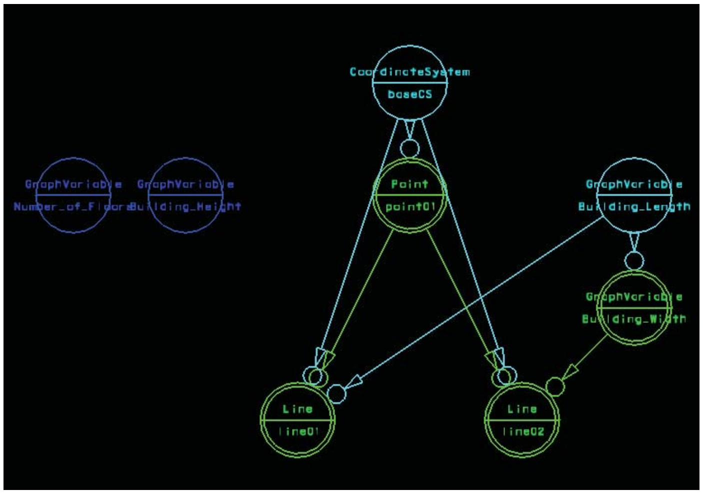


4 Building_Width GV value changed

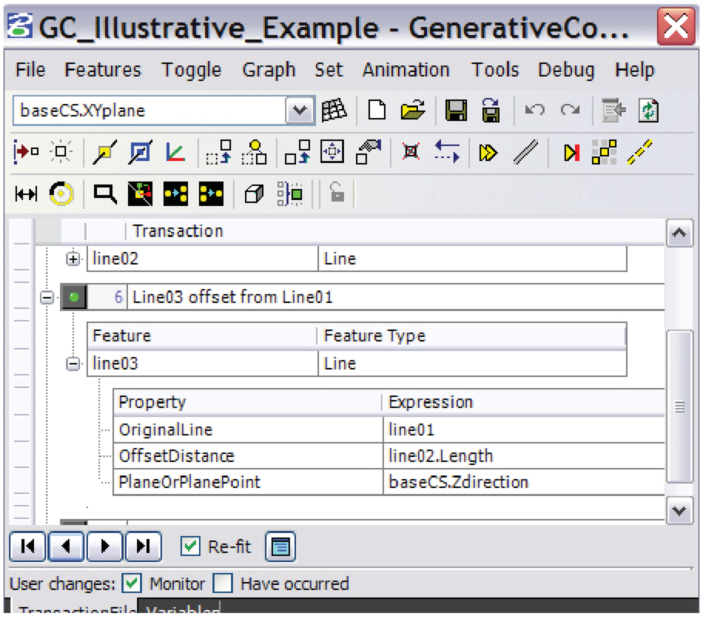
Feature	Feature Type
Building_Width	GraphVariable

Property	Expression
Value	Building_Length*0.5

74 Graph Variable Building_Width changed.



75 Symbolic view of component dependencies.



Transaction

line02	Line

6 Line03 offset from Line01

Feature	Feature Type
line03	Line

Property	Expression
OriginalLine	line01
OffsetDistance	line02.Length
PlaneOrPlanePoint	baseCS.Zdirection

☑ Re-fit

User changes: ☑ Monitor ☐ Have occurred

76 Offset of Line03 from Line01.

6.1.5 Parametric Modeling Based on the Biological Genome

William Lethaby writes in his *Architecture: an Introduction to the History and Theory of the Art of Building* from 1911 that "[s]ome day we shall get a morphology of the art by some architectural Darwin, who will start from the simple cell and relate it to the most complex structure."

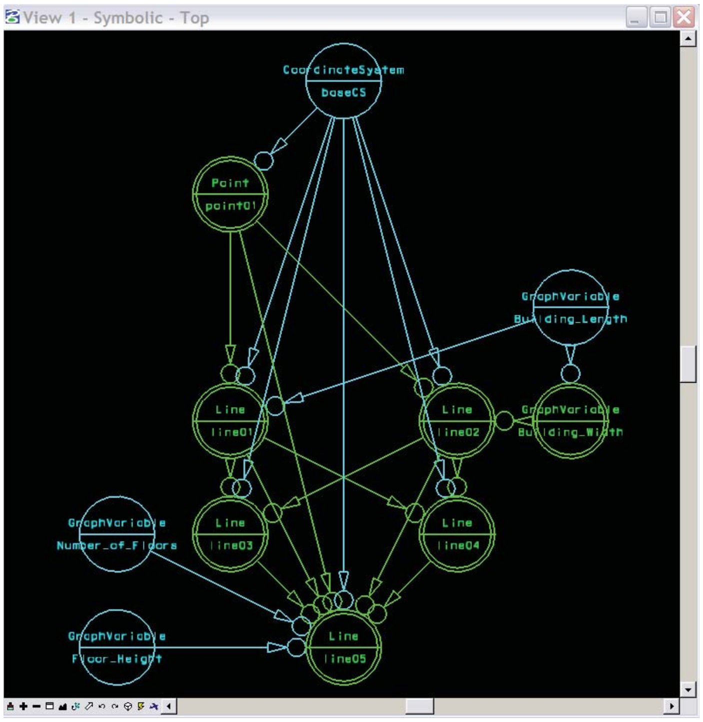
Genomic Background

All living organisms contain DNA which is a nucleic acid that contains the genetic instructions specifying the biological development of all cellular forms. The DNA molecule is composed of a vast sequence of nucleotide bases arranged into chromosomes which represent physically separate molecules. Each chromosome contains genes which are the principal physical and functional units of heredity. Genes themselves are specific sequences of nucleotide bases that encode instructions for the manufacture of proteins. It is the proteins that execute most biological functions and comprise the majority of cellular structures. Proteins are large molecules composed of smaller amino acid subunits. Unique chemical properties characterize the twenty different amino acids and it is these properties that cause the protein molecule to fold itself into various three dimensional structures that perform a particular function within the cell.

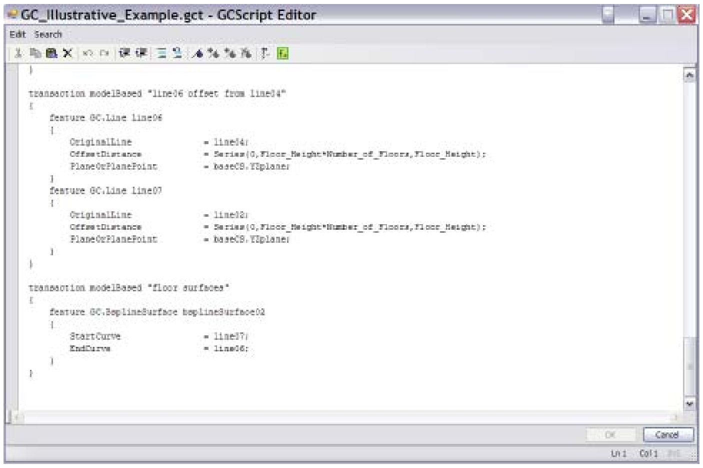
The amalgam of all proteins in a cell is referred to as a cellular proteome. The entire collection of all cellular proteomes in an organism is referred to as the complete proteome. While the genome is relatively unchangeable, the proteome is quite dynamic and undergoes constant changes in response to numerous intra- and extra-cellular environmental influences. The chemistry and behavior of a protein is derived from the static gene sequence and by the influence of other proteins in the cell which it encounters and with which it reacts.

The process of creating a protein from a segment of DNA is one that follows a path from informational to physical. A sequence of instructions creates a physical molecule. If we delve a little deeper into how this mechanism operates certain rules develop that can be relevant to architectural design practices.

Erwin Schrodinger, the famous physicist, published a book in 1944 entitled *What is Life?* In his book he posited that chromosomes contained what he referred to as the "hereditary code-script" of life. He noted however that "…the term code-script is, of course, too narrow. The chromosome structures are at the same time instrumental in bringing about the development they foreshadow. They are law-code and executive power – or to use another simile, they are architect's plan and builder's craft – in one." He envisioned the dualistic nature of these elements to be intertwined in the molecular structure of the chromosomes. Through an understanding of the molecular structure it was then possible to understand both the "architect's plan" and the eventuality

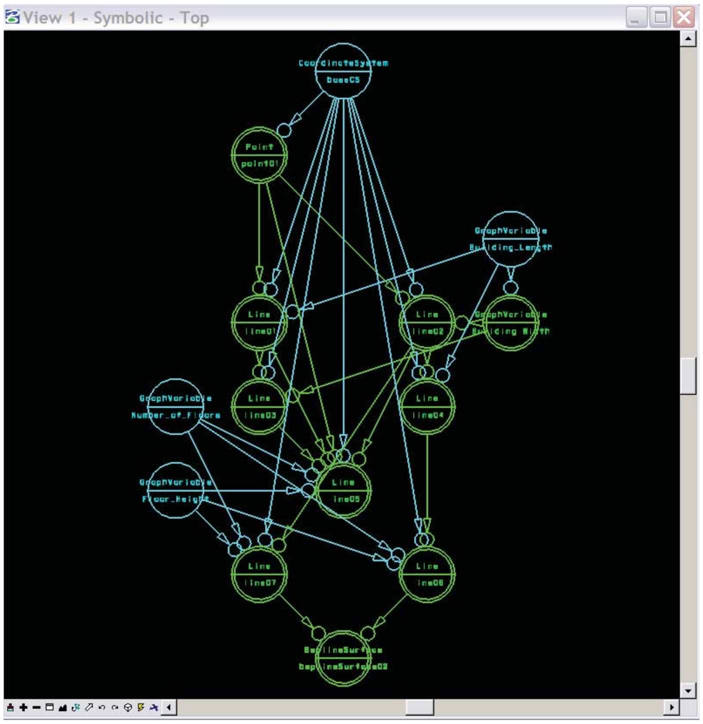


77. Symbolic view of model and dependencies for Line05.

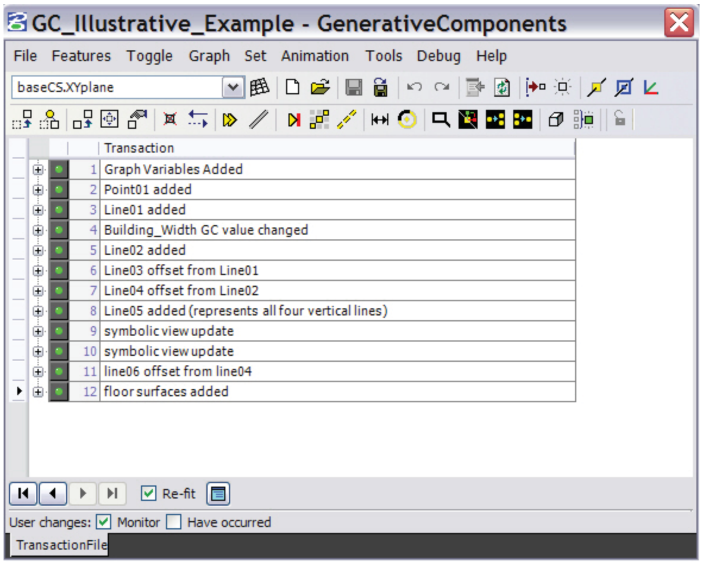


```
}

transaction modelBased "line06 offset from line04"
{
    feature GC.Line line06
    {
        OriginalLine             = line04;
        OffsetDistance           = Series(0,Floor_Height*Number_of_Floors,Floor_Height);
        PlaneOrPlanePoint        = baseCS.YZplane;
    }
    feature GC.Line line07
    {
        OriginalLine             = line02;
        OffsetDistance           = Series(0,Floor_Height*Number_of_Floors,Floor_Height);
        PlaneOrPlanePoint        = baseCS.YZplane;
    }
}

transaction modelBased "floor surfaces"
{
    feature GC.BsplineSurface bsplineSurface02
    {
        StartCurve               = line07;
        EndCurve                 = line06;
    }
}
```

78. View of GC Script Editor and relevant programming code.



79. Symbolic view of component dependencies.



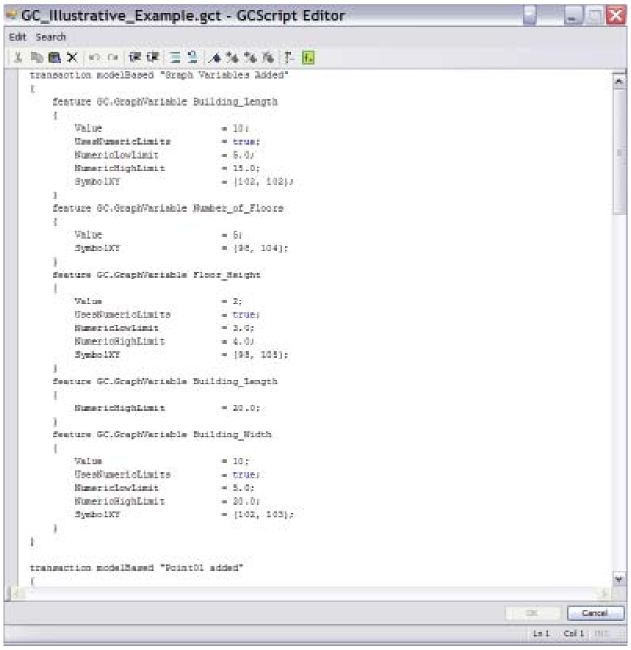
	Transaction
1	Graph Variables Added
2	Point01 added
3	Line01 added
4	Building_Width GC value changed
5	Line02 added
6	Line03 offset from Line01
7	Line04 offset from Line02
8	Line05 added (represents all four vertical lines)
9	symbolic view update
10	symbolic view update
11	line06 offset from line04
12	floor surfaces added

80. TransactionFile view

produced through the "builder's craft." (Schrodinger 1944)

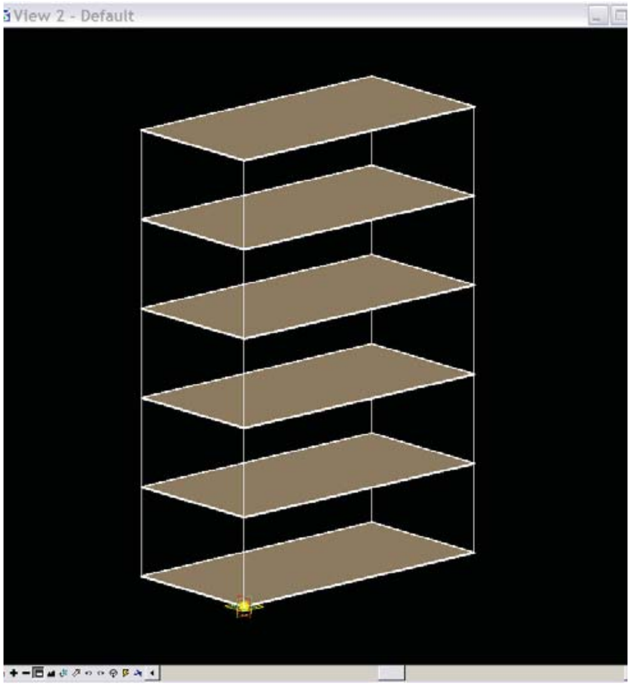
DNA – The nucleotide sequence is relatively fixed and unchangeable containing within it all of the instructions to build an organism. As noted previously the number of cells contained within the human body is 10,000 times greater than the number of instructions contained within the DNA sequence. The human genome therefore has developed ways in which to produce an incredibly complex form from a comparatively small instruction set.

When a project is ready for construction the design documentation and digital models for the project must be able to fully explain and instruct all parties involved as to how it will be constructed. Ideally it would be preferred to have one CAD database that could handle every aspect of the project including visualization, documentation, structural and material optimization, and export for manufacturing. Although a large amount of planning and organization is quite helpful in carrying a project along it is in the approach to design and the design itself where novel methods lead to efficient outcomes. Taking inspiration from natural reductive instructional and generative techniques as outlined in *Chapter 2,* such as patterning, bilateral symmetry, multiplicity of function, size correlation and inbuilt redundancy it becomes possible to reduce the complexity of architectural design at its outset. The approach to a design and its realization should be viewed as a logical progression where steps taken to reduce the complexity of the design process

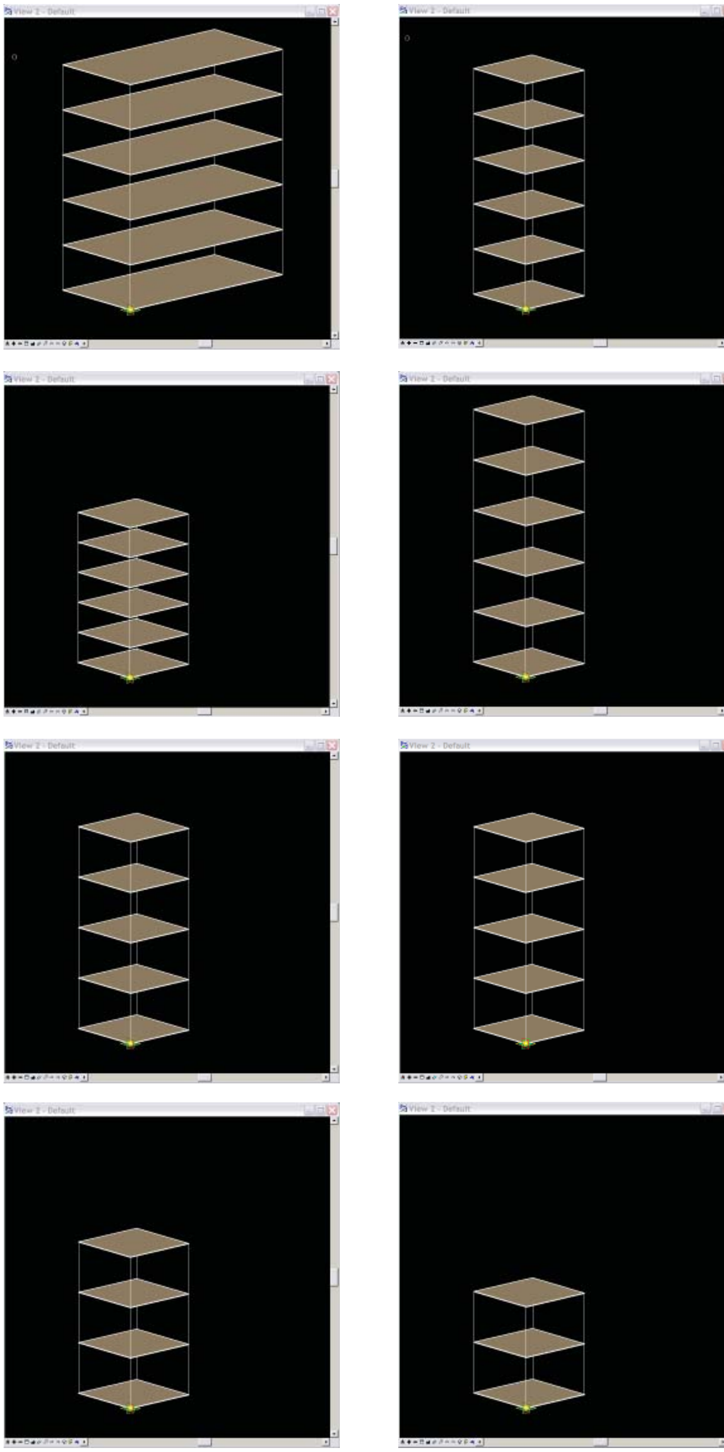


```
transaction modelBased "Graph Variables Added"
{
    feature GC.GraphVariable Building_Length
    {
        Value                   = 10;
        UsesNumericLimits       = true;
        NumericLowLimit         = 5.0;
        NumericHighLimit        = 15.0;
        SymbolXY                = {102, 102};
    }
    feature GC.GraphVariable Number_of_Floors
    {
        Value                   = 5;
        SymbolXY                = {98, 104};
    }
    feature GC.GraphVariable Floor_Height
    {
        Value                   = 2;
        UsesNumericLimits       = true;
        NumericLowLimit         = 3.0;
        NumericHighLimit        = 4.0;
        SymbolXY                = {98, 105};
    }
    feature GC.GraphVariable Building_Length
    {
        NumericHighLimit        = 20.0;
    }
    feature GC.GraphVariable Building_Width
    {
        Value                   = 10;
        UsesNumericLimits       = true;
        NumericLowLimit         = 5.0;
        NumericHighLimit        = 20.0;
        SymbolXY                = {102, 103};
    }
}

transaction modelBased "Point01 added"
{
```

81. GCScript Editor



82. Model View.



83. Symbolic view of component dependencies.

early on will greatly reduce the complexity of the design product in the later stages.

Chromosomes – Segments of DNA containing different instruction sets. If the complete DNA sequence were to be physically laid out in a line it would measure approximately two meters in length. (McGraw 1999) Obviously this incredible amount of information can become unwieldy if there is not an efficient way to organize and utilize it. In this manner the genomic information is separated into a number of chromosomes containing a different subset of the complete DNA sequence with each being responsible for producing a different set of functional products. The division of instructions also allows the cellular mechanisms to perform a number of processes on individual chromosomes all the while maintaining the full DNA sequence and full functionality of the cell. All of the chromosomes are contained within the nucleus of the cell as a unit. See Figure 84. This image illustrates a unique method for the visualization of the chromosomes and hence the discrete informational units of the genome where levels of detail emerge depending on the required depth and detail of information.

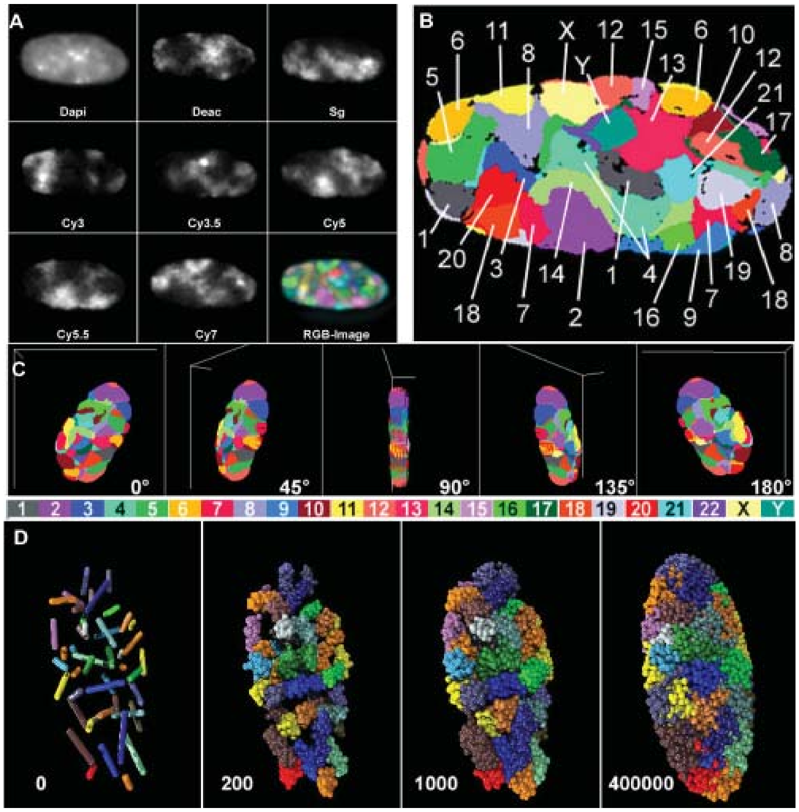
An architectural project must utilize the knowledge and resources of a number of different specialists like engineers, HVAC or daylighting, that help to develop specific areas of the design for incorporation into the final product. If we envision a digital system for the effective management of the enormous information being delivered by a variety of sources then each of these contributors can be thought of as chromosomal constituents. Rather than all working collectively the various groups involved would be able to work independently on fulfilling their own requirements yet still contribute effectively to the final form of the project. It would become unwieldy if every group involved in the project was required to work from the whole digital model. The file size and complexity of this model would quickly grow too large for efficient utilization. Different sectors of the AEC community utilize different programs for developing and analyzing their designs. A complex 3D model developed by an architect often contains extraneous information which is not required by the structural engineers who as a result must resort to building their own more simplified structural model. Ideally then, the building information contained within the digital database would exist on multiple levels of granularity so that each discipline could work effectively with it. Each design discipline would view and work with the model and the elements within it at the required level of complexity in that only a subset of the total building information would be visible. A beam for example may depending on its immediate graphical or analytical function be represented as a solid model for assembly, a finite line element for structural analysis, as source code for CNC operations or as a pure graphic for rendering purposes. The equivalent representation from biological modeling can be seen in Figure 85.

As a subset of the architectural portion of the design it is here that GC first comes into play.

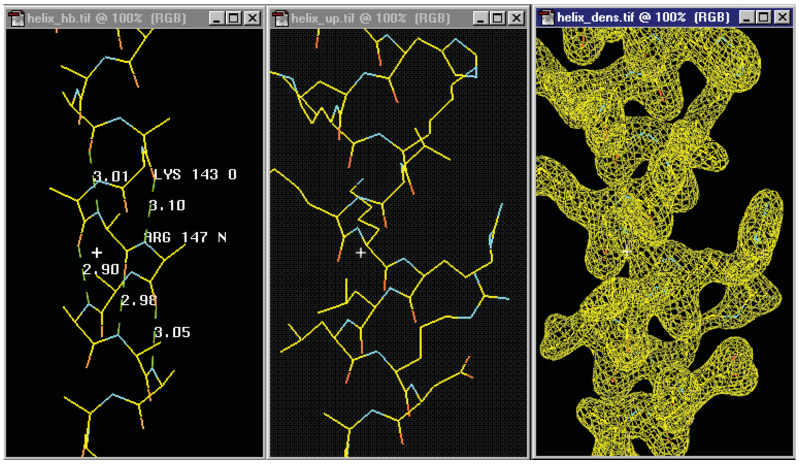
The program itself represents the opportunity for import/export to a number of other design and analysis programs as part of a large feedback loop. Depending on user input and the definition of new components, the GC design system is able to be refined for future use. In this regard GC essentially goes through one generation of development every time a new component(s) is/are created. Over time the program will grow in its ability to cater to the individual complexity associated with the various disciplines and firms that use it. At the same time there is an inbuilt capability of GC that allows individual components from different versions of the program to be exchanged if desired.

Genes – Each chromosome is further subdivided into a number of genes that are each responsible for encoding for individual proteins. This subdivision however exists on an informational level as the genes are all contained within the physical chromosome. This is the smallest informational unit within the genome that contains the instructions necessary for the production of a functional physical unit that aids in carrying out all of the functions within the human body.

If the chromosome represents each discipline involved in the progressive design of a project then the gene represents the information developed by and contained within these disciplines. The designs that they develop represent the transition from practice to implementation. As such the strategies used in this area are crucial in establishing a closed feedback



84. 24-Color 3D FISH (Fluorescence in situ hybridization) Representation and Classification of Chromosomes in a Human G0 Fibroblast Nucleus



85. Protein model showing varying levels of amino acid detail from left to right.
A) Hydrogen bonding in alpha-helix backbone
B) Image with additional side chains
C) Electron density image

system that is essential for a proper design to progress from design to construction. At this point the idea becomes craft.

All of the components contained within GC can be likened to the genes that enable an organism to be developed. Just as there are multiple **alleles** for eye color or hair color so too does a GC Point or other component contain multiple update methods that allow for unique geometric configurations. The programmatic **genotype** defines a specific **phenotype** and it is useful here to note that the expression of the phenotype is related to the interaction of the polypeptide gene products and the environment. This is one of nature's ways of allowing for diversity while still maintaining a fixed number of instructions. See Figure 86. Accordingly, the physical results rely on both the relatively static instruction set as well as the fluid influences imposed by the variability of environmental stresses. So too then it is useful if the digital environment can utilize a logical and ordered design palette that delivers multiple results based on unique combinations of components. There are a number of ways a point or surface can be derived, Figure 87, but it is in the way that the components associate with each other that influence how they behave. In this way a simple set of components can define a complex array of constructions.

Proteins – Complex molecules made up of amino acid subunits. Many proteins are **enzymes** or subunits of enzymes, catalyzing chemical reactions. Other proteins play structural or mechanical roles, such as those that form the struts and joints of the cytoskeleton or those serving as biological scaffolds for the mechanical integrity and tissue signaling functions.

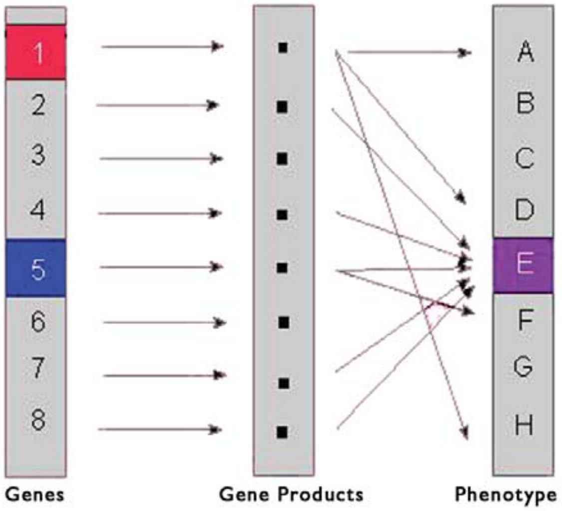
A protein is the functional manifestation of a polypeptide gene product where individual instances are assembled to create the final building form. It should be noted however that a functional protein may arise from a single polypeptide in its tertiary structure or from the assembly of two or more polypeptides into a quaternary structure. Protein construction proceeds along a path from primary to quaternary structure with increasing morphological complexity attained in each phase. Like the process of DNA to protein, so too does the four stage development of the protein itself proceed from informational representation to physical manifestation.

Primary Structure – The covalently bonded structure of the molecule. This includes the sequence of amino acids, together with any disulfide bridges. All the properties of the final protein form and function are determined, directly or indirectly, by the primary structure. Any folding, hydrogen bonding, or catalytic activity depends on the primary structure. See Figure 88.

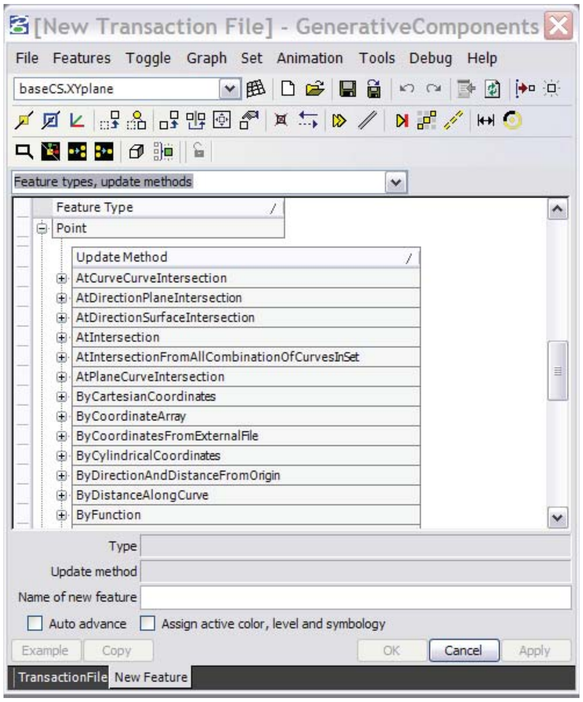
Primary Structure in Practice – The aim here is to begin developing a framework upon which the design and subsequent alteration of a building and its structure can be carried.
If the development of a design model in the

digital environment is to be useful in all stages of the design then it must be constructed in a logical manner that can be understood by all relevant disciplines and structured to allow for change. The adherence to a method of design that allows the history of the design and the instructions for its creation to be included and referenced for both progress and necessary changes is very powerful. Like the sequence of amino acids in the protein that are derived from the genes, Figure 89, the primary data structure of the specific architectural design file should exist as an entity within the digital program in that the code based instructions should specify all of the necessary information required to generate the desired components and model. In this case the transaction code within GC represents an ordered arrangement of the instructions necessary for progression of the design. See Figure 90.

GC contains within it a number of parametric instructional commands that define the shape of the structural elements and the final form of the structure itself. A symbolic view of the transaction script graphically illustrates the dependencies that each design feature has with regard to itself and its surrounding members. All of the subsequent physical generation of manufactured pieces and the final form itself are dependent upon the arrangement and instructions given within the transaction script.



86. Diagram of relationship between genotype and phenotype. The genes (1-5) on the left govern the formation of a gene product (1 gene - 1 polypeptide). A gene product can affect a number of features. A phenotype may be the result of the combined effects of several gene products.



baseCS.XYplane

Feature types, update methods

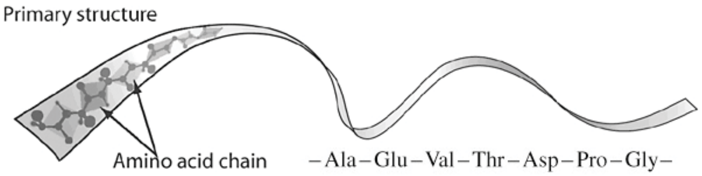
Feature Type
Point

Update Method
AtCurveCurveIntersection
AtDirectionPlaneIntersection
AtDirectionSurfaceIntersection
AtIntersection
AtIntersectionFromAllCombinationOfCurvesInSet
AtPlaneCurveIntersection
ByCartesianCoordinates
ByCoordinateArray
ByCoordinatesFromExternalFile
ByCylindricalCoordinates
ByDirectionAndDistanceFromOrigin
ByDistanceAlongCurve
ByFunction

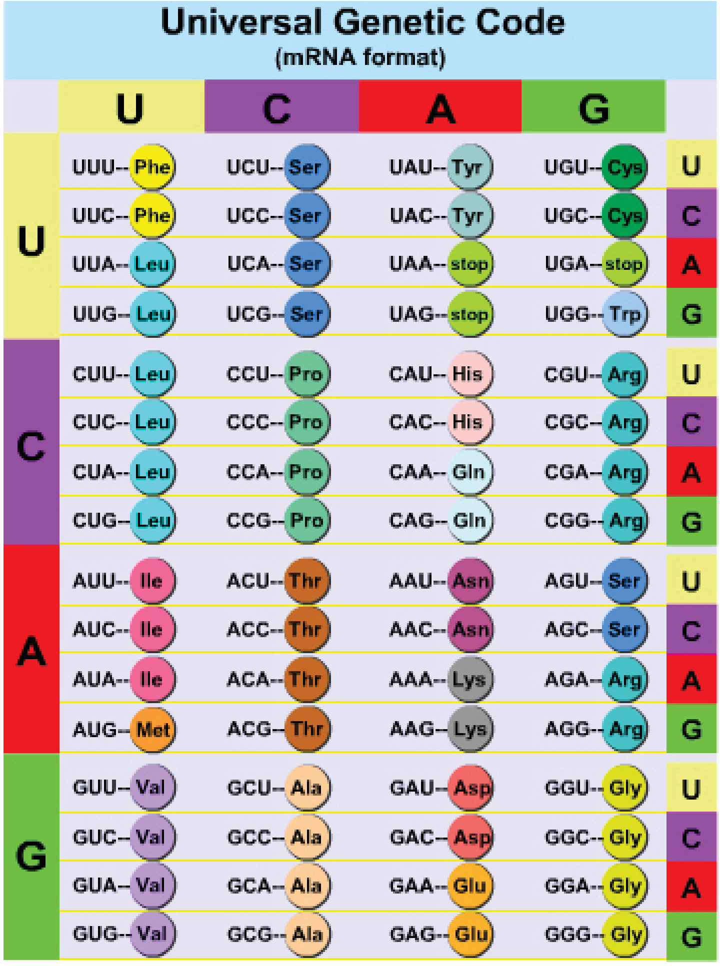
Type
Update method
Name of new feature

☐ Auto advance ☐ Assign active color, level and symbology

87. GenerativeComponents Point component and the subset of update methods by which the Point is recalculated.



88. Primary protein structure. The amino acid chain is a long sequence of amino acids.



Universal Genetic Code

(mRNA format)

	U	C	A	G	
U	UUU--Phe	UCU--Ser	UAU--Tyr	UGU--Cys	U
	UUC--Phe	UCC--Ser	UAC--Tyr	UGC--Cys	C
	UUA--Leu	UCA--Ser	UAA--stop	UGA--stop	A
	UUG--Leu	UCG--Ser	UAG--stop	UGG--Trp	G
C	CUU--Leu	CCU--Pro	CAU--His	CGU--Arg	U
	CUC--Leu	CCC--Pro	CAC--His	CGC--Arg	C
	CUA--Leu	CCA--Pro	CAA--Gln	CGA--Arg	A
	CUG--Leu	CCG--Pro	CAG--Gln	CGG--Arg	G
A	AUU--Ile	ACU--Thr	AAU--Asn	AGU--Ser	U
	AUC--Ile	ACC--Thr	AAC--Asn	AGC--Ser	C
	AUA--Ile	ACA--Thr	AAA--Lys	AGA--Arg	A
	AUG--Met	ACG--Thr	AAG--Lys	AGG--Arg	G
G	GUU--Val	GCU--Ala	GAU--Asp	GGU--Gly	U
	GUC--Val	GCC--Ala	GAC--Asp	GGC--Gly	C
	GUA--Val	GCA--Ala	GAA--Glu	GGA--Gly	A
	GUG--Val	GCG--Ala	GAG--Glu	GGG--Gly	G

89. Universal Genetic Code specifying relationship between the nucleotide bases and the amino acids derived from them. The information contained in the nucleotide sequence of the mRNA is read as three letter words (triplets), called codons. Each word stands for one amino acid.

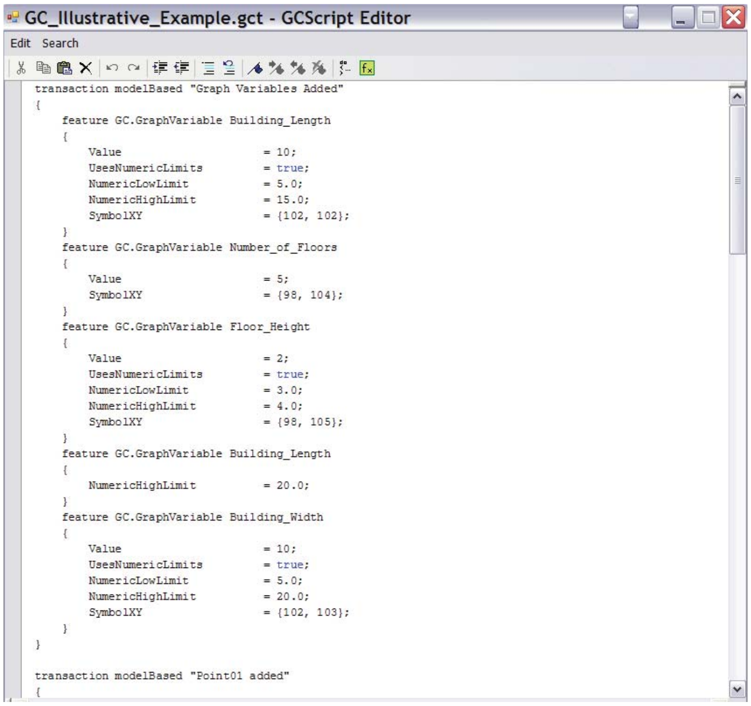
Secondary Structure – The orderly hydrogen bonded arrangements, alpha helix and pleated sheet, if present are called the secondary structure of the protein. The formation of the secondary structure is a function of the type of bonding that occurs within the molecule. See Figure 91.

Secondary Structure in Practice – In all manufacturing processes that are completed on a large scale where constructions derived from one piece of material are impossible it is necessary to rely on the accretion of building elements to complete the whole. Often times these members require a number of operations to be performed on them to allow for joining to other members as well as to derive their final form. CNC manufacturing relies on the output code from the design software to drive the relevant tooling and machines that create the physical elements. More than a graphical representation of the individual construction elements the secondary structure of the design holds within it the instructions necessary for their manufacturing. This information may appear in the form of code necessary for physical development of the element including laser cutting, milling, roll forming, thermoforming, brake forming or as information related to the placement of the member either by laser etching or bar code printouts for part scanning on site. The secondary structure then is a progression of the primary structure in that the developed code and instructions have been translated from GC language to a variety of different languages that can then help to define the tertiary form and placement of individual elements.

Tertiary Structure – The complete three dimensional conformation of the molecule. The secondary structure is a local structure that is formed of and may include the alpha helical, pleated sheet or random coil structure. The tertiary structure includes all the secondary structure and all the kinks and folds in between. See Figure 92.

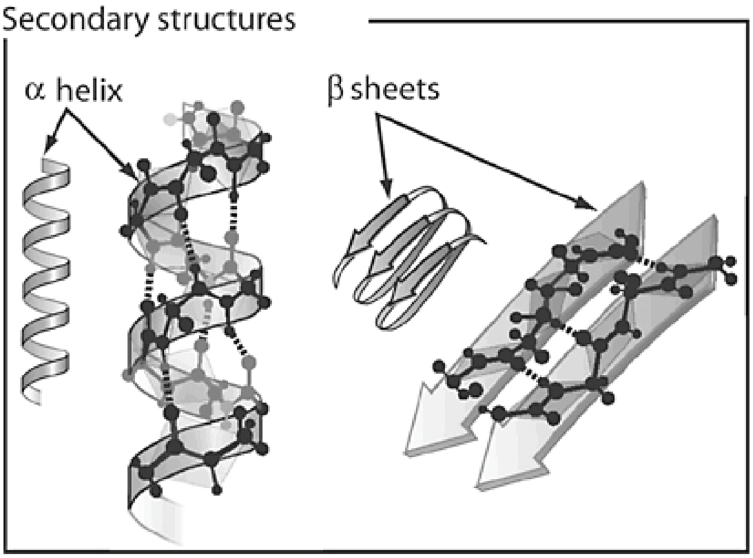
Tertiary Structure in Practice – The result of the transaction script and operations performed in the secondary structure produces the final component form. This physical manifestation of the modeling component represents a single building element that will be used for final construction. The component, in its tertiary form, may function as an independent building unit or it may be combined with other elements into a more complex assembly.

As there is often a need to produce physical models for verification purposes, GC allows a user to define features for the scaling of the model in the primary and secondary structures that enables the output of the tertiary components to vary from model to full production size. The ability of GC to suppress various transaction steps allows the designer to selectively add or remove detail to the model depending on the scale to which is it being produced. Ideally the elements produced in this phase will be designed according to their function either on their own or in concert with other elements.

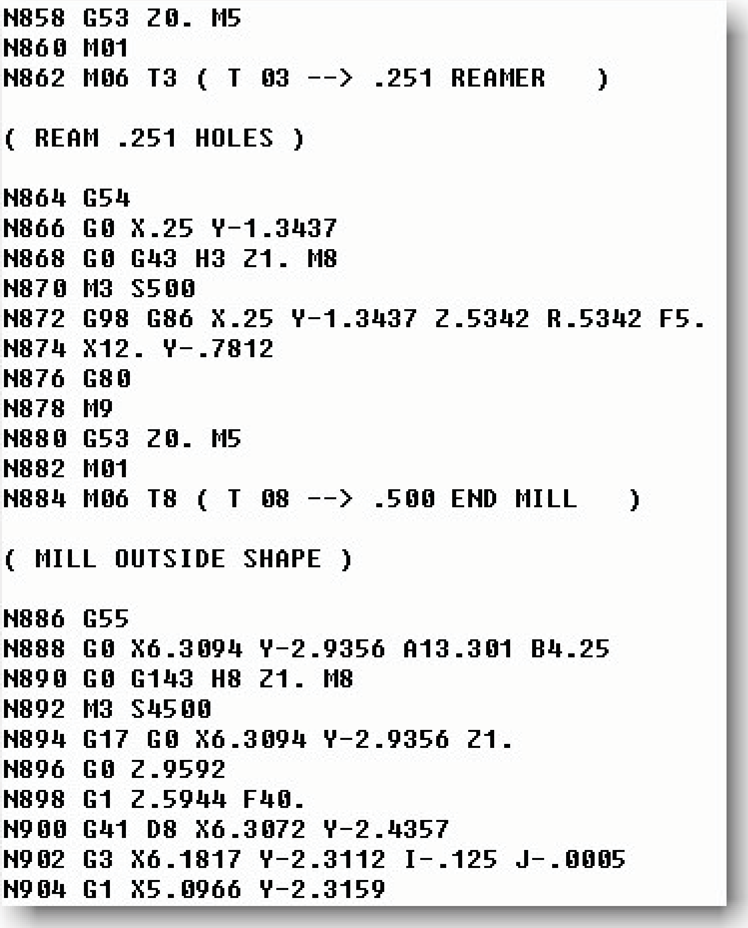


```
transaction modelBased "Graph Variables Added"
{
    feature GC.GraphVariable Building_Length
    {
        Value                       = 10;
        UsesNumericLimits           = true;
        NumericLowLimit             = 5.0;
        NumericHighLimit            = 15.0;
        SymbolXY                    = {102, 102};
    }
    feature GC.GraphVariable Number_of_Floors
    {
        Value                       = 5;
        SymbolXY                    = {98, 104};
    }
    feature GC.GraphVariable Floor_Height
    {
        Value                       = 2;
        UsesNumericLimits           = true;
        NumericLowLimit             = 3.0;
        NumericHighLimit            = 4.0;
        SymbolXY                    = {98, 105};
    }
    feature GC.GraphVariable Building_Length
    {
        NumericHighLimit            = 20.0;
    }
    feature GC.GraphVariable Building_Width
    {
        Value                       = 10;
        UsesNumericLimits           = true;
        NumericLowLimit             = 5.0;
        NumericHighLimit            = 20.0;
        SymbolXY                    = {102, 103};
    }
}

transaction modelBased "Point01 added"
{
```

90. GenerativeComponents transaction file.



91. Secondary structure of protein molecule.



```
N858 G53 Z0. M5
N860 M01
N862 M06 T3 ( T 03 --> .251 REAMER   )

( REAM .251 HOLES )

N864 G54
N866 G0 X.25 Y-1.3437
N868 G0 G43 H3 Z1. M8
N870 M3 S500
N872 G98 G86 X.25 Y-1.3437 Z.5342 R.5342 F5.
N874 X12. Y-.7812
N876 G80
N878 M9
N880 G53 Z0. M5
N882 M01
N884 M06 T8 ( T 08 --> .500 END MILL   )

( MILL OUTSIDE SHAPE )

N886 G55
N888 G0 X6.3094 Y-2.9356 A13.301 B4.25
N890 G0 G143 H8 Z1. M8
N892 M3 S4500
N894 G17 G0 X6.3094 Y-2.9356 Z1.
N896 G0 Z.9592
N898 G1 Z.5944 F40.
N900 G41 D8 X6.3072 Y-2.4357
N902 G3 X6.1817 Y-2.3112 I-.125 J-.0005
N904 G1 X5.0966 Y-2.3159
```

92. G-Code for milling machine operation. The coding specifies a number of different operations or requirements that the machine is required to perform. For example:

G53 = motion in machine coordiante system
M01 = optional program stop
M06 = tool change
G54 = use preset work coordinate system 1
M3 = turn spindle clockwise

Quaternary Structure – Refers to the association of two or more peptide chains in the complete proteins. Essentially it is the building of the active protein molecule through the interaction of the unique tertiary forms of the peptide chains. See Figure 93. Not all proteins exhibit quaternary structure however, and they may in fact be fully functional in their tertiary conformation.

Quaternary Structure in Practice – The quaternary structure represents the final assemblage of the unique tertiary components. It can be viewed as the functional equivalent of an accretion of building elements where a larger component is derived from multiple smaller or less complex elements. The depth of functional interaction here can occur on degrees of involvement with each other. An individual element such as a structural member can combine with other members to produce an elaborate wall structure. Each tertiary element combines to form a structural unit that functions on a large scale. Alternatively, the quaternary structure could also represent an arrayed surface population of adaptive cladding panels for that same wall. The addition of all the tertiary and quaternary elements will form the following proteome. See Figures 94-96.

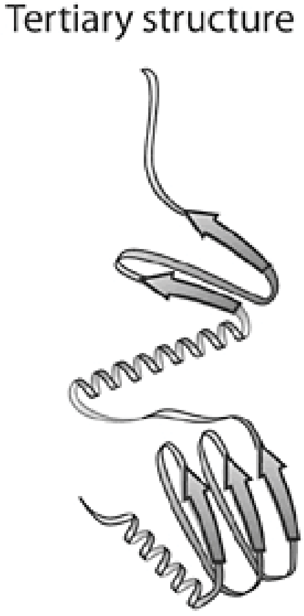
Proteome

The final form of the building and its components as realized in its built configuration represents a static version of the proteome as captured after all of the relevant design forces have affected it. The digital version of the proteome is however able to change and could have the capacity to drive the evolution of another project with similar formalistic requirements but varying morphological constraints. In essence, a new environmental condition will be able to interact with the program and define a new building with existing instructions.

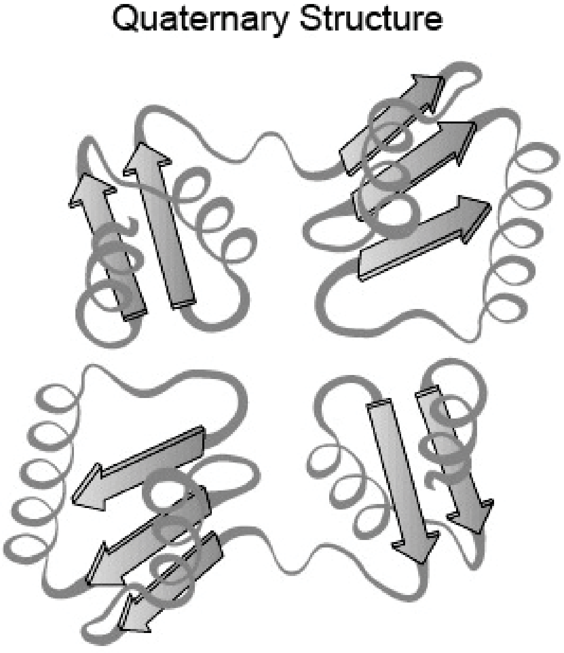
6.1.6 Interoperability and BIM (Building Information Modeling)

In creating a design system that effectively functions on and within a number of levels to provide ease of use in all design disciplines, the issue of interoperability arises. Interoperability is a term that refers to the "ability to manage and communicate electronic and project data between collaborating firms' and within individual companies' design, construction, maintenance, and business process systems… Interoperability relates to both the exchange and management of electronic information, where individuals and systems would be able to identify and access information seamlessly, as well as comprehend and integrate information across multiple systems." (Gallaher 2004, p.ES-1)

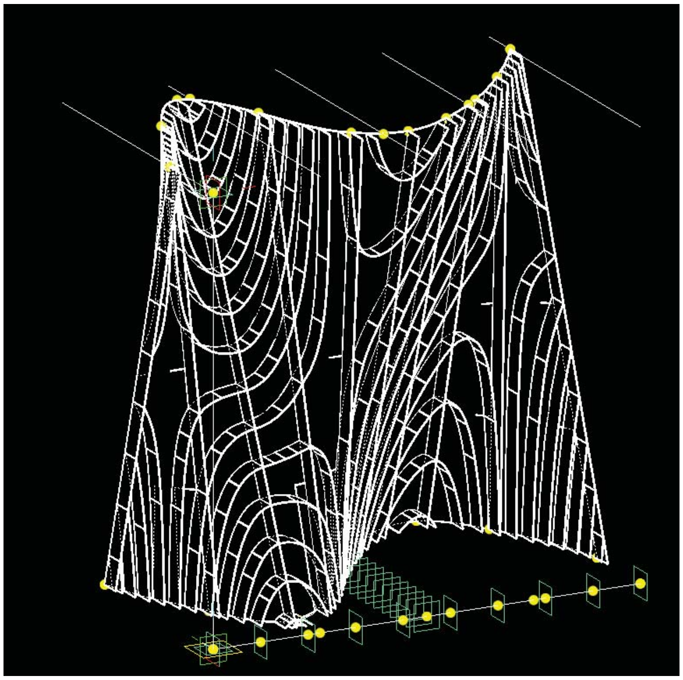
A number of manufacturing sectors including computer, automobile and aircraft have already made advances in the integration of design and manufacturing, maximizing automation technology, and replacing many paper documents with electronic equivalents. The AEC industry however, has yet to realize the



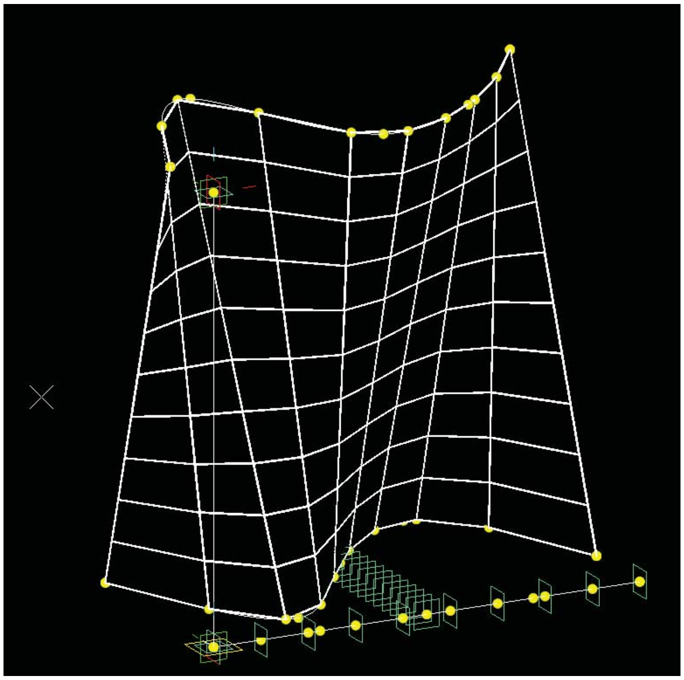
93. Tertiary structure of protein molecule.



94. Quaternary structure of protein molecule.



95. Structural elements.



96. Adaptive panel cladding system.

potential savings available with a widespread application of these approaches.

The values quantified for the U.S. capital facilities supply chain in 2002 indicate that the costs of inadequate interoperability through the life-cycle of a building for the AEC community including specialty fabricators and suppliers totaled US$5.176 Billion. This represents between one and two percent of industry revenue but these values have been recognized as representing only a portion of measurable interoperability cost losses. (Gallaher 2004, p.ES-7) It is possible to see then how a reformation in the process and product of design and construction could lead to potential savings with regard to both time and money.

BIM as it is known is a term that describes a number of modeling environments that allow for the partial parametric generation of a 3D building model with associated logical output of 2D drawings, component lists, building costs, structural analysis, etc. On top of this is the ability for information exchange between participants in all aspects of the building from design to manufacture to construction. While other industries using integrated digital environments such as CATIA, SolidWorks, etc. have attempted to utilize a holistic design approach to design and manufacture, the architecture industry has lagged behind. With the evolution of Gehry Technologies Digital Project, Graphisoft ArchiCAD, Allplan, and Autodesk Revit the architectural field is now home to a much more sophisticated set of design software. There is still much more room for development, however. (Schodek 2005, p184)

6.1.7 Additional Areas for Further Research

There are a number of additional areas that are well suited to and contribute to the progressive development of digital design for the AEC community. These approaches also strive to develop a design through a minimal amount of instructions and design parameters. The following section briefly outlines the premise of each but they are intended for illustrative purposes and as such lie outside the scope of this thesis.

6.1.7.1 Genetic Algorithms

In a Genetic Algorithm (GA), a chromosome (also sometimes called a genome) is a set of parameters which define a proposed solution to the problem that the GA is trying to solve. The chromosome is often represented as a simple data string although a wide variety of other data structures are also in use as chromosomes.

A GA creates many chromosomes, either randomly or by design, as an initial population. These chromosomes are each evaluated by the fitness function, which ranks them according to how good their solution is. The chromosomes which produced the best solutions, relatively speaking within the population, are allowed to breed, also called crossover. The best chromosomes' data is mixed, hopefully producing more refined subsequent generations. The functional design of the GA can vary dramatically from one to the next and it is the programmer that defines the amount of user input that will allow progression to occur. While a GA may carry out all of its computation automatically, an Interactive Genetic Algorithm may be used that requires human intervention at a number of key steps that have been defined for it.

The GA is essentially a structured method of selecting between alternative design possibilities. In principle, this method of selection could be integrated into the GC design environment to aid in the selection or derivation of designs that must fulfill a number of quantifiable criteria.

6.1.7.2 Rule Based Programming

The fundamental approach to rule based programming is the implementation of replacement rules for processing rather than procedural constructs. In this approach a number or collection of rules is developed that defines the actions that are to be taken by the program with regard to specific situations. In an architectural sense the design requirement may be the effective storage of the design experience from various projects, not at the level of the design itself, but at the level of the principle of assembly behind the designs. Rather than actually documenting the design itself the program is infused with the rules for the design and it creates the required details depending on the particular stylistic or construction principles that are written into the program. (Seebohm 1998) Here, the program is acting in a manner that allows for multiple outcomes depending on the current environment in which it is functioning. The possibilities for a functional and automatic feedback loop exist but there is added complexity in tracing the logic string and ensuring quality assurance.

6.1.7.3 Nanotechnology

Nanotechnology represents the physical realization of AEC industry on a truly cellular level. By reducing architectural constructions to a scale measured in nanometers the possibilities for organic or quasi-organic forms become possible. A building could theoretically be programmed to grow itself based on the instructions of the architect. Like current 3D printing technology the building could raise itself as one cohesive unit rather than an amalgamation of disjunctive assemblies. Buildings could repair themselves, transmit information about their current status with regard to temperature, stress, fatigue, air quality and any number of other desirables. They could change shape, porosity with regard to ventilation or ingress/egress. The possibilities at this level of architectural construction are almost limitless but the fruition of development in this area will only come with an incredible design mechanism that is able to control it.

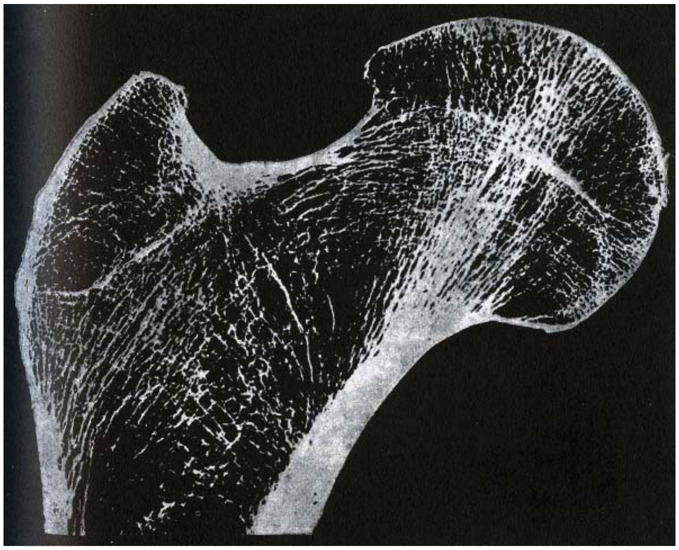
6.2 Design Concept #2 - Ruled Surface Structure

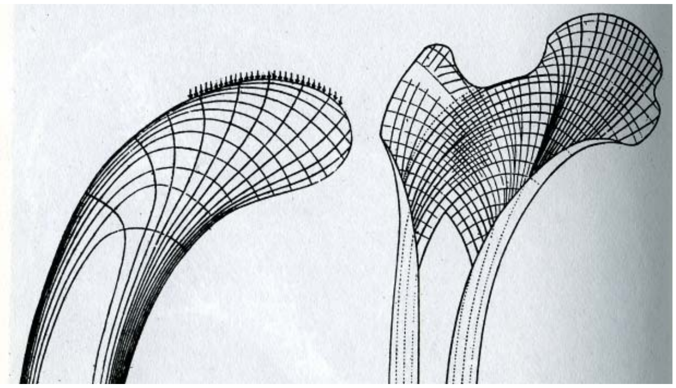
The complexity involved in creating non-orthogonal structures is often associated with higher design, production and labor costs. This has been a negative influence on the proliferation of these types of structures particularly in North America, where the economic vision focuses on the short term. This design investigation seeks to develop a concept for the design and construction of these types that satisfies the criteria outlined in *Section 5.3*.

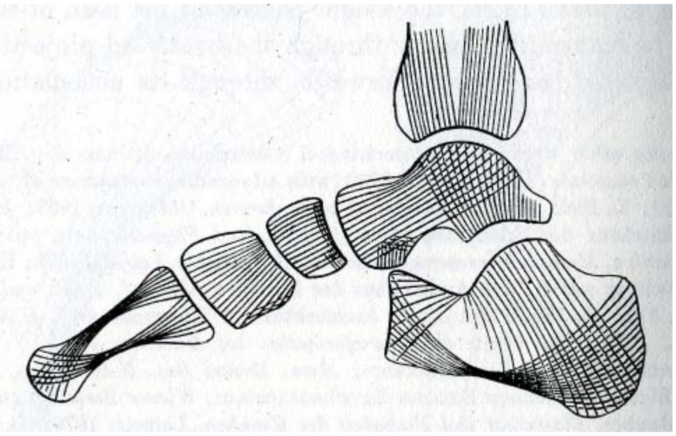
6.2.1 Inspiration

The development of an organism from youth to maturity occurs with a number of environmental and internal stresses acting on it which help to determine its final form. As illustrated previously however, their response to these stresses may act in a static or dynamic way.

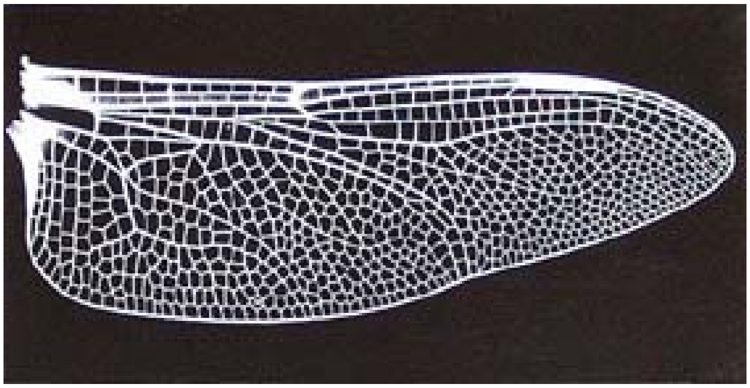
Bone morphology changes throughout time and is in a constant state of reformation to balance the forces acting on it. This closed loop system of reformation is able to sense a variety of environmental variables and change itself accordingly. In addition to the dynamic nature of bone, it also possesses a unique cross-linked internal structural pattern that provides incredible strength with a minimal investment of material and weight. The structure of the tibia bone in the human leg is capped by a widened tip that covers the hollow cylindrical shaft that it rests on. The interesting structural implication here is how the vertical pressures acting upon the head of the bone are trans-



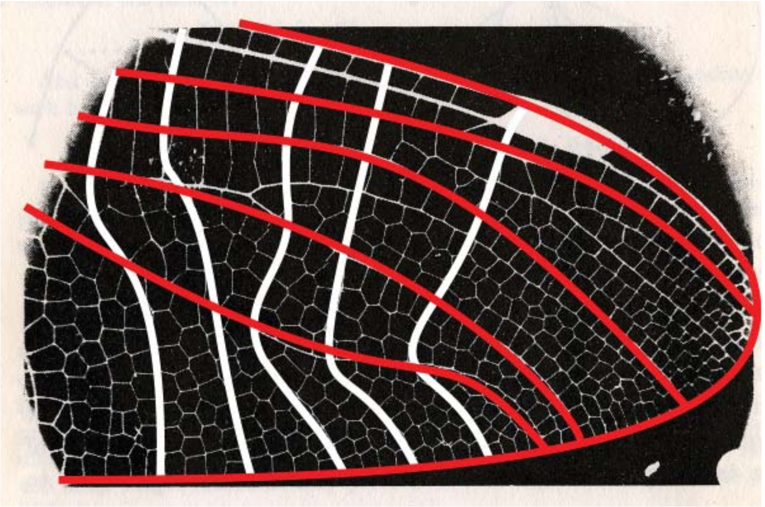




97. Head of the human femur in section
98. Crane-head and femur
99. Diagram of stress-lines in the human foot.



100. Dragonfly wing.



101. Primary and secondary veins of dragonfly wing.

ferred to the walls of the hollow shaft below. Within the hollow space there exists a variety of living tissue including marrow, blood vessels, and others; among which is an intricate latticework of "trabeculae" of bone which form the "cancellous tissue." See Figure 97.

"The trabeculae, as seen in the longitudinal section of the femur, spread in beautiful curving lines from the head to the hollow shaft of the bone…nothing more nor less than a diagram of the lines of stress, or directions of tension and compression, in the loaded structure: In short, that Nature was strengthening the bone in precisely the manner and direction in which strength was required…" (Thompson 1963, p976) See Figures 98-99.

The dragonfly wing appears to exhibit a complicated and seemingly random structural system consisting of a network of various sized veins. See Figure 100. To duplicate and enlarge this structure in order to fulfill an architectural role would be impractical and extremely labor intensive. However, if the wing is examined in finer detail it is possible to identify the overall structural trends that determine its primary configuration and thus design a simpler architectural structure with similar properties. The wing is traversed longitudinally by a series of strong veins that run more or less parallel to each other. Finer veins run between the main veins in a meshwork of "cells." See Figure 101. The walls of the cells within the meshwork while subdivided into a matrix exhibit tendencies to follow lines of running at angles to the main structural veins. (McLendon 2005, p2)

6.2.2 Design Outline

In order to begin development of a design approach for non-orthogonal structural building skins that allow for fluidity, changeability and overall ease of design, manufacture and construction it is first necessary to arrive at a proper form for exploration. A number of surfaces have been investigated in this thesis from flat to compound curves. Of particular interest is the ruled developable surface in that curvilinear forms can be derived from flat panel materials. While this characteristic is important with regard to ease of manufacture and construction of the surface condition it also allows for a novel approach to the development of the structural members that form it. With a conscientious approach to the design of the ruled surface it is hypothesized that the primary, secondary and tertiary members can all be fabricated out of identical width linear lengths of material that must merely be bent in one direction if at all depending on their function and location. This will have to be done however by putting aside some current assumptions of design and construction which will be illustrated when required.

The shape of a building element has the capacity to be different or identical to any number of other elements within the building. In a relatively simple rectilinear building many of its elements could theoretically be interchanged as with one wall stud for another. Without proper and extensive documentation however it becomes difficult to properly locate elements that may have similar configurations but different physical properties for strength, etc. This situation may exist in a multi-floor construction where the members on the lower floors are stronger yet have identical morphology to members directly above them. While this may result in an increase in the requirement for construction documentation it does make manufacturing easier as there is a large degree of replication and standardization. Linear components also reduce the requirement for intensive CNC manufacturing that although quite efficient and accurate can become quite labor intensive if each element requires a different setup for clamping, forming, etc.

In a curvilinear construction there is often a requirement for many unique pieces that need to be placed in many different locations and transferability cannot occur. Although it may at first seem daunting to construct a building enclosure with many unique pieces the simple fact that they are unique limits their organization to only one possibility. With an efficient numbering or labeling system it is possible to construct it with a small number of instructions for assembly rather than a comprehensive collection of construction drawings for building element location and orientation. In effect, the instructions for the physical form of the building itself are contained within the three dimensional conformation of the individual building elements. The presence of many unique non-orthogonal structural members however often requires multiple elaborate template layouts for laser or plasma cutting usually carrying with them a certain degree of material waste.

As noted in section *2.1.4 Hierarchy of Structure*, the trend for orthogonal constructions in the dissipation of internal and external forces is to transmit them downward in an additive vertical fashion. The presence of localized stresses in the form of impact or environmental anomalies can cause catastrophic failure to occur.

Structural patterning is quite prevalent in construction today where multiple unitized elements are distributed throughout the building in an effort to reduce design and construction time. The case quite often though is that there is an associated hierarchy of structural forces where smaller elements dissipate their forces into successively larger elements in a vertical fashion until they are transmitted to the ground. A failure in one of the base elements can prove catastrophic for the building as the force distribution is additive in each subsequent element. Natural principles favor an alternative approach to the distribution of forces where they are dissipated among many different pathways thus avoiding localized stresses on the organism. In this thesis, the natural approach to structural design principles as they relate to exoskeletons will be used. Structure and skin will be integrated into one unit rather than existing as separate entities. The final form of the structural elements will be partially dependent on the final form of the skin which will allow the two to develop concurrently.

The scenario developed here will attempt to produce a design that allows a certain degree of building element modularity for ease of manufacture while maintaining morphological individuality for uncomplicated construction. At the same time the digital portion of the design will facilitate a generative closed feedback loop where additions to the whole or changes to certain predefined areas will provide automatic update of all the required fabrication and construction requirements with a minimal number of instructions. The in-built customizability of the design will also allow the design to be useful in a variety of building scenarios rather than be unique to only one site. The physical form of the design will be derived so that stresses are distributed throughout the structure in a number of different directions thus minimizing the presence of localized stresses and the possibility of structural failure. In the end it is hoped that through an efficient and logical process of design, manufacture and construction that it will be possible to produce a final form that is aesthetically pleasing, applicable and relevant in a variety of building applications, efficient for affordable construction and structurally sound.

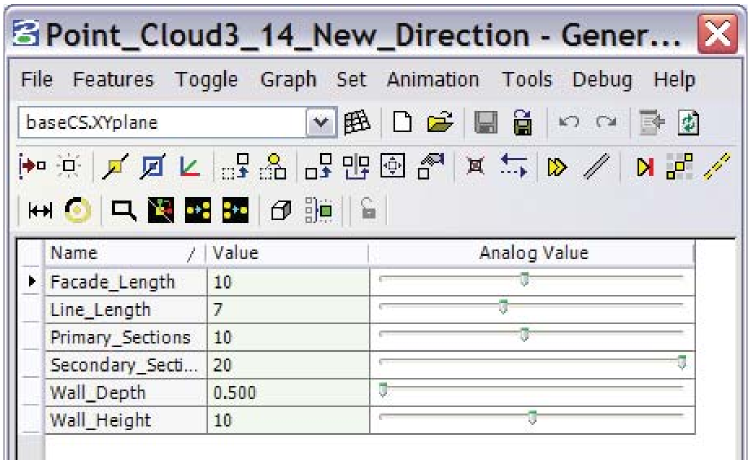
In keeping with the design model outlined in *6.1 Design Documentation* there are a number of approaches that can be taken to arrive at a desired final product. The direction outlined below represents one pathway of the design. After the resultant model has been created there will be a number of questions asked about its feasibility both positive and negative and how the design can be improved from there. While it is intended that a complete building project from start to finish would attempt to utilize the entire design philosophy set out in *Section 6.1* the focus of this design concept will be contained within an approach to developing a base parametric model that is suitable for export and use in a variety of different analysis and manufacturing programs. GenerativeComponents will be used as the digital software for generation of the design and as a platform for drawing and manufacturing export.

6.2.3 Design Product

The starting point is to develop a design condition that can be applied to a variety of sites and applications. Once that scenario is in place it is possible to begin developing a model that is able to adapt to those conditions. To reduce initial complexity of the design requirements the façade was restricted to only one face of a potential building. This type of condition could exist in an infill condition or within a restrictive urban site.

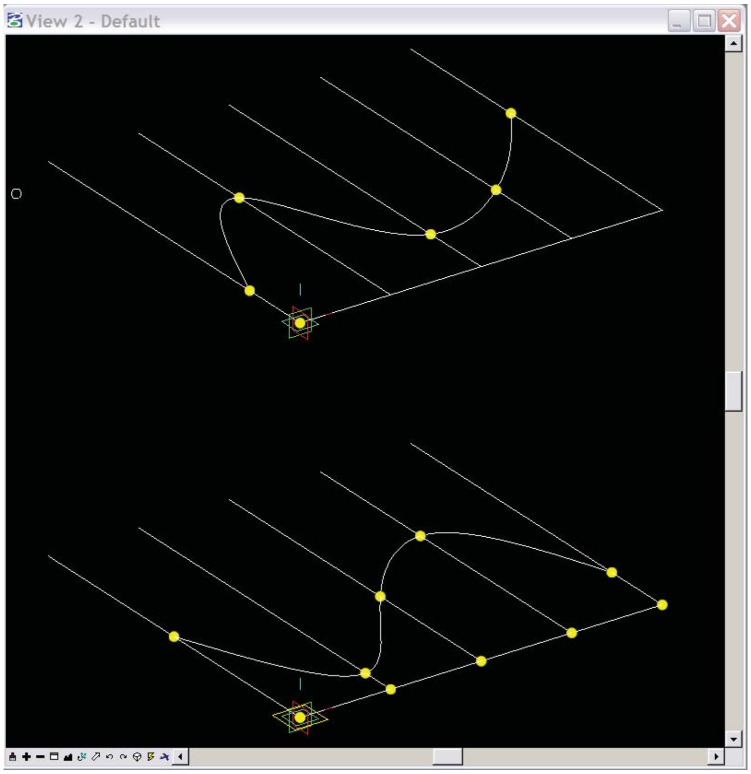
1. **Identification of key parameters that will contribute to the functionality of the model and allow for the desired level of variability in the design.**

When the initial design of a building is taking place there are often a large number of variables that are unfixed and changeable. By carefully planning the strategy for the development of the building concept then it becomes possible for these variables to become exactly that. Changes and deformations to the overall design can be quickly visited and revisited. In this case the following variables will be allowed for.

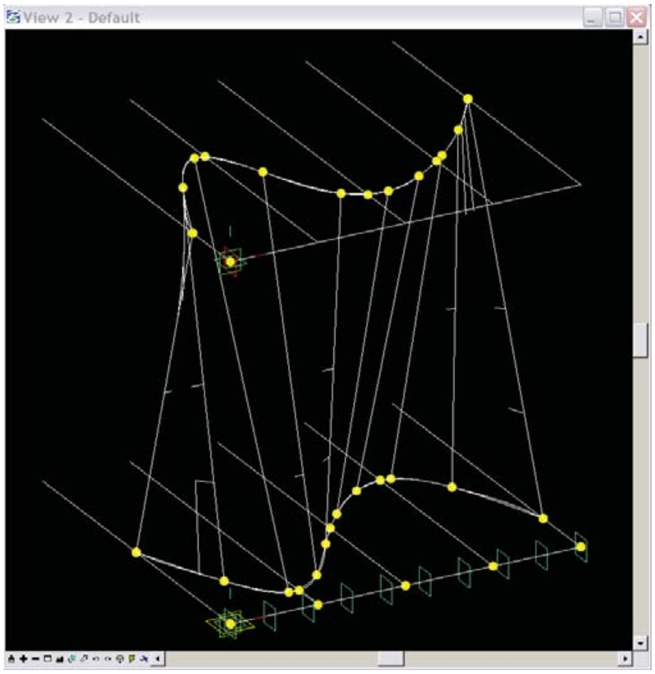


Name	Value	Analog Value
Facade_Length	10	
Line_Length	7	
Primary_Sections	10	
Secondary_Secti...	20	
Wall_Depth	0.500	
Wall_Height	10	

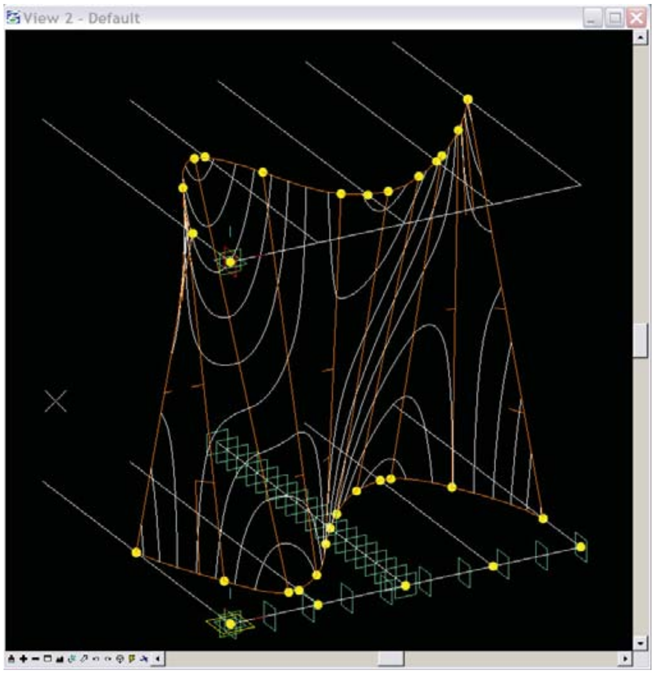
102. Graph Variables



103. Layout parameters and defining curves.



104. YZ Planes and the resulting BsplineSurface and primary structural member layout lines.



105. XZ Planes and the resulting secondary/tertiary layout lines derived from the BsplineSurface.

- Wall length
- Wall height
- Wall thickness
- Number of sections for deriving the primary structural elements
- Number of sections for deriving the secondary/tertiary elements

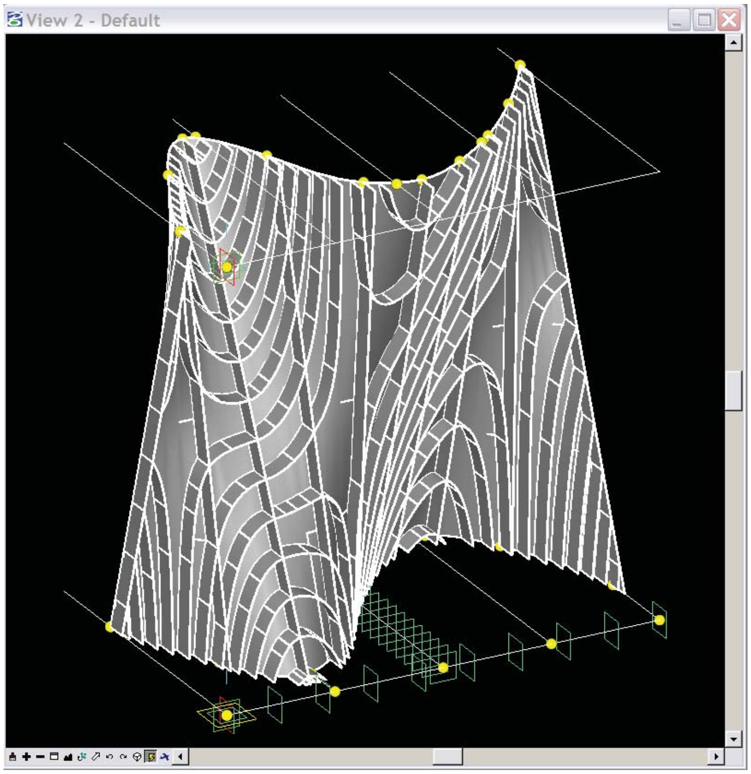
Now that the variables for design have been identified it is possible to begin working in GC to create graph variables that define these parameters and allow for their manipulation. It should be noted however that the expression deriving the variable output may in fact rely on the output from another component which must be created before the GV in order to be recognized due to the dependency hierarchy. In this case all of the expressions for the GVs will be independent and stand alone in their variability. See Figure 102.

2. **Development of the design model with a logical progression of generative features.**

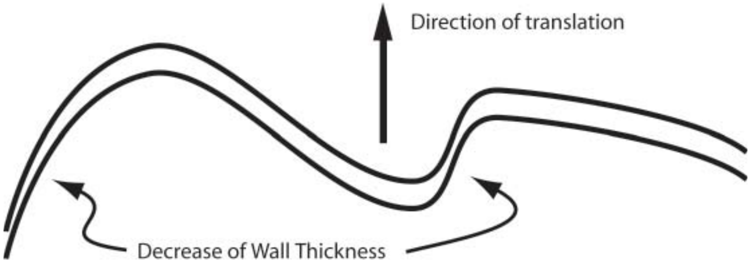
The first step here is to create a virtual envelope of layout parameters that allow for the three dimensional definition of the final form. The value of the layout lines that describe these parameters are based on the GVs created previously. Layout points are created along a series of equally spaced bays which define the upper and lower curves that will define the ruled surface. The position of each layout point is individually variable which allows the designer to change the definition curves and the subsequent surface derived from them. See Figure 103.

The location of the primary structural elements required for the facade are developed by intersecting a variable number of evenly spaced YZ planes based on subdivision of the Façade_Length with the curves defining the ruled surface. The points produced from the intersection of those curves will then be used to define both the structural members and the RuledBsplineSurface facade. This approach guarantees that the structural members will lie directly in plane with the ruled surface itself. When dealing with bezier curves and surfaces derived from them there can be discrepancy in correlation between the surface and curves if there are a different number of nodes present as is the case here. The layout lines that must be physically replicated on site for foundation work, etc. can be derived from the BsplineSurface thus maintaining the best possible construction tolerance. See Figure 104.

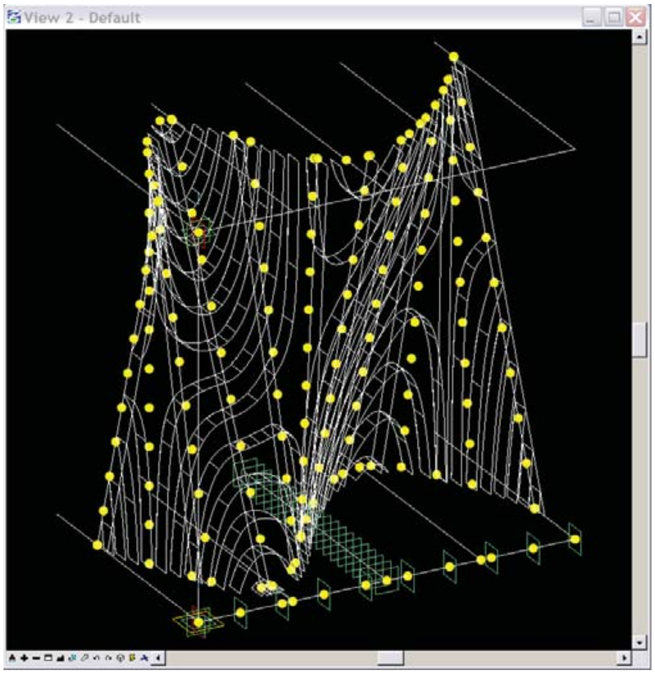
A variable set of XZ planes is created that run perpendicular to the YZ planes. The intersection of these with the BsplineSurface will produce curves defining the conformation of the secondary/tertiary structural members. In defining the members this way it is intended that their natural conformation will follow lines of stress within the structure where member density will increase based on the curvature of the facade. It should be noted that in this model, the derivation of the members occurs without any external loading conditions which would need to be addressed in subsequent iterations. The fact that secondary/tertiary members meet the primary structural members at varying angles develops a triangulated



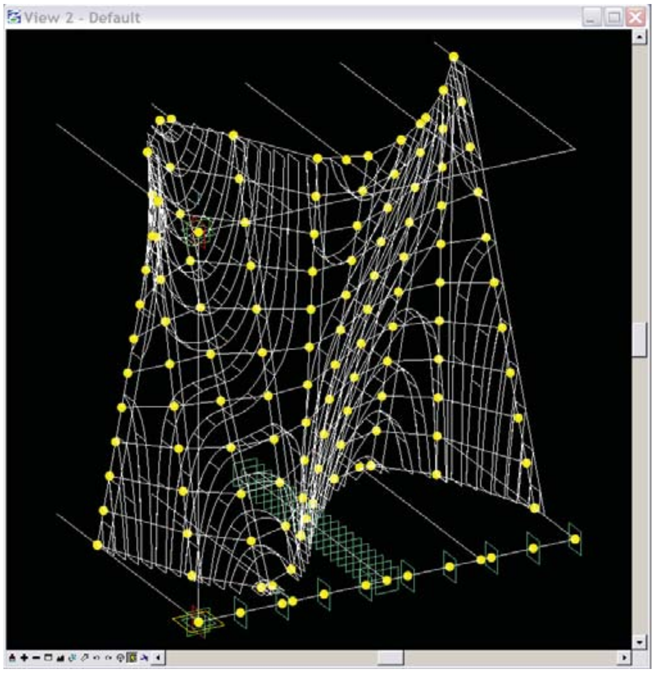
106. Extrusion of the primary and secondary/tertiary members in the Y direction.



107. Direction of translation and associated decrease in wall thickness.



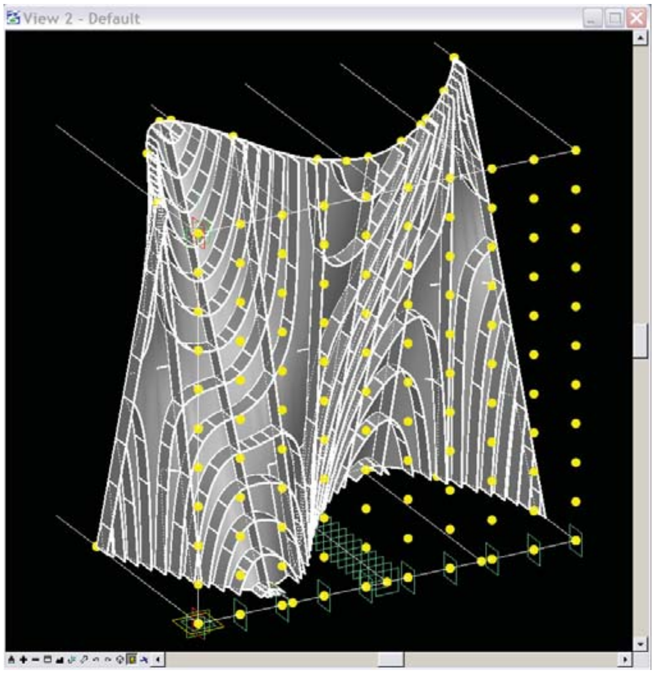
108. UV Points on BsplineSurface



109. Surface panels on BsplineSurface

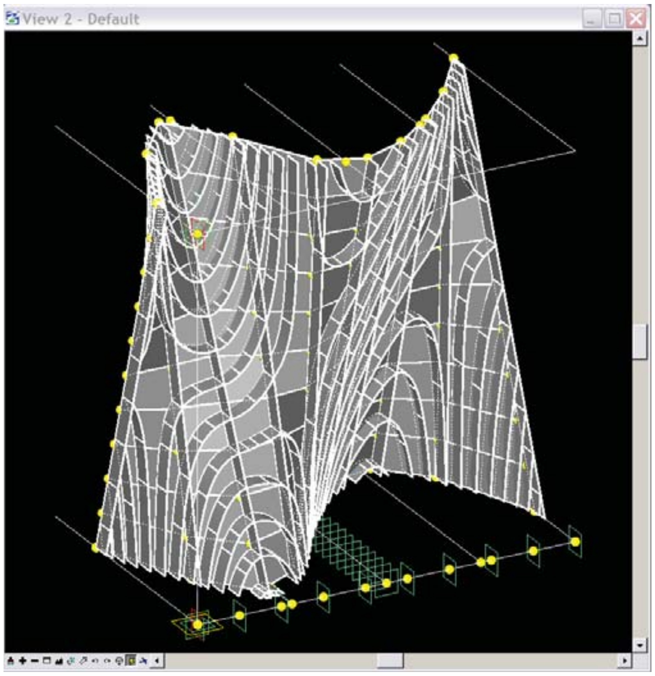
structural framework that resists not only vertical compression but horizontal shear in both the X and Y directions. The other appreciable benefit to secondary/tertiary members being derived this way comes from the fact that they are curvilinear in the direction perpendicular to, and linear in line with, their length. This means that their fabrication can sidestep the CNC driven cutting that would be required if they were curved in the direction of their length. It should be noted that this arrangement can only be realized with the use of a developable ruled surface. See Figure 105.

After construction of the curves defining the primary and secondary/tertiary members they can be extruded in the Y direction the desired depth of the wall. A variable length line whose expression is defined by the Wall_Depth GC is used to create extruded BsplineSurfaces along the structural layout curves. This method of extrusion creates structural members that are all of an identical depth. See Figure 106. Once again this aids in ease of production by the allowing the manufacturer to create the members out of linear strips of plate steel that can be easily sheared or cut to width with minimal adjustment of machinery. While this does allow for ease of production there are some considerations that must be recognized in order to prevent design oversights from occurring. The straightforward extrusion or translation of a surface, as is the case here, into a solid produces one in which the wall thickness will vary depending on the curvature of the surface and its alignment to the direction of translation. See Figure 107. As surface curvatures increase and the wall direction comes closer to the direction of translation, the wall thicknesses will diminish until a point is reached were the two surfaces would intersect. If the planar constraints allow for an extruded surface without intersection then the inner and outer surfaces will be identical in shape. This means that any panel configurations derived from the surfaces will also be identical inside and out effectively halving the number of unique panel configurations that would be necessary with a surface that has been offset. If intersections or unacceptable wall depths occur then either an adjustment of the layout curves defining the surface or a different design approach would be required at that location.



110. Point grid created based on location of the primary elements

3. **Creating output conditions for visualization, construction drawings, fabrication etc.**



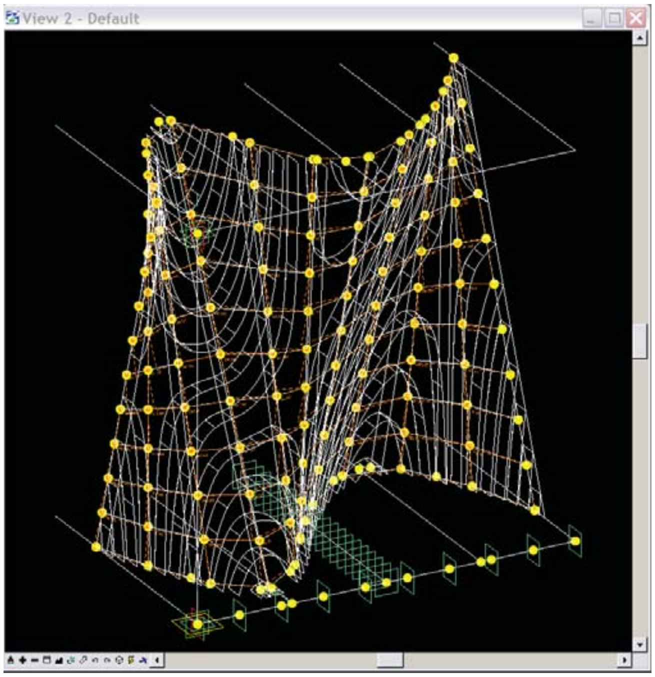
111. Surface panels created from projection of point grid onto the BsplineSurface

Now that the design model has been created it is necessary to begin the process of translating the developmental and visual information it contains into a format for manufacturing and construction. While the BsplineSurface defining the skin condition could potentially be constructed from one large piece of material, this obviously becomes difficult when the structure increases in size. With this being the case it becomes necessary then to subdivide the surface into a number of smaller surface panels for manufacturing and construction.

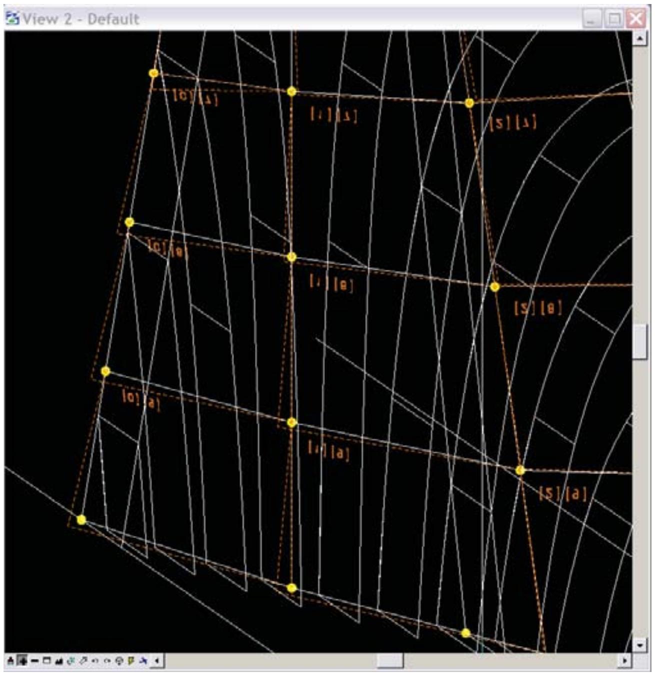
There are a multitude of ways to create the surface panels with each approach having distinct benefits and drawbacks. Here, two of those approaches will be developed. The first method involves populating the BsplineSurface with a series of variable UV points which are points described on a 3D surface by 2D transformations along it. These points serve to define the corners of the surface panels which can then be derived by creating shapes between them. See Figure 108-109.

The shapes created between the UV points are planar and as a result do not conform exactly to the ruled surface. This condition can result in improper sizing of the manufactured panels. As the density of UV points on the surface is increased so too does the correlation of their form to the native form of the BsplineSurface thus reducing error. The position of the UV points does not correspond with the location of the primary structural members so a separate panel attachment system would need to be developed which would increase production and construction cost. In this particular design scenario this method of surface subdivision is the least efficient.

The second method involves creating a virtual point grid corresponding to the location of the primary structural members in the X direction and an arbitrary value set by the designer in the Z direction. The panels produced here are similar to the UV derived panels in that they are composed of planar surfaces and hence are not as accurate as possible. Their fastening to the structure becomes much easier in that their vertical edges line up with the primary structural members. See Figure 110-111.



112. ConstructionDisplay is added with text for location of the panels on the facade.



113. Detail of ConstructionDisplay and text style applied to the panels for export to FabricationPlanning.

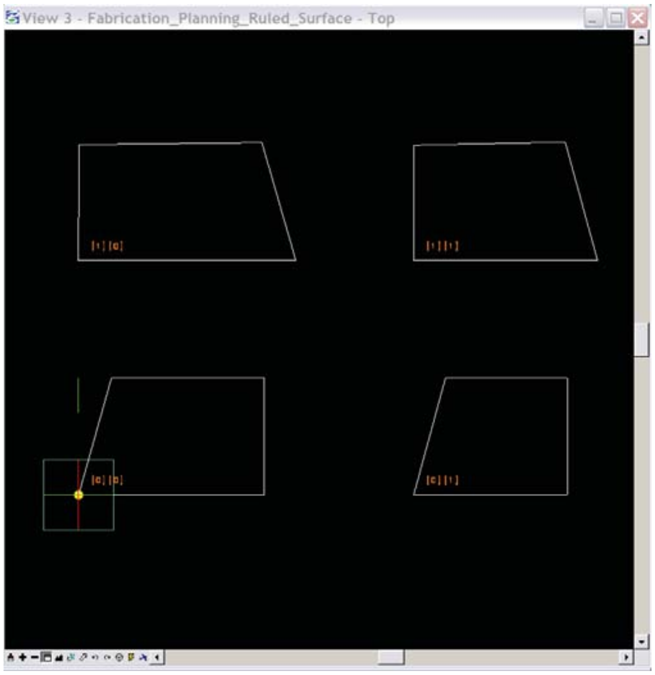
The third method would build on the second in that a point grid would again be used to define panel corner points on the surface. This time however, and with further research, the panels would be derived by intersecting the lines connecting the surface points with the BsplineSurface and flattening them. This would create panels that are developed from the BsplineSurface itself thus being much more accurate than the planar approximations from the first and second methods.

The fourth method of panel development would involve extracting and developing the entire BsplineSurface into a separate fabrication file where it could be subdivided with a regular grid. The interior panels in this instance could all me made exactly the same size which would greatly reduce manufacturing time. However, the same situation for fastening would arise as with the first method.

The benefit of using GC to develop these methods is that each one can exist within the same transaction file and they can be selectively turned on or off when required. This allows for the designer to revisit, change or develop any one or combination depending on any number of construction variables or requirements such as cost, delivery schedules, manufacturing capabilities, etc.

After the panels have been created in the 3D model it is possible to export them to another file for fabrication. A new Model is created that is used to import the flattened panels from the 3D model. A TextStyle is created

114. Flattened panels ready for laser cutting in the FabricationPlanning file.



115. Detail of text style applied to panels for ease of identification and optional scribing by laser. cutter

that will be used to label the individual panels for laser etching and their location in the 3D model. The FabricationPlanning feature is used to export the 3D panels into the 2D Model and the TextStyle is applied. The visibility of the TextStyle is controlled by creating a feature called ConstructionDisplay that can toggle it on or off. The 2D FabricationPlanning file can then be directly exported to a laser cutter for fabrication.

The development of all of the structural members in the model and for fabrication and construction would proceed in a similar manner. As of the writing of this thesis the GC program is still in its pre-beta phase and as such does not contain all of the functionality that is expected with the first release. The ability to develop and export G-code required to drive CNC rollers and manufacturing machines is expected to be contained with the fully developed version.

6.2.4 Design Evaluation

The design concept developed here represents an approach to design that uses biomimetic principles of stress based growth, self assembly, sensing and responding, scale increases, and the power of shape.

The benefits derived from using these principles in the GC parametric design environment are appreciable with regard to both the design itself as well as the associated manufacturing and production requirements.

Advantages

Translating the BsplineSurface instead of offsetting it.

- Allows the inner and outer panels to be of identical shape.
- Allows the structural members to be composed of identical width material.

Vertically sectioning the BsplineSurface to derive the secondary/tertiary members.

- Members can be made from linear strips of roll formed flat sheet.
- Laser/plasma cutting is required only at structural intersections and not at structural member edges. Exterior edges can be sheared which drastically reduces manufacturing time.
- The 3D conformation of the members ensures that they can only be placed in their correct location.

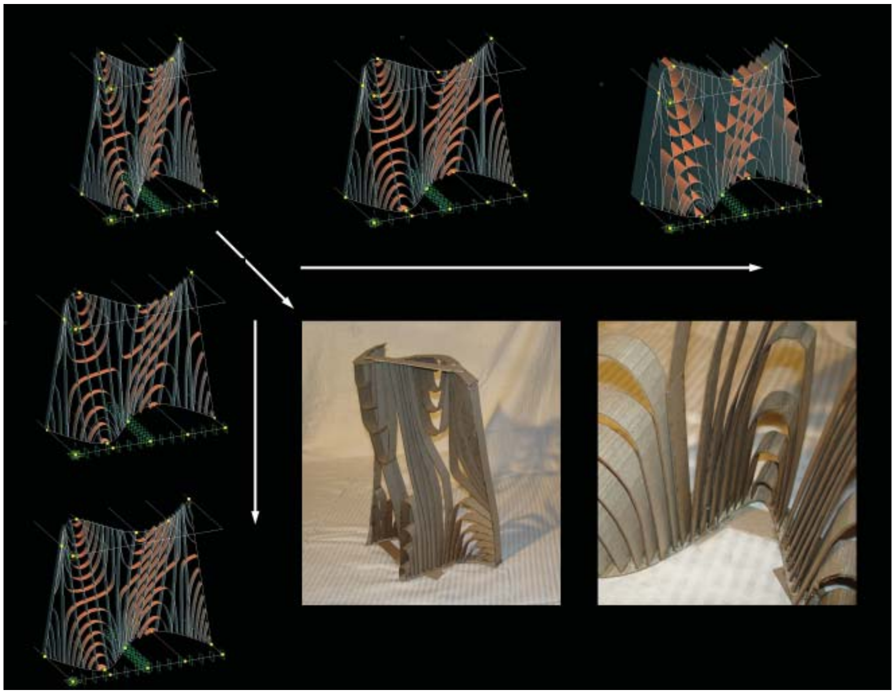
Development of the model in the GC parametric environment.

- The relative freedom of hierarchical organization created in the transaction file allows a completed and sometimes awkwardly built model to be easily updated and the feature elements to be laid out in a cleaner more concise manner. Any new person coming into the project will know and be able to follow in a linear manner exactly how the model was built, what its outputs are and the method in which it can be manipulated in an existing or potential context. See Figures 116 & 177.

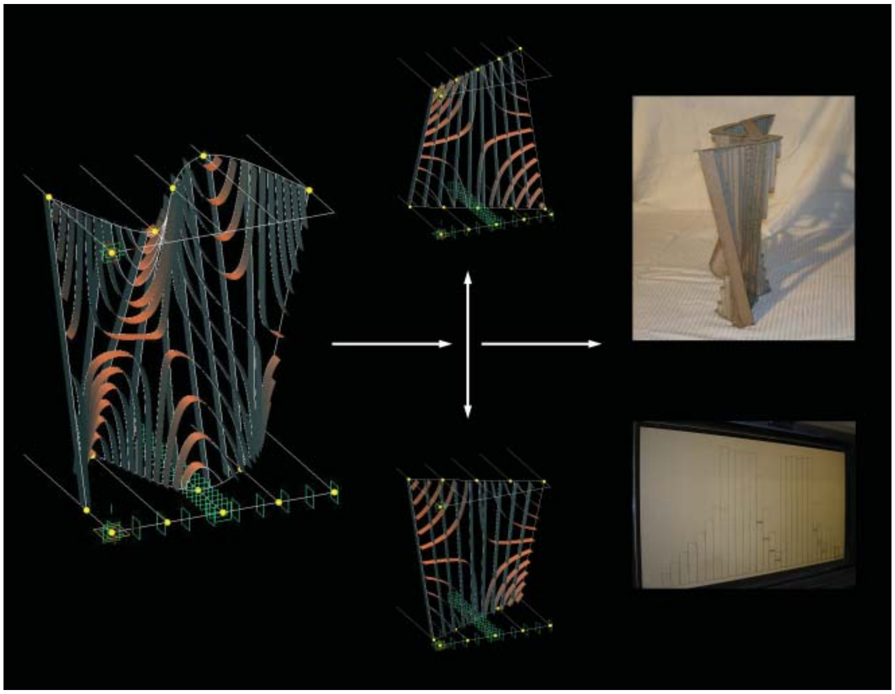
- Variability of the design allows for an analysis of structure and rapid readjustment of the design to suit.
- Dimensional material changes due to scale increases can be factored into the model.
- Drawings and code required for manufacturing and production are instantly updated as required.
- Multiple design scenarios can be visited and revisited without loss of functionality or invested time.
- The completed model can be used for a variety of projects due to its adaptability.

Disadvantages

- As the curvature of the layout curves increase the effective thickness of the displaced surface becomes less. If the curvature becomes too great then the thickness will be insufficient to allow for the necessary building components and insulation. In this case it would be necessary to incorporate a new wall component that replicates the function of the original wall component in its own implementation. While the façade will then develop a characteristic crease in its folding the material and financial benefits of the overall design will still be maintained. The incorporation of the new component will essentially change the direction of extrusion in a direction perpendicular to the facade direction.
- The digital model is relevant only with a design brief that would benefit from its use. A different type of design approach or morphological requirement would necessitate the development of a new model.



116. Instantaneous translation of building configuration



117. Instantaneous translation of building configuration

118. Rendering of potential building configuration.

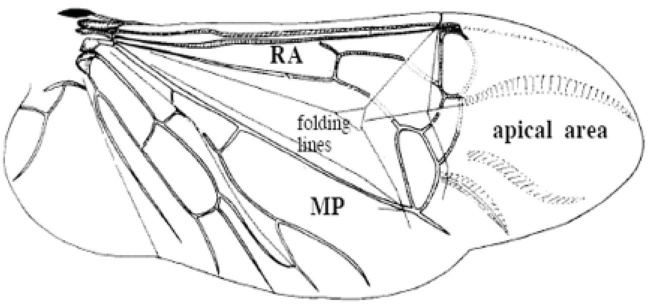
6.3 Design Concept #3 - Folded chevron structure

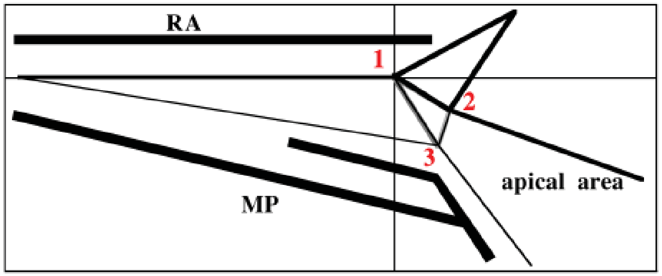
Structure in nature takes many forms which serve to absorb the stresses and environmental conditions imposed on an organism. Of particular interest with regard to this design concept is that of folded and deployable structural forms. This section will examine both static and dynamic deployment with the development of a design for each.

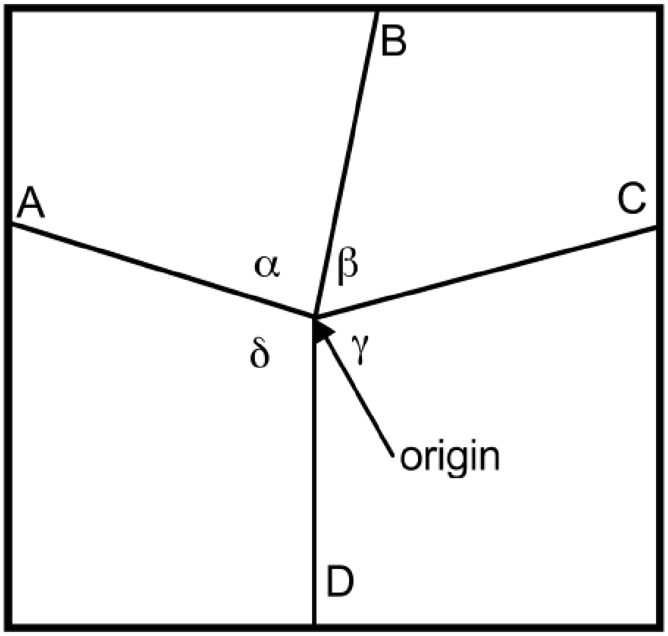
6.3.1 Inspiration

As a variety of natural organisms develop they undergo deployment as a process of attaining their final form. A tightly packaged and folded parcel will unfold according to predetermined patterns that determine its final shape. This process occurs in insect wings, flower petals and plant leaves. Insect wings are an interesting structural group in that different insects display various methods of deployment. The dragonfly wing is deployed by filling its primary structural veins with hemolymph which also serves to prevent it from becoming brittle. The wing itself however maintains its shape once deployed and it is its passive bending that allows for the dragonfly's unique capabilities of flight. (McLendon 2005, p1)

A beetle on the other hand must employ a system for repeated wing deployability as the larger and fragile hind wings must fold in order to be protected by their more robust forewings. The patterns of folding as seen in Figures 119-121, to exhibit rules for folding that







119. Right hind wing of *Priacma Serrata* (bleach beetle) showing folding pattern and the major veins (RA & MP).
120. Digitized folding pattern of Cantharis Livida.
121. Basic mechanism of four panels connected by four folding lines that intersect at one point. Most complex folding patterns consist of a combination of several basic mechanisms.

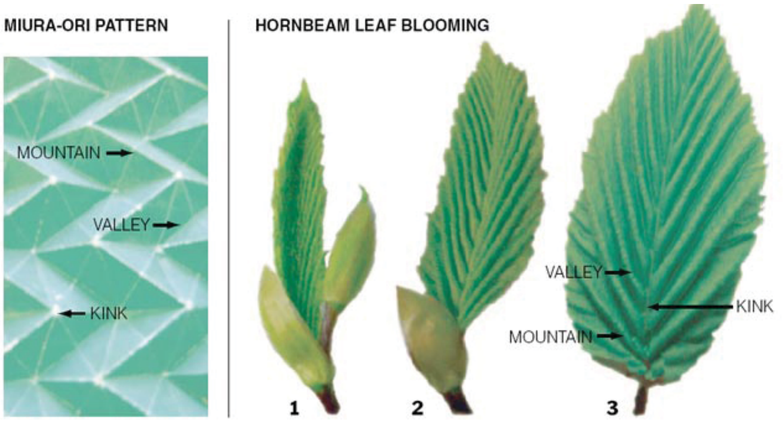
have been described in mathematical terms. (Haas 1998, p2-6)

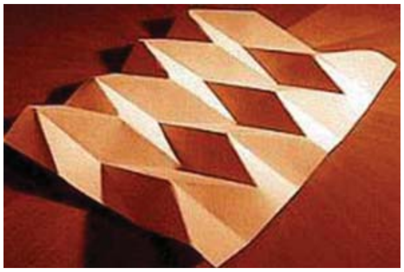
The pattern of unfolding in the beetle wing is similar to that of the hornbeam leaf which has been examined by Julian Vincent, co-director of the department of biomimetics at University of Reading. The similarities of folding structures here also parallel the developments of Koryo Miura, a Japanese space scientist, in the field of origami. In 1970, Miura proposed a paper folding pattern – named Miura-ori – that folds up in two dimensions at right angles thus taking up very little space. Its deployment is also unique in that it unfolds by pulling only on the two ends without subsequent hand repositioning. (Forbes 2000). See Figures 122 & 123.

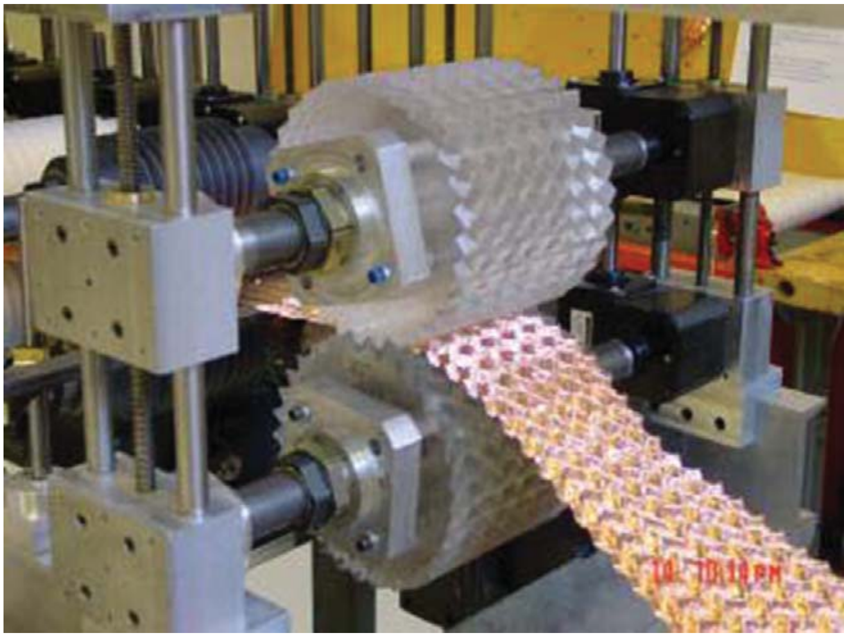
Until recently the Muira-ori technique was difficult to implement on large sheet structures that require a multitude of folds. Research Professors at Rutgers University however, developed a technique to produce a product similar to the Miura-ori folds through roll formers. The product of their research was subjected to stress analysis against conventional honeycomb structures and was found to surpass them in all regards. (Basily 2004a). See Figures 124 & 125.

6.3.2 Design Outline

The issue of deployability in nature is an interesting one due to the relevance it has in both architectural design and construction. The process of deployability in an architectural sense







122. Miura-ori pattern & Hornbeam leaf blooming.
123. Folded sheet with Miura-ori pattern.
124. Continuous sheet folding machine.
125. Continuous sheet folding machine.

can occur in either a static or dynamic way. The two designs developed here, while similar with regard to the base chevron shape that they use, are meant as separate explorations into the applicability of parametric design in the context of deployability. The static deployment design seeks to derive instantly updateable instructional information for laser cutting and brake-forming operations that will yield the proper three dimensional forms. The dynamic deployment design will see the creation of a system that will allow an individual chevron component to be arrayed and manipulated in real time for ease of manufacturing with regard to itself as well as the required structures on which it will depend for their deployment.

These two systems then, represent different approaches to nature's process of sensing and responding. In the first case the chevron pattern will sense (receive input) from the form of the surface to which it is applied and it will respond (create output) for the necessary information related to its manufacture. The second design will be a preliminary platform that serves to act as inputs for the development of additional design products (outputs). These additional products could represent folded and unfolded layout dimensions and coverage areas, deployability paths for the design of collapsible linkages, or volume requirements for storage.

Static Deployment

Architecture as it relates to built form does not arise spontaneously either in its design or physical manifestation. The structure develops through a series of iterative processes that produce a final form. The manufacturing and construction of the design occurs in a number of stages with the structure essentially growing in place. This deployment of built form can thus be thought of not only in a physical sense but also in a temporal sense. The reference to static deployment here represents a process that results in the generation of a static form derived from the deployment of individual constituent parts, in this case plate folded structural members, into a compound curved surface. The form will be created from linear strips of flat plate steel that are cut and folded into the correct orientation. Like the flexible structure of the wing before being stiffened the native form of the flat plate steel exhibits a low resistance to bending which is increased through mechanical folding into a modified Miura-ori pattern. In this case the typically planar chevron will be required to exhibit a slight deformation in one dimension which can be kept small enough to be attained through slight tension induced in construction rather than with mechanical bending in their manufacture.

GenerativeComponents will be used to develop the compound surface and the flattened strips for manufacturing. The surface configuration will be responsive to user input and the Miura-ori pattern derived from it will compensate to suit any desired curved surface.

Dynamic Deployment

As the name suggests, dynamic deployment involves the capacity of the structure to change shape over time. This characteristic is useful in a wide variety of architectural applications from retractable roofs, facades and floor decks. Again, the Miura-ori folding pattern will be used but in a fashion that adheres to a more strict interpretation of its form with regard to the shape and size of the folding units.

6.3.3 Design Product

Static Deployment

The desire for this design is to produce a system of structural chevrons that senses the surface to be populated and responds by altering their shape to suit the requirements of the surface. In this case the size and shape of the chevron will be dictated by input values in the form of the distribution of UV points created on the surface. Once the proper configuration has been realized then the chevron shapes produced will be flattened and exported to a separate fabrication planning file for manufacturing. This design builds on the ideas put forth in the Design Concept #1 where after completion of the chevron population system it will be translated into a new GenerativeComponents Feature.

1. **Identification of key parameters that will contribute to the functionality of the model and allow for the desired level of variability in the design.**

As the design is meant to be quite flexible in its application the parameters defining its generation will be kept to a minimum. The morphological complexity of the design will come from the derivation of the surface to which it is being applied. The graph variables defining the associative parameters therefore will be the following:

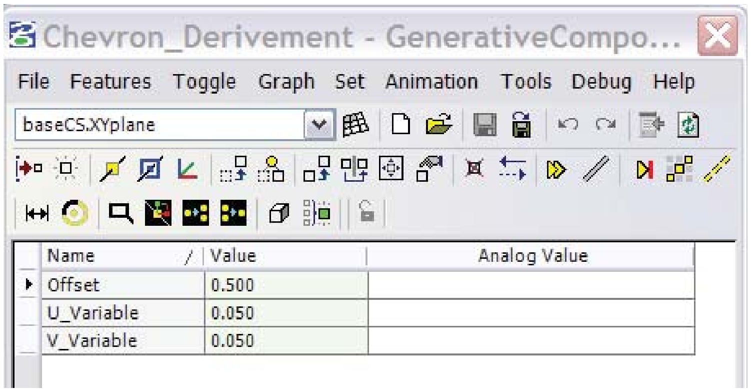
- U points
- V points
- Offset depth

The UV points will define the planar area of the individual chevrons while the offset depth will determine the thickness of the derived surface. See Figure 126.

2. **Development of the design model with a logical progression of generative features.**

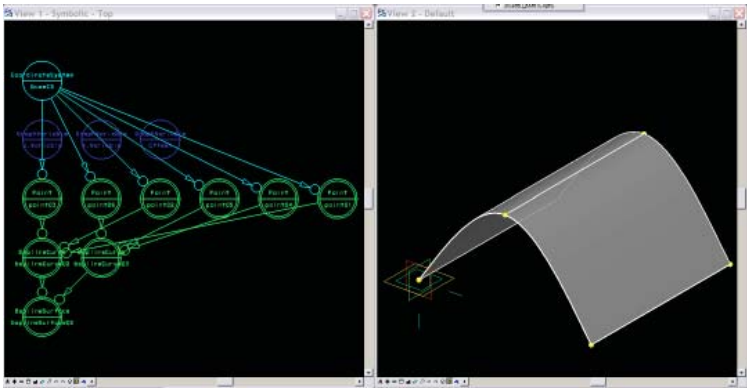
The starting point for the development of the chevron system is to create a surface on which the chevron will be applied. A simple BsplineSurface will be used. It should be noted that the generation of the new GC Feature based on the chevron system will be dependent on an external BsplineSurface and as such the initial surface used to develop the chevrons will not be included in the new GC Feature. The ability of GenerativeComponents to create new Features from a subset of Features within a larger model is very powerful.

The initial BsplineSurface consists of two BsplineCurves that are derived from two sets of three points. See Figure 127.

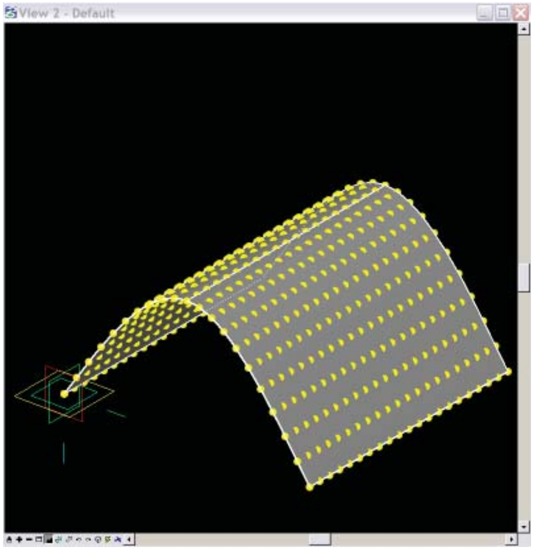


Name	Value	Analog Value
Offset	0.500	
U_Variable	0.050	
V_Variable	0.050	

126. Graph Variables.



127. Initial BsplineSurface.



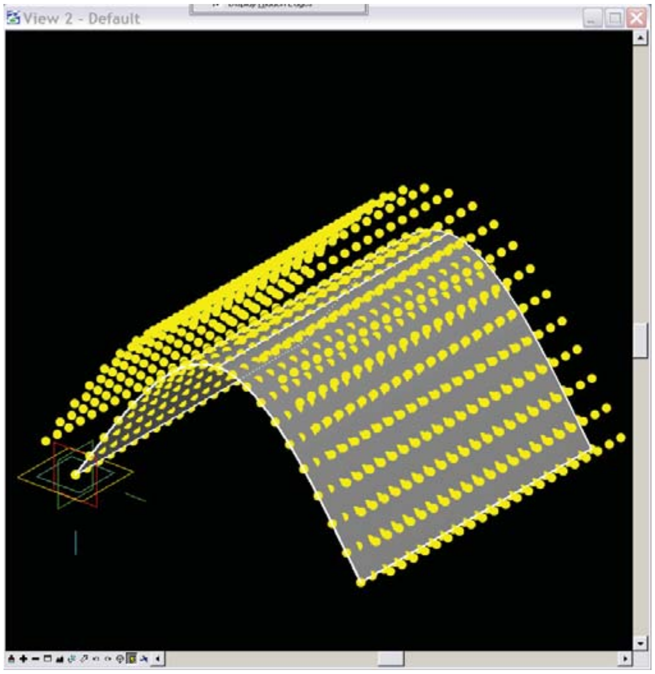
128. UV Points on BsplineSurface.

The BsplineSurface is then populated with a grid of UV points. See Figure 128. The degree to which the surface is divided and populated by the points is dependent on the U_Variable and V_Variable graph variables. An identical configuration of UV points is offset from the surface UVs in order to establish a point field in which the chevrons can be created. See Figure 129. The height of the offset points above the BsplineSurface is dependent on the Offset graph variable.

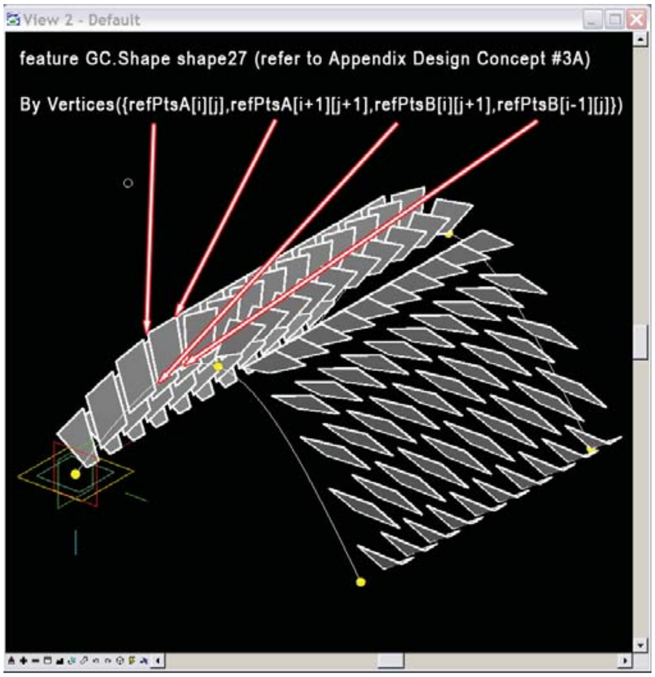
The development of the chevrons is a four part process in that each facet of an individual chevron unit is programmed independently. Each transaction however, creates one facet of every chevron on the surface. See Figure 130. In this way, the whole surface is populated with only four individual steps. See Figure 131. The facets that are created automatically configure themselves to suit the localized morphological conditions of the surface to which they are applied.

Once the chevron facets have been developed and tested for functionality and variability it is then possible to convert the system into a new Generative Component Feature that can be used and applied to future designs much like the use of a Point, Line or Surface. The Generate New Feature Type dialog box allows one to create a name for the new feature as well as define the input and output parameters that the new feature will use for its creation. In this case, the BsplineSurface will be used as the input for the development of the chevrons. The user will be prompted to define values for the Offset, U_Variable and V_Variable. These values may be changed at any time. See Figure 132.

After creating the new feature it can be applied to any BsplineSurface that the user wishes. Here, the feature has been applied to the ruled surface that was created in Design Concept #2. As noted, countless morphological possibilities exist from this single derived feature. See Figure 133, 134 & 135.

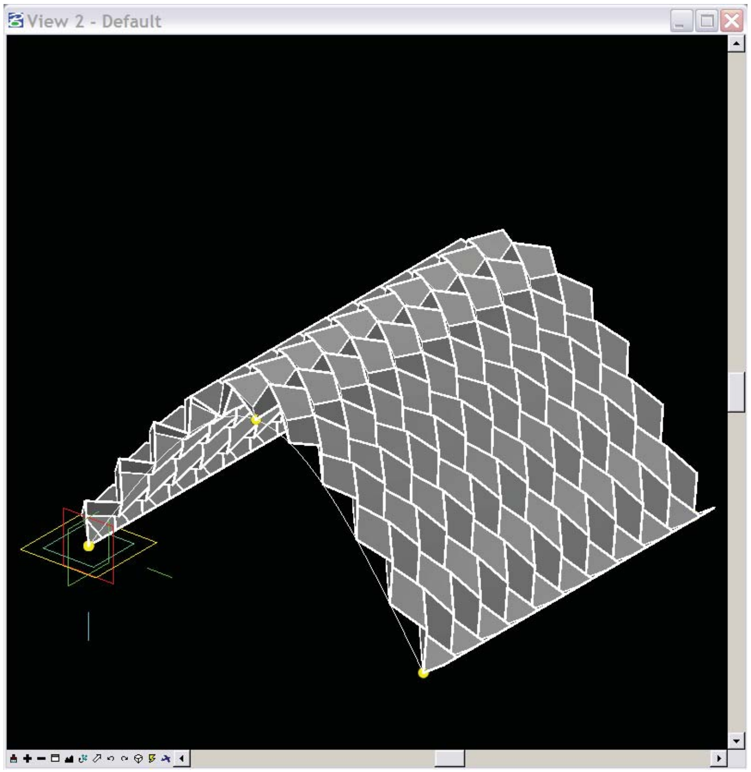


129. Offset points from UV points.

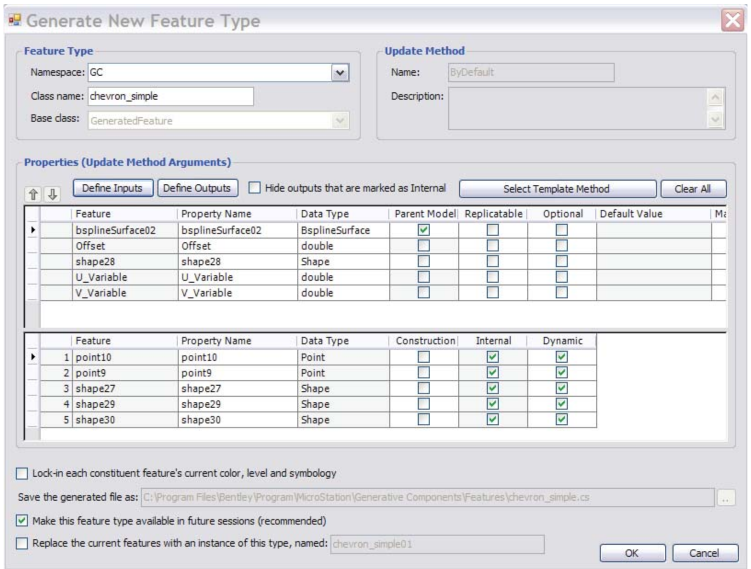


```
feature GC.Shape shape27 (refer to Appendix Design Concept #3A)

By Vertices({refPtsA[i][j],refPtsA[i+1][j+1],refPtsB[i][j+1],refPtsB[i-1][j]})
```

130. Chevron facet development



131. Full chevron facet surface.



Generate New Feature Type

Feature Type

Namespace: GC

Class name: chevron_simple

Base class: GeneratedFeature

Update Method

Name: ByDefault

Description:

Properties (Update Method Arguments)

☐ Hide outputs that are marked as Internal

Feature	Property Name	Data Type	Parent Model	Replicatable	Optional	Default Value
bsplineSurface02	bsplineSurface02	BsplineSurface	☑	☐	☐	
Offset	Offset	double	☐	☐	☐	
shape28	shape28	Shape	☐	☐	☐	
U_Variable	U_Variable	double	☐	☐	☐	
V_Variable	V_Variable	double	☐	☐	☐	

	Feature	Property Name	Data Type	Construction	Internal	Dynamic
1	point10	point10	Point	☐	☑	☑
2	point9	point9	Point	☐	☑	☑
3	shape27	shape27	Shape	☐	☑	☑
4	shape29	shape29	Shape	☐	☑	☑
5	shape30	shape30	Shape	☐	☑	☑

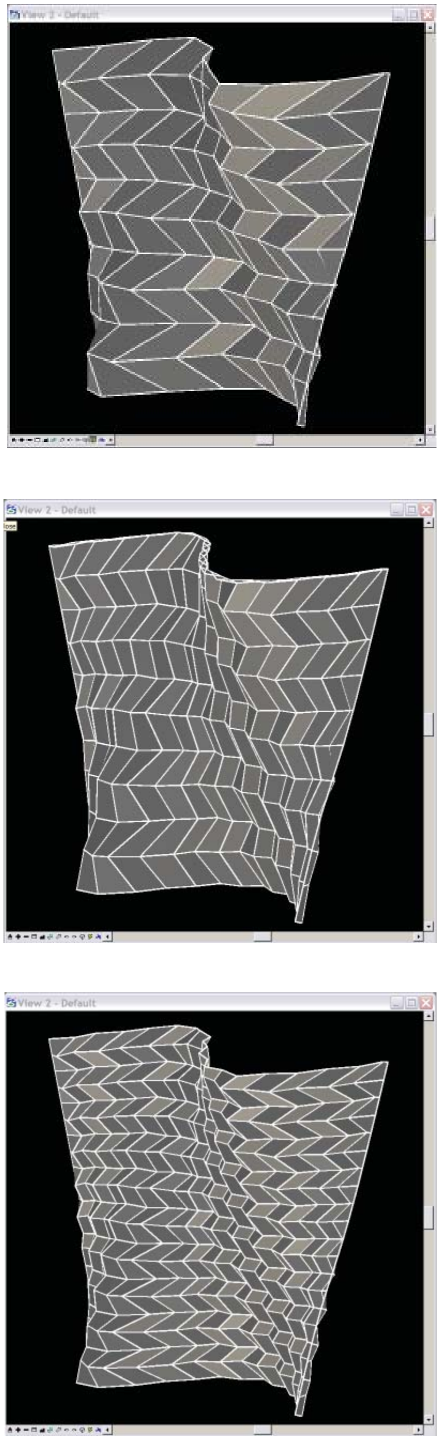
☐ Lock-in each constituent feature's current color, level and symbology

Save the generated file as: C:\Program Files\Bentley\Program\MicroStation\Generative Components\Features\chevron_simple.cs

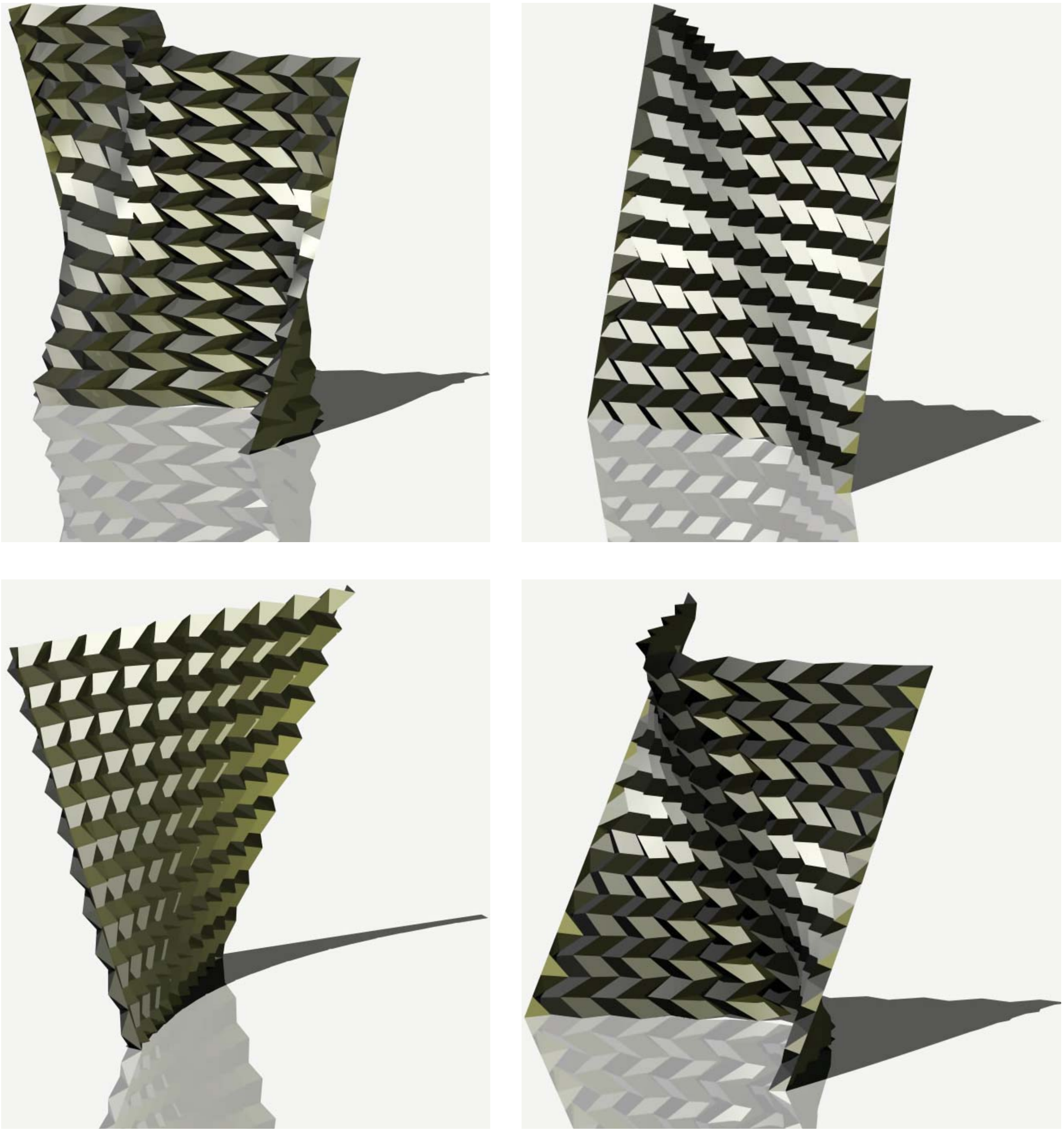
☑ Make this feature type available in future sessions (recommended)

☐ Replace the current features with an instance of this type, named: chevron_simple01

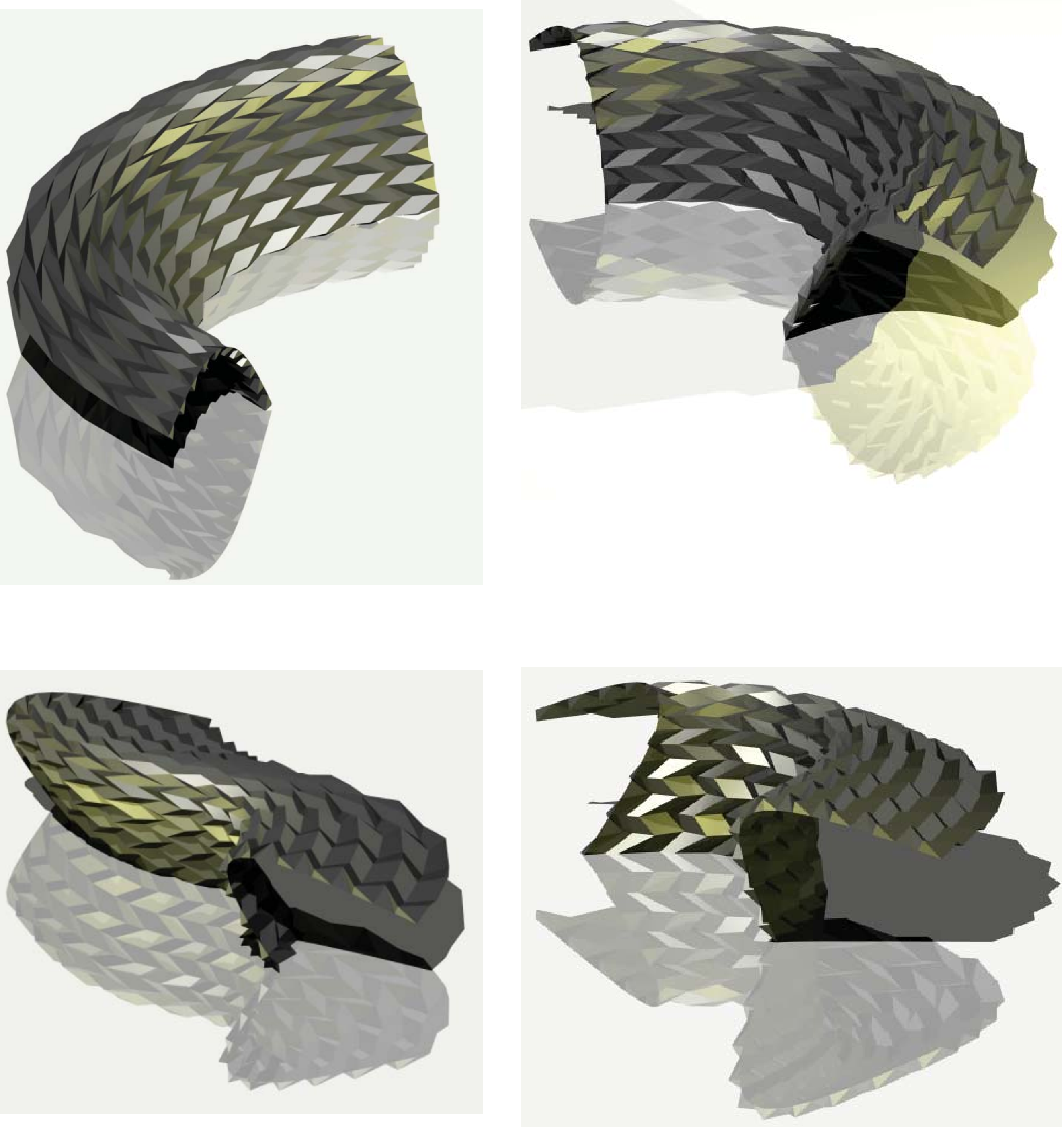
132. Generate Feature Type Interface.



133. Application of chevron component to Design Concept #2



134. Sequence of renderings showing facade reconfiguration and instantaneous chevron component update.



135. Sequence of renderings showing canopy reconfiguration and instantaneous chevron component update.

Dynamic Deployment

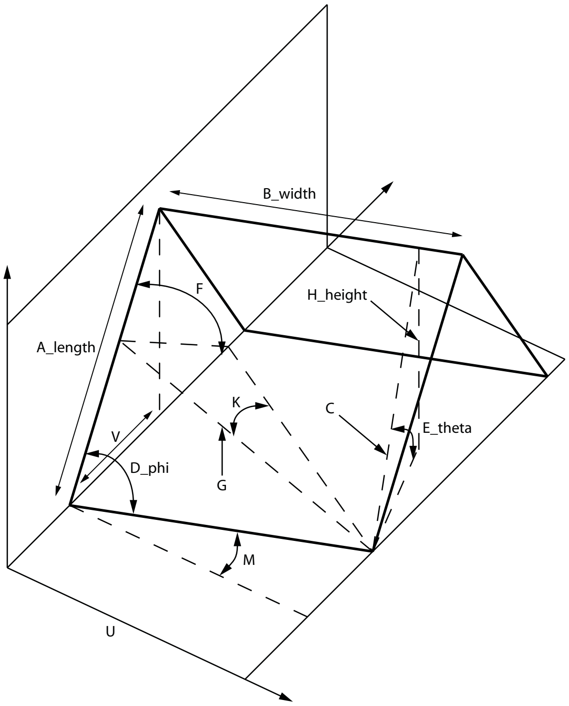
This exercise investigates the associative aspect of GC with regard to dynamic control. While relatively straightforward in morphology, the development of the chevron in this case is based not on the form of the surface to which it is applied, rather its shape is derived from a set of equations whose resulting outputs function as inputs for others. Once the equations determining the control parameters have been set up it will be possible to create a new Feature based on these parameters that can be arrayed in a number of configurations to suit the potential design requirements.

1. **Identification of key parameters that will contribute to the functionality of the model and allow for the desired level of variability in the design.**

According to mathematical equations developed based upon the Miura-ori pattern (Basily 2004a, p4-5) it was possible to create a number of graph variables that would allow for the creation of the dynamic chevron. See Figures 136 & 137.

A_length = input value
B_width = input value
C = (A_length)*sin(D_phi)
D_phi = input value (0-90 deg.)
E_theta = input value (0-90 deg.)
F = $\sin^{-1}$(sin(D_phi)*sin(E_theta))
G = (B_width)*sin(D_phi)
H_height = (A_length)*sin(D_phi)*sin(E_theta)
K = $\sin^{-1}$(tan(F)/tan(D_phi))
M = $\tan^{-1}$(1/(tan(D_phi)*cos(E_theta)))
U = (B_width)*cos(M)
V = (A_length)*cos(K)

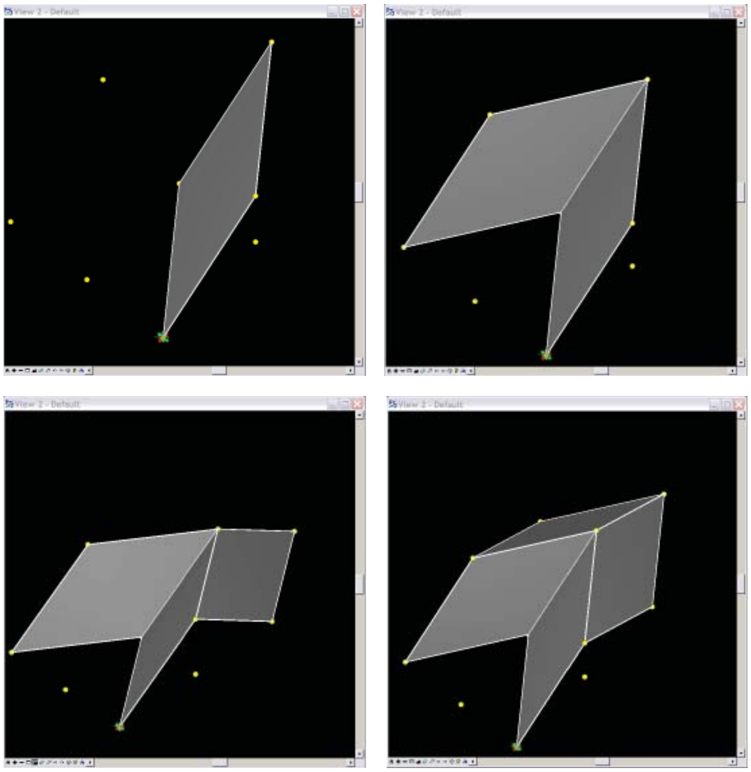
136. Chevron unit equations.



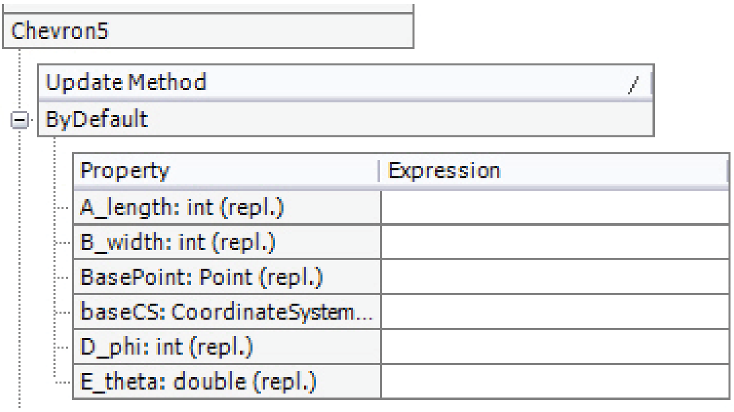
137. One unit of chevron quintet with numeric variables.

2. **Development of the design model with a logical progression of generative features.**

After resolution of the graph variables the next step was to begin creating control points that determine the vertices whose relative positions rely on the associative relationships of the graph variables. With the knowledge that the



138. Progressive development of chevron facets.



Chevron5

Update Method

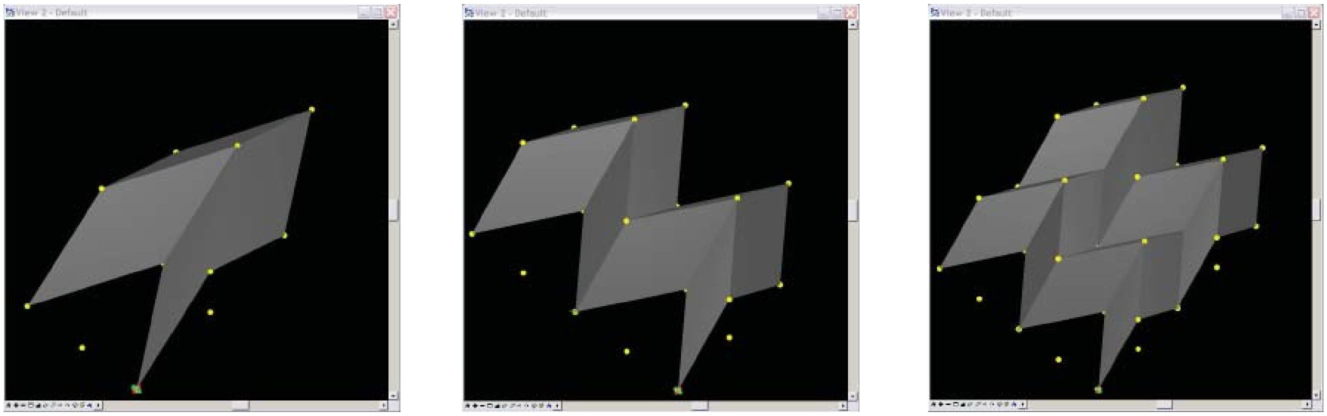
ByDefault

Property	Expression
A_length: int (repl.)	
B_width: int (repl.)	
BasePoint: Point (repl.)	
baseCS: CoordinateSystem...	
D_phi: int (repl.)	
E_theta: double (repl.)	

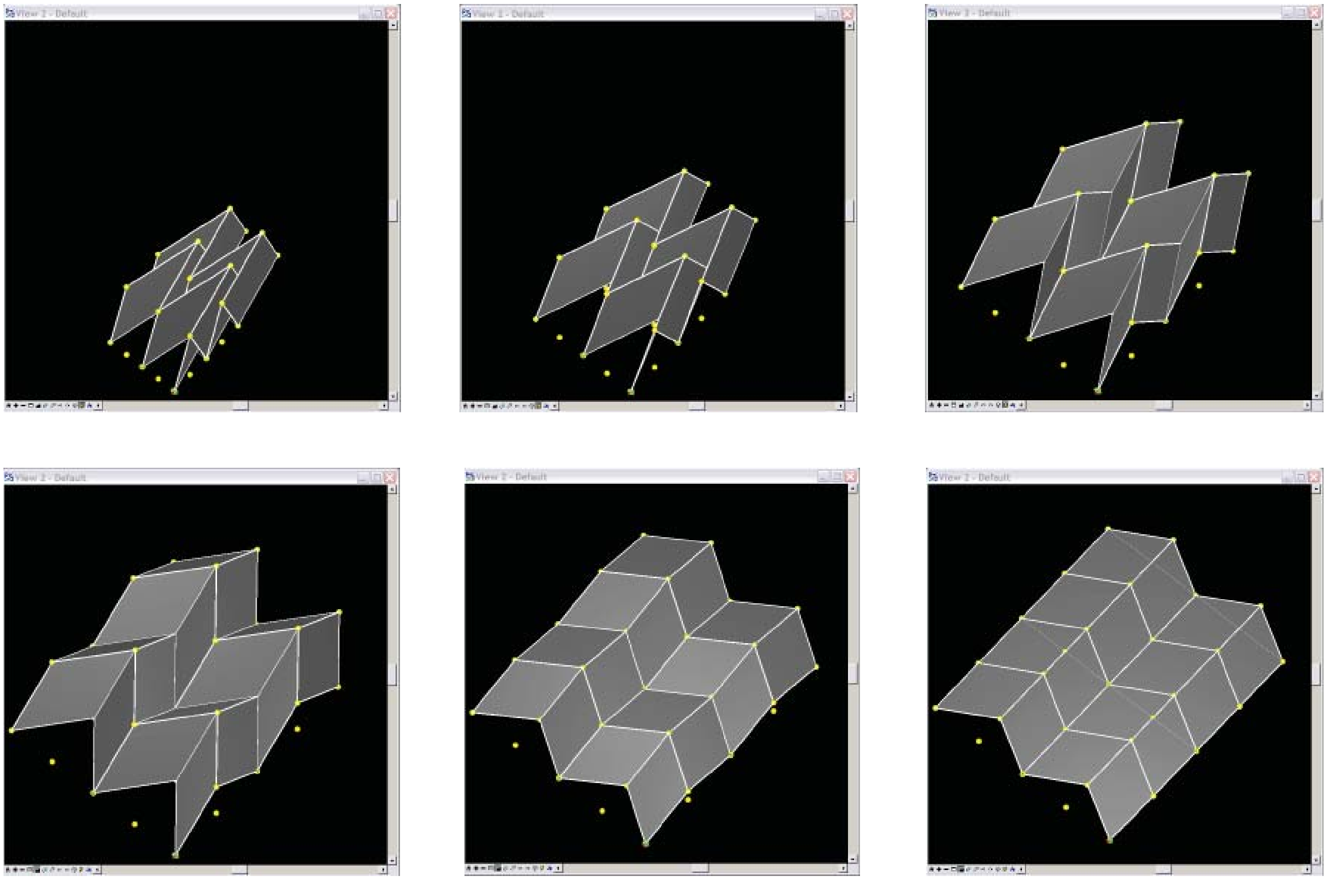
139. Chevron inputs for update method.

chevron unit would be developed into a new GC chevron component Feature it was necessary to develop a methodology for the replication and population of the chevron across a surface or defined area. It was decided that the four facet chevron unit would be placed according to one control point and that subsequent iterations of the chevron would use this point for their creation and placement. The base point was created at the (0, 0, 0) origin of the baseCS. All of the subsequent points and facets are then based on their association to this point or points associated with it. The derived points create what is essentially a point cloud armature on which it was possible to develop the surface facets. The facets were created between the appropriate control points by using the Shape.By Vertices feature. This process was repeated three additional times to create a four sided chevron unit which is able to be altered via manipulation of the input values for the graph variables A_length, B_width, D_phi and E_theta. While this design exercise incorporates variability into all four of these values it is intended for ease of production that these values would not be continuously variable but would begin to form a line of discrete sizes available to the consumer similar to the standardization of sizing for lumber, steel, and the like. However, with the provision for variability the possibility for custom production runs is still maintained. See Figure 138.

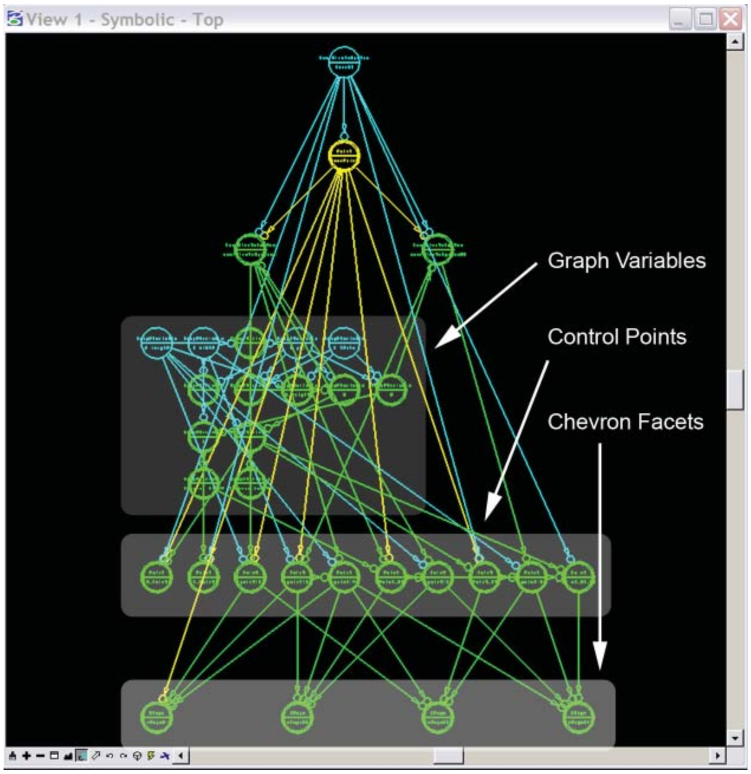
At this point the completed chevron was made into a new Feature in the same manner used in the creation of the chevron Feature in the



140. Population of baseCS with chevron components.



141. Dynamic movement of chevron units.



142. Symbolic view of chevron component derivation and relationships.

design *6.3.3 Static Deployment*. The inputs for the new Feature are a coordinate system, one corner point (BasePoint) for defining its location, numerical values for the length and width of the individual chevron facet dimensions as well as numerical values that define the angle of the chevron above the plane of the coordinate system (E_theta) and angle (D_phi) defining the shape of the physical chevron material from square (90 degrees) to a pronounced diamond (greater than 0 degrees) . The first angle will be infinitely variable, from 0 degrees representing fully open to 90 degrees representing fully closed, which allows for dynamic folding of the chevron. The second angle will be predetermined based on manufacturing requirements. See Figure 139.

Once created, the completed chevron Feature can be replicated to create larger surfaces that are dynamic based upon the graph variable values of E_theta which acts to fold and unfold the chevrons, and D_phi which represents the physical shape of the chevron facets. In a dynamic structure, E_theta would remain continuously variable and D_phi while variable in the development of the digital model would remain static after manufacturing has occurred. See Figures 140 & 141.

Figure 142 shows the complete symbolic view representing the progression from GraphVariables to the chevron facets. It is this assembly that has been converted into a complete chevron feature for application to alternate surfaces.

6.3.4 Design Evaluation

Static Deployment

This design concept strived to develop a system for populating complex surfaces with a structural chevron form that can be derived from flat sheets of CNC formed steel. The idea was based on the process of natural deployability, sensing and responding, self assembly and the power of shape.

GenerativeComponents was once again used extensively in the development of the chevron system. As the system itself can adapt to a variety of surface configurations there is no definite final form for evaluation which is precisely what was intended for the final product.

Advantages

- Throughout the development of a design, changes to the form of the exterior are often necessary to accommodate for programmatic changes, budgetary requirements among others. In keeping with biomimetic principles of design where all of the organism's systems develop in unison rather than in sequence it is beneficial if the architectural design can proceed in a similar manner. This means that all of the building systems should be integrated into the design from the outset. The required structural support for the building is of immense importance and can have profound effects on the placement of other systems such as HVAC. In this case the parametric structural system has the ability to update itself when necessary design changes occur then a lot of time can be saved with regard to recalculation and changes to support system location. Any additional requirements for either structure, finishing or system integration could thus be associated with the chevron feature and become instantly updateable as well.

- The chevron form used has been tested in a variety of loading and crushing tests (Basily 2004a) and has been found to outperform honeycomb panels in all directions. Depending on the application and size that the chevron system is to be produced there are a number of options that can occur for ensuring proper rigidness. Like honeycomb surfaces the ideal scenario would be to cover the chevrons with a double layer of material that is bonded to the chevron substrate. This application would be useful for aircraft applications, door panels, or interior wall partitions. The requirement for the outer skin is to triangulate the pattern and overcome the inherent flexibility of the chevron material which may be cardboard, or a light gauge metal. While not as strong as a dual skin, it is possible to utilize a single sided stiffening skin to allow exposure of the other side for aesthetic purposes. As the scale of the chevrons increase to encompass a building façade it would be possible to use thicker plate steel that is much more resistant to deformation and thus could potentially resist the stresses on it without the need for a skin.

- The unfolded chevron strips are derived from linear strips of flat steel that are cut and brake formed into their proper configuration. The only requirement for plasma or laser cutting would occur along the exterior edges of the strips. This slight zigzag cut pattern would effectively determine the location of the required bends thus reducing manufacturing time.

Areas for Development

- With the exception of a planar surface, any other surface that the chevron system is applied to will result in chevrons of different shape and size. Typical chevrons applied to a flat surface will have facets that are of identical shape and size. Moreover the facets themselves will be planar. To effectively populate a complexly curved surface the facets will be forced out of their planar configuration. While the ability of the chevron material to deform under these conditions may be relatively insignificant with thin gauge materials the situation can intensify with thicker plate materials. This potential problem can be reduced by increasing the number of chevrons or increasing the offset depth.

- At the writing of this thesis GenerativeComponents does not yet support the ability to export the g-code necessary to drive the brake forming operations required to produce the chevron system. This is being addressed and will be contained within future versions of the program.

- The development of the transaction file that produced the chevron system although satisfying the morphological requirements set out in the brief fails to create the chevrons in a linear pattern that would be able to be unfolded for manufacturing. The existing file creates arrays of each individual chevron facet of the four part chevron unit. Upon further development the transaction file will be refined to correct this.

- The individual chevron facets developed in the program are realized by creating a Shape based on vertices within the script. These Shapes are contiguous and non-planar relating to their proper configuration. When these shapes are turned into Solids for export to STL for 3D printing the Shapes generated are non-contiguous and planar which results in an incorrect model. Further development of the model will attempt to create the chevron facets out of BsplineSurfaces instead of Shapes which will allow for proper Solid generation.

- The current version of GC fails to unfold the chevron facet Shapes into the FabricationPlanning model properly. The shapes although non-planar in the 3D model should be forced planar in the Fabrication-

Planning model for proper manufacturing. Again, this should be remedied in future versions.

Dynamic Deployment

This final design concept is a slight departure from the development of non-orthogonal structures in that its form is developed according to mathematical formulas that ensure planarity with respect to the chevrons.

Advantages

- The ability to create complex dependencies between variables examines the reductive instructional methods used in nature. By varying one Graph Variable within the set of variables it is possible not only to dynamically alter the configuration of the design, but it also allows one to view the tangible changes that occur in all of the Graph Variables. The products of these values which can represent areas, lengths, volumes, angles, or any other desirable are instantly available to the designer after every change occurs in the model and can be exported to text files or spreadsheets for further use. For example, the path that a point takes during model deployment can be recorded at a number of stages allowing a direction path to be created that could be used for the design of necessary mechanisms or linkages.

- Once the developed chevron model has been converted into a new Feature it is possible to replicate it over a desired surface. Each independent chevron behaves the same way so that any changes made will propagate throughout the entire model. This drastically reduces the time required in altering a design that requires a large amount of units.

Areas for Development

- The design developed here is derived according to its relationship to the base coordinate system rather than a surface situation. This means that all instances of the chevron feature must be contained either on or in relation to the planar base coordinate system. A progression of the design to allow for the population of a non-planar surface would require that its placement be dependent on a surface rather than a coordinate system much like the static design scenario.

- If the design is to conform to a non-planar configuration then it will also be necessary to integrate graph variables that allow for a certain amount of material deformation within the individual chevron facets. The amount of deformation allowable would be dependent on the material to be used as well as the native shape and size of chevron to be used.

7.0 Discussion and Conclusion

7.1 Discussion

This thesis sought to derive both a method and concepts for architectural design and construction that take their inspiration from biomimicry, essentially the "abstraction of good design from nature" (Aldersey-Williams 2006, p168) The key to an effective biomimetic investigation required the thoughtful selection of observed natural properties that satisfied a well defined list of desirables that were to be reached.

The concepts put forth in the thesis are valuable in that they were produced through a rigorous approach to design based on finding solutions for problems that were delineated at the outset of the investigation. This process allowed for the creation of designs that answered the question of what the design was to do rather than what was to be designed. In approaching the generation of the concepts in this manner, the depth and transferability of the designs becomes greater, where one design can adapt to a multitude of different environments and scenarios. The adaptability of the design comes about through examining not only the design but the process of design as well. Parametric design, namely in the form of the GenerativeComponents design platform, was able to provide a framework for the concepts based on the human genome that allowed them to be effectively developed both digitally and physically. The innovative way in which GenerativeComponents allows the designer to create complex geometries while also giving provision for integrating design intent is very powerful with regard to emulating the evolutionary adaptations present in natural design.

There is however a disjunction between the extensive period of time over which natural evolution occurs versus the relatively short time period for development of architectural design works. While GC allows for the simultaneous progression of multiple designs, the quantitative and qualitative measure of these designs in terms of a proven standard fall short of their natural counterparts that have had countless generations to arrive at their native form. The possibility for an accelerated evolutionary digital design component arises with the prospect of using genetic algorithms in conjunction with GC to produce and analyze a much greater number of design alternatives within the specified design time available. The inbuilt parametric variability of the chosen design means that it remains active and applicable in other design scenarios where all of the previous analysis and design time remains intact within the functionality of the specific GC transaction script. Subsequent designs then can be developed based on the outcome and conclusions derived from previous designs thus promoting a continuous evolutionary design progression on a reduced timeframe.

A parallel between natural design possibilities and the limitation of GC exists, where the evolution of natural organisms or digital designs occurs within and not between possible outcomes. Humans exist in a variety of different configurations with regard to variability of height, weight, color and many other characteristics. However, all of these exist as variations to a well defined template that is not variable, as occurs with bilateral symmetry and the reality of a homeothermic existence. An extensive modification to the human form or systems with regard to the non-variable core design aspects would constitute the development of a new species which would have fundamental differences that could not easily be translated back into their original form. With regard to parametric design, GC contains limitations within it with regard to the amount of design variability that can occur if not thoroughly thought out in the definition of the variables and parameters of the design. If a plan is conceived of as a square, it cannot easily be changed parametrically into a circle. Parametric software then is most useful in providing variability within and not between design concepts. This point is crucial in determining at what point parametric design should enter the design equation. The designer must have a preconceived notion of how and in what form the final product will take if they are to effectively use GenerativeComponents throughout the design process.

The human genome contains all of the information necessary to produce the gene products that derive the organism. The final form of the organism however is not contained within the genetic information, for it is in the interaction with the environment and between the various gene products that produce the respective phenotype. The parametric aspect of the script file contained within GenerativeComponents acts essentially in the same manner, where a set of environmental conditions developed by the designer are created that mix different combinations of gene products, in the form of points, lines, arcs, etc, to arrive at a final form. By varying the conditions within the script file, the designer is able to influence the phenotype of the design without altering the base genes that contain the formational information. In this way, GC provides an interesting corollary to the human genome in that the program itself contains the genetic information to create specific gene components that when combined in a script file produce the desired building phenotype.

The correlation between the human genome and parametric design, in the form of GC, is successful in that provides a developmental design framework that allows designers to comprehend the vast possibilities available with parametric design as well as providing strategies for their implementation. This fact is strengthened with the realization that the developmental and evolutionary limitations inherent in the human genome have paralleled the current limitations in GC and may also provide markers and solutions for possible problematic areas that may arise in the future of GC development.

At present, GenerativeComponents is best suited to the early stages of a design where a large amount of construction detail is not necessary. It is envisioned that the system will continue to be developed to the point where it will be able to output the necessary construction information required for project completion. A true parametric design system would have the capacity to be relevant and contain a fully variable model complete with as much construction detail as required. Additionally, the model would be able to be exported into all necessary AEC computation software for analysis by all parties involved. The advances in BIM have provided a relatively robust parametric design environment, however they approach parametric design in a different manner than GC. The majority of BIM software essentially creates smart objects that carry with them geometric information for manufacturing, documentation and their location within a building. Parametric changes act on the level of individual elements which can in turn affect the other elements like it. GC has the ability to integrate changes beyond the individual element and widespread alterations can influence any number of desired elements. When BIM and GC are able to effectively work together it will create a very robust and highly adaptive parametric design system that can be used throughout the entire design and construction process.

7.2 Conclusion

This thesis presents the development of a process for architectural design that parallels the way in which the human genome contains and provides the information necessary for the creation of natural forms. This process is illustrated with the use of parametric design software in the form of GenerativeComponents, where its application to the design of curvilinear architectural surfaces with integral structure aids in resolving one subset of the larger architectural problem of linking all components and systems of a design parametrically along biomimetic principles.

The AEC community as a whole, much like organisms in nature, must compete in an increasingly competitive environment that rewards efficiency and innovative approaches that find solutions to complex problems. With this being the case it follows that in order to be competitive one must look at ways in which to reduce complexity and increase efficiency not only in the final built form but in the way the form is designed and built as well. It should be noted that the issue of competitiveness does not occur superficially between the resources within firms of architects but more importantly in the wholeness of their design solutions and the ability to perform extensive studies of design alternatives as necessary. The competitive aspect with regard to software innovation and the tools available for design will diminish as they become widely accepted, therefore it is in the process of design where firms will differentiate themselves based on the nature of their design approach and therefore in how they use the tools available to them. The well ordered, logical process of design, as illustrated with the GenerativeComponents parametric model based on the human genome, provides one type of platform that allows the designer to effectively develop and realize innovative design solutions.

Incorporation of parametric software into the process of designing a project allows for a design that derives its solutions through an ordered developmental process acting in concert with an idea for the final design concept. The ability of the architect to step forwards and backwards sequentially through a design as well as to pursue multiple variations of a design simultaneously carries with it the ability to drastically reduce the time invested in exploring potential design alternatives while increasing the time available to effectively complete the design.

Through the visualization of a project in a variety of formats whether they be symbolic, 3D model or transaction based, the designer is able to structure the development of the design to parallel the possible modes of construction that will be utilized. Once again the designer is able express their intent for the design much like Gaudi and his contemporaries were able to do with their own. In order to explain his design for an innovative parabolic arch, Gaudi did not merely draw the form, rather he built a hanging chain model where lines of tension become lines of pure compression when inverted. When draped with cloth, the chain represented a model of his arch. He was able to use the most efficient method available to communicate his design intent to all of the parties involved in the project.

Paul Fletcher, co-founder of the Teamwork Initiative which is a "learn by doing" consortium composed of members from the United Kingdom's most successful AEC firms that are seeking ways to document best practices in collaboration and interoperability and the use of information technology, states that "(in) a conventional project each discipline's design intent is ambiguous to the others because they use different symbology to represent building features and they don't know enough of each other's design intent from a two-dimensional drawing. Designing from scratch in 3D means no need to interpret, because the design intent and the features that would normally be represented by symbols (are physically represented) as 3D objects." (Newton 2003)

The ability to represent a design then not only in a 3D format but in a symbolic and transaction based manner extends the ability of the designer to effectively communicate their design intent to all members of the AEC community involved. Again, the task of creating a design system that links all components of a design parametrically along biomimetic principles is aided in that the information necessary for the realization of the design is available in a format that establishes and allows for a greater cohesiveness and interoperability between design contributors.

In looking at the natural developmental process both in terms of coding and physical maturation of an organism, the framework developed enables the designer to strategically assess the requirements of a project and the relationship of the design disciplines associated with it. This aids in the creation of an efficient work strategy at every level of the design process.

The designer however, must be cognisant of their limitations of digital design knowledge for while it is possible to create an almost limitless array of shapes and forms with the latest digital modeling software that can be easily transferable between AEC contributors, it is quite easy to allow the program itself to drive the morphology of the design.

Architect Greg Lynn outlined a number of key points related to the way in which designers pursue their creativity and the methods in which they use the computer to develop them. In a conversation with Yu-Tung Liu, Lynn stated that it is necessary to master a system so that mastering succeeds, where creativity is not limited by knowledge of the system but succeeds when the system becomes transparent. He went on to state that design is an issue of mathematics and digital technology is inherently sculptural and expressive. In practice, theory should precede technique. (Yu-Tung Liu 2002)

While parametric design is a powerful tool with which to create, organize and produce designs, it is in the way that the designs are developed that is of crucial importance. The mathematical derivation of complex forms defines them in a way that can allow for a layering of complexity with regard to manufacture and construction that would be more difficult in freely developed forms. For example, the layout points, radii and other aspects of a mathematically derived curve can be easily calculated within the program due to the nature of the curve itself.

The formal success of the thesis design concepts for curvilinear surfaces with integral structure lay in their ability to easily adapt to a number of morphological conditions with minimal user intervention. From a design standpoint the architect is able to invest more time in ensuring that the design works well as a cohesive and developed project as a whole rather than manually deriving the individual units that must be created for its completion. With time, the GenerativeComponents program could be populated with an increasing array of unique design components that could act on various scales of the design from form to detail thus compounding the efficiency of the design process.

In concert with a well developed process for architectural design, the thesis also puts forth methods that reduce the complexity of the translation from the digital design to built form. The designs for curved building surfaces with integral structure were able to be developed from linear and planar pieces of material that would require minimal processing to achieve their final form. This has the benefit of reducing the complexity of manufacturing and effectively reduces error and cost as a result. The ability of GenerativeComponents to create relevant manufacturing files directly from the 3D model means that the time required to produce or adjust shop drawings to reflect changes in a design is minimal.

Finally, the conscious effort to derive structural components whose three dimensional conformation necessitates their orientation and placement in a specific manner reduces the number of construction drawings required and the possible confusion associated with the erection of the building. With this being the case, the contractors are able to be given a small set of instructions specifying the process in which the pieces are to be assembled rather than having to create an exhaustive set of drawings that specify the location of each piece. In effect, the final form of the components ensures a proper final form of the structure.

Appendix

A1. Design Concept #1 - GenerativeComponents Script File for 6.1.4.4 Illustrative Example

```
transaction modelBased "Graph Variables Added"
{
    feature GC.GraphVariable Building_Length
    {
        Value                   = 10;
        UsesNumericLimits       = true;
        NumericLowLimit         = 5.0;
        NumericHighLimit        = 15.0;
        SymbolXY                = {102, 102};
    }
    feature GC.GraphVariable Number_of_Floors
    {
        Value                   = 5;
        SymbolXY                = {98, 104};
    }
    feature GC.GraphVariable Floor_Height
    {
        Value                   = 2;
        UsesNumericLimits       = true;
        NumericLowLimit         = 3.0;
        NumericHighLimit        = 4.0;
        SymbolXY                = {98, 105};
    }
    feature GC.GraphVariable Building_Length
    {
        NumericHighLimit        = 20.0;
    }
    feature GC.GraphVariable Building_Width
    {
        Value                   = 10;
        UsesNumericLimits       = true;
        NumericLowLimit         = 5.0;
```

```
        NumericHighLimit          = 20.0;
        SymbolXY                  = {102, 103};
    }
}
transaction modelBased “Point01 added”
{
    feature GC.Point point01
    {
        CoordinateSystem          = baseCS;
        Xtranslation              = 0;
       Ytranslation               = 0;
        Ztranslation              = 0;
        SymbolXY                  = {99, 101};
    }
}
transaction modelBased “Line01 added”
{
    feature GC.Line line01
    {
        StartPoint                = point01;
        Direction                 = baseCS.Xdirection;
        Length                    = Building_Length;
        SymbolXY                  = {99, 103};
    }
}
transaction modelBased “Building_Width GC value changed”
{
    feature GC.GraphVariable Building_Width
    {
       Value                      = Building_Length*0.5;
    }
}
transaction modelBased “Line02 added”
{
    feature GC.Line line02
    {
        StartPoint                = point01;
```

```
        Direction                   = baseCS.Ydirection;
        Length                      = Building_Width;
        SymbolXY                    = {101, 103};
    }
}
transaction modelBased "Line03 offset from Line01"
{
    feature GC.Line line03
    {
        OriginalLine                = line01;
        OffsetDistance              = Building_Width;
        PlaneOrPlanePoint           = baseCS.Zdirection;
        SymbolXY                    = {99, 104};
    }
}
transaction modelBased "Line04 offset from Line02"
{
    feature GC.Line line04
    {
        OriginalLine                = line02;
        OffsetDistance              = Building_Length*(-1);
        PlaneOrPlanePoint           = baseCS.Zdirection;
        SymbolXY                    = {101, 104};
    }
}
transaction modelBased "Line05 added (represents all four vertical lines)"
{
    feature GC.Line line05
    {
        StartPoint                  = {point01,line01.EndPoint,line02.EndPoint,line03.EndPoint};
        Direction                   = baseCS.Zdirection;
        Length                      = Floor_Height*Number_of_Floors;
        SymbolXY                    = {100, 105};
    }
}
transaction modelBased "line06 offset from line04"
{
```

```
    feature GC.Line line06
    {
        OriginalLine                  = line04;
        OffsetDistance                = Series(0,Floor_Height*Number_of_Floors,Floor_Height);
        PlaneOrPlanePoint             = baseCS.YZplane;
        SymbolXY                      = {101, 106};
    }
    feature GC.Line line07
    {
        OriginalLine                  = line02;
        OffsetDistance                = Series(0,Floor_Height*Number_of_Floors,Floor_Height);
        PlaneOrPlanePoint             = baseCS.YZplane;
        SymbolXY                      = {99, 106};
    }
}
transaction modelBased "floor surfaces added"
{
    feature GC.BsplineSurface bsplineSurface02
    {
        StartCurve                    = line07;
        EndCurve                       = line06;
        SymbolXY                      = {100, 107};
    }
}
```

A2. Design Concept #2 - GenerativeComponents Script File for Ruled Surface Structure

```
transaction modelBased "Graph Variable (Facade_Length)"
{
   feature GC.GraphVariable Facade_Length
   {
      Value                          = 10;
      UsesNumericLimits              = true;
      NumericLowLimit                = 1.0;
      NumericHighLimit               = 20.0;
   }
   feature GC.GraphVariable Line_Length
   {
      Value                          = 10;
      UsesNumericLimits              = true;
      NumericLowLimit                = 5.0;
      NumericHighLimit               = 10.0;
      SymbolXY                       = {103, 103};
   }
   feature GC.GraphVariable Primary_Sections
   {
      Value                          = 10;
      UsesNumericLimits              = true;
      NumericLowLimit                = 1.0;
      NumericHighLimit               = 20.0;
   }
   feature GC.GraphVariable Secondary_Sections
   {
      Value                          = 10;
      UsesNumericLimits              = true;
      NumericHighLimit               = 10.0;
      SymbolXY                       = {96, 106};
   }
   feature GC.GraphVariable Wall_Depth
   {
      Value                          = 2;
      UsesNumericLimits              = true;
```

```
        NumericLowLimit              = 1.0;
        NumericHighLimit             = 5.0;
        SymbolXY                     = {96, 104};
    }
    feature GC.GraphVariable Wall_Height
    {
        Value                        = 10;
        UsesNumericLimits            = true;
        NumericLowLimit              = 5.0;
        NumericHighLimit             = 15.0;
        SymbolXY                     = {96, 104};
    }
}
transaction modelBased “Primary_Layout_Line (Base Line)”
{
    feature GC.Line Primary_Layout_Line
    {
        StartPoint                   = baseCS;
        Direction                    = baseCS.Xdirection;
        Length                       = Facade_Length;
        SymbolXY                     = {99, 101};
    }
}
transaction modelBased “cs01 (CS from baseCS)”
{
    feature GC.CoordinateSystem baseCS_Ztranslated
    {
        CoordinateSystem             = baseCS;
        Xtranslation                 = 0;
        Ytranslation                 = 0;
        Ztranslation                 = Wall_Height;
        SymbolXY                     = {102, 101};
    }
}
transaction modelBased “Primary_Layout_Line copy (from Base Line)”
{
    feature GC.Line Primary_Layout_Line_copy01
```

```
    {
        FeatureToCopy              = Primary_Layout_Line;
        From                       = baseCS;
        To                         = baseCS_Ztranslated;
        SymbolXY                   = {102, 102};
    }
}
transaction modelBased "Secondary_Line_Layout_Points (Distribution of Points on Base Line)"
{
    feature GC.Point Secondary_Line_Layout_Points
    {
        Curve                      = Primary_Layout_Line;
        NumberAlongCurve           = 5;
        SymbolXY                   = {97, 102};
    }
}
transaction modelBased "Secondary_Layout_Line (Lines from Secondary_Line_Layout_Points)"
{
    feature GC.Line Secondary_Layout_Line
    {
        StartPoint                 = Secondary_Line_Layout_Points;
        Direction                  = baseCS.Ydirection;
        Length                     = Line_Length;
        SymbolXY                   = {100, 103};
    }
}
transaction modelBased "Secondary_Layout_Line_Ztranslation (Copy of Secondary_Layout_
Line)"
{
    feature GC.Line Secondary_Layout_Line_Ztranslation
    {
        FeatureToCopy              = Secondary_Layout_Line;
        From                       = baseCS;
        To                         = baseCS_Ztranslated;
        SymbolXY                   = {102, 104};
    }
}
```

```
transaction modelBased “Bottom_Distances”
{
    feature GC.Point Bottom_Distances
    {
        Curve                           = Secondary_Layout_Line;
        Distance                        = {5,1,4,6,2};
        SymbolXY                        = {100, 105};
    }
}
transaction modelBased “Top_Distances”
{
    feature GC.Point Top_Distances
    {
        Curve                           = Secondary_Layout_Line_Ztranslation;
        Distance                        = {2,6,2,3,6};
        SymbolXY                        = {102, 105};
    }
}
transaction modelBased “Layout_Curves (Curves through Bottom and Top Distances)”
{
    feature GC.BsplineCurve Layout_Curves
    {
        FitPoints                       = {Bottom_Distances,Top_Distances};
        SymbolXY                        = {101, 106};
    }
}
transaction modelBased “bsplineSurface01 (Through Layout_Curves)”, suppressed
{
    feature GC.BsplineSurface bsplineSurface01
    {
        StartCurve                      = Layout_Curves[0];
        EndCurve                        = Layout_Curves[1];
    }
}
transaction modelBased “Primary_Planes (X section planes)”
{
    feature GC.Plane Primary_Planes
```

```
    {
        Curve                           = Primary_Layout_Line;
        NumberAlongCurve                = Primary_Sections;
        NumberAlongCurveOption          = null;
        SymbolXY                        = {99, 106};
    }
}
transaction modelBased "point02 set (Intersection of Primary_Planes and bottom Layout_
Curves)"
{
    feature GC.Point point02
    {
        Plane                   = Primary_Planes;
        Curve                   = Layout_Curves[0];
        SymbolXY                = {99, 107};
    }
}
transaction modelBased "point03 set (Intersection of Primary_Planes and top bsplineCurve02)"
{
    feature GC.Point point03
    {
        Plane                   = Primary_Planes;
        Curve                   = Layout_Curves[1];
        SymbolXY                = {101, 107};
    }
}
transaction modelBased "Facade_Surface (From point set - point02 and point03)"
{
    feature GC.BsplineSurface Facade_Surface
    {
        Points                  = {point03,point02};
        SymbolXY                = {100, 108};
    }
}
transaction modelBased "Secondary_Planes (Y section planes)"
{
    feature GC.Plane Secondary_Planes
```

```
    {
        Curve                               = Secondary_Layout_Line[2];
        NumberAlongCurve                    = Secondary_Sections;
        NumberAlongCurveOption              = null;
        SymbolXY                            = {98, 106};
    }
}
transaction modelBased "change in section variable"
{
    feature GC.GraphVariable Secondary_Sections
    {
       Value                        = 15;
        NumericHighLimit            = 20.0;
    }
}
transaction modelBased "chevron skin"
{
    feature GC.chevron_skin1 chevron_skin101
    {
        bsplineSurface02            = Facade_Surface;
        Offset                  = 0.5;
        U_Variable                = .05;
       V_Variable                 = .05;
    }
}
transaction modelBased "Section_Curves (Interesection of Secondary_Planes and bsplineSurface01)"
{
    feature GC.Curve Section_Curves
    {
        Plane                   = Secondary_Planes;
        Surface                  = Facade_Surface;
        SymbolXY                    = {98, 109};
    }
}
transaction modelBased "Graph changed by user"
{
```

```
    feature GC.BsplineSurface Facade_Surface
    {
        Display                 = DisplayOption.Hide;
    }
}
transaction modelBased "change in section variable"
{
    feature GC.GraphVariable Secondary_Sections
    {
       Value                   = 20;
    }
}
transaction modelBased "Line03"
{
    feature GC.Line line03
    {
        StartPoint                  = Secondary_Line_Layout_Points[0];
        Direction                   = baseCS.Ydirection;
        Length                      = 2;
        SymbolXY                    = {97, 105};
    }
}
transaction modelBased "line03 related to Graph Variable_Offset Length"
{
    feature GC.Line line03
    {
        Length                      = Wall_Depth;
    }
}
transaction modelBased "bsplineCurve02"
{
    feature GC.BsplineCurve bsplineCurve02
    {
        FitPoints                   = {line03.StartPoint,line03.EndPoint};
        SymbolXY                    = {97, 109};
    }
}
```

```
transaction modelBased "bsplineSurface02 (Section Extrusions)"
{
    feature GC.BsplineSurface bsplineSurface02
    {
            Function  = function (Curves01,Direction01)
                        {
                        Print(Curves01.Count);
                        for (int i = 0; i <= Curves01.Count-1; i++)
                                    {
                                    Print(Curves01[i].Count);
                                    if(Curves01[i].Count==0)
                                    {
                                    BsplineSurface mySurface = CreateChildFeature("BsplineSurface",this);
                                     mySurface.FromRailsAndSweptSections(Direction01,null, Curves01[i]);
                                    }
                                    else
                                    {
                                    for (int j = 0; j < Curves01[i].Count; ++j)
                                    {
                                     BsplineSurface mySurface = CreateChildFeature("BsplineSurface",this);
                                    mySurface.FromRailsAndSweptSections(Direction01,null,
                                    Curves01[i][j]);
                                    }
                                    }
                                    }
                        };
            FunctionArguments        = {Section_Curves,bsplineCurve02};
            SymbolXY                 = {98, 111};
    }
}
transaction modelBased "Hide bsplineSurface01", suppressed
{
    feature GC.BsplineSurface bsplineSurface01
    {
        Display                      = DisplayOption.Hide;
    }
}
```

```
transaction modelBased “Change Wall_Depth”
{
    feature GC.GraphVariable Wall_Depth
    {
        Value                               = 1;
    }
}
transaction modelBased “curve01_Vertical_Secondary_Sections”
{
    feature GC.Curve Vertical_Secondary_Sections
    {
        Plane                               = Primary_Planes;
        Surface                             = Facade_Surface;
        SymbolXY                            = {100, 109};
    }
}
transaction modelBased “Section Curves (Intersection of plane 02 and bsplineSurface01)”
{
    feature GC.BsplineSurface bsplineSurface03
    {
            Function  = function (Curves02,Direction02)
            {
            {
            for (int i = 0; i <= Curves02.Count-1; i++)
            {
            BsplineSurface mySurface = CreateChildFeature(“BsplineSurface”,this);
            mySurface.FromRailsAndSweptSections(Direction02,null, Curves02[i]);
            }
            }
            };
        FunctionArguments           = {Vertical_Secondary_Sections,bsplineCurve02};
        SymbolXY                = {100, 111};
    }
}
transaction modelBased “Change Wall_Depth”
{
    feature GC.GraphVariable Wall_Depth
```

```
    {
        Value                       = 0.5;
        NumericLowLimit             = 0.5;
    }
}
transaction modelBased "New Model - Fabrication Planning and CS"
{
    feature GC.CoordinateSystem Fabrication_Planning_Ruled_SurfaceBaseCS
    {
        Model                       = "Fabrication_Planning_Ruled_Surface";
        SymbolXY                    = {103, 111};
    }
}
transaction modelBased "Shape01", suppressed
{
    feature GC.Shape shape01
    {
        Surface                     = Facade_Surface;
        Tolerance                   = 0.2;
        SymbolXY                    = {102, 110};
    }
}
transaction modelBased "Line01"
{
    feature GC.Line line01
    {
        StartPoint                  = Secondary_Line_Layout_Points[0];
        Direction                   = baseCS.Zdirection;
        Length                      = Wall_Height;
    }
}
transaction modelBased "Point05"
{
    feature GC.Point point05
    {
        Curve                       = Primary_Layout_Line;
        NumberAlongCurve            = Primary_Sections;
```

```
    }
}
transaction modelBased "Point07_Point_grid_on_Facade_Surface", suppressed
{
    feature GC.Point point07
    {
        CoordinateSystem              = baseCS;
        Xtranslation                  = 0;
        Ytranslation                  = 0;
        Ztranslation                  = Series(0,Wall_Height,1);
        Origin                        = point05;
        Replication                   = ReplicationOption.AllCombinations;
    }
}
transaction modelBased "Point06", suppressed
{
    feature GC.Point point06
    {
        Surface                                = Facade_Surface;
        PointToProjectOntoSurface              = point07;
        ProjectionVector                       = baseCS.Ydirection;
    }
}
transaction modelBased "shape03", suppressed
{
    feature GC.Shape shape03
    {
        Points                        = point06;
        Fill                          = true;
    }
}
transaction modelBased "Graph changed by user"
{
    feature GC.BsplineSurface Facade_Surface
    {
        Display                       = DisplayOption.Hide;
    }
```

```
    feature GC.BsplineSurface bsplineSurface02
    {
        Construction            = ConstructionOption.Construction;
    }
        feature GC.BsplineSurface bsplineSurface03
    {
        Construction            = ConstructionOption.Construction;
    }
}
transaction modelBased “fabricationPlanning01 in line with primary structure”, suppressed
{
    feature GC.FabricationPlanning fabricationPlanning01
    {
        CoordinateSystem            = Fabrication_Planning_Ruled_SurfaceBaseCS;
        Shapes                      = shape03;
        Xspacing                    = .25;
       Yspacing                     = .25;
        ForcePlanar                 = true;
    }
}
transaction modelBased “UV_points_on_surface”
{
    feature GC.Point point01
    {
        Surface                     = Facade_Surface;
        U                           = Series(0,1,0.1);
       V                            = Series(0,1,0.1);
        Color                       = 0;
        FillColor                   = -1;
        LineWeight                  = 0;
        LineStyle                   = 0;
        LineStyleName               = “0”;
        Level                       = 1;
        LevelName                   = “Level 1”;
        RoleInGraph                 = RoleInGraphOption.Output;
        RoleInExampleGraph                  = null;
        RoleInComponentDefinition           = null;
```

```
        ComponentInput                    = null;
        ComponentInputReplication         = null;
        ComponentOutput                   = null;
        Replication                       = ReplicationOption.AllCombinations;
        Dynamics                          = DynamicsOption.Dynamics;
        Update                            = UpdateOption.Immediate;
        Construction                      = ConstructionOption.Normal;
        Modify                            = ModifyOption.Fixed;
        Display                           = DisplayOption.Display;
        ConstructionDisplay               = DisplayOption.Hide;
        DimensionDisplay                  = DisplayOption.Hide;
        HandleDisplay                     = DisplayOption.Hide;
        LabelDisplay                      = LabelOption.Hide;
        MaximumReplication                = true;
        Free                              = true;
        ComponentDefinitionInitialization = null;
        SymbolXY                          = {100, 109};
        SymbolicModelDisplay              = null;
        ComputeGeometryInParameterSpace = null;
    }
}
transaction modelBased “point04_UV_Points_on_Surface”
{
    feature GC.Point point04
    {
        Surface                     = Facade_Surface;
        U                           = Series(0,1,0.1);
        V                           = Series(0,1,0.1);
    }
    feature GC.Point point04
    {
        Replication                 = ReplicationOption.AllCombinations;
    }
}
transaction modelBased “Create text style”
{
    feature GC.TextStyle Style01
```

```
    {
        Height                          = 0.05;
        Width                           = 0.05;
        HeightOffset                    = 0.1;
        WidthOffset                     = 0.1;
        TextColor                       = 1;
    }
}
transaction modelBased "shape02_Place shapes on surface"
{
    feature GC.Shape shape02
    {
        Points                          = point04;
        Fill                            = true;
        SkipAlternates                  = false;
        Facet                           = FacetOption.Quads;
        TextStyle                       = Style01;
    }
}
transaction modelBased "Turn construction display on"
{
    feature GC.Shape shape02
    {
        ConstructionDisplay             = DisplayOption.Display;
    }
}
transaction modelBased "Layout shapes on unfold model"
{
    feature GC.FabricationPlanning fabricationPlanning02
    {
        CoordinateSystem                = Fabrication_Planning_Ruled_SurfaceBaseCS;
        Shapes                          = shape02;
        Xspacing                        = 1;
        Yspacing                        = 1;
        TextStyle                       = Style01;
    }
}
```

```
transaction modelBased "Turn on construction display"
{
    feature GC.FabricationPlanning fabricationPlanning02
    {
        ConstructionDisplay            = DisplayOption.Display;
    }
}
```

A3. Design Concept #3A - GenerativeComponents Script File for Static Deployment - Development of chevron_feature01

```
transaction modelBased "Graph Variables"
{
   feature GC.GraphVariable U_Variable
   {
      Value                          = 0.05;
   }
   feature GC.GraphVariable V_Variable
   {
      Value                          = 0.05;
   }
   feature GC.GraphVariable Offset
   {
      Value                          = 0.5;
   }
}
transaction modelBased "create bspline surf"
{
   feature GC.Point point03
   {
      CoordinateSystem               = baseCS;
      Xtranslation                   = 0;
      Ytranslation                   = 4;
      Ztranslation                   = 0;
      HandleDisplay                  = DisplayOption.Display;
   }
   feature GC.Point point06
   {
      CoordinateSystem               = baseCS;
      Xtranslation                   = 4;
      Ytranslation                   = 4;
      Ztranslation                   = 0;
      HandleDisplay                  = DisplayOption.Display;
   }
   feature GC.Point point02
```

```
    {
        CoordinateSystem            = baseCS;
        Xtranslation                = 0;
        Ytranslation                = 2;
        Ztranslation                = -2;
        HandleDisplay               = DisplayOption.Display;
    }
    feature GC.Point point05
    {
        CoordinateSystem            = baseCS;
        Xtranslation                = 4;
        Ytranslation                = 2;
        Ztranslation                = -2;
        HandleDisplay               = DisplayOption.Display;
    }
    feature GC.Point point04
    {
        CoordinateSystem            = baseCS;
        Xtranslation                = 4;
        Ytranslation                = 0;
        Ztranslation                = 0;
        HandleDisplay               = DisplayOption.Display;
    }
    feature GC.Point point01
    {
        CoordinateSystem            = baseCS;
        Xtranslation                = 0;
        Ytranslation                = 0;
        Ztranslation                = 0;
        HandleDisplay               = DisplayOption.Display;
    }
}
transaction modelBased “bsplinecurve02,03 and bsplinesurface02”
{
    feature GC.BsplineCurve bsplineCurve02
    {
        FitPoints                   = {point01,point02,point03};
```

```
    }
    feature GC.BsplineCurve bsplineCurve03
    {
        FitPoints                   = {point04,point05,point06};
    }
    feature GC.BsplineSurface bsplineSurface02
    {
        Curves                      = {bsplineCurve02,bsplineCurve03};
    }
}
transaction modelBased "UV points"
{
    feature GC.Point point9
    {
        Surface                     = bsplineSurface02;
        U                           = Series(0,1.01,U_Variable);
       V                            = Series(0,1.01,V_Variable);
        Replication                 = ReplicationOption.AllCombinations;
    }
}
transaction modelBased "create point offsets"
{
    feature GC.Point point10
    {
        Surface                     = bsplineSurface02;
        U                           = Series(0,1.01,U_Variable);
       V                            = Series(0,1.01,V_Variable);
        D                           = Offset;
        Replication                 = ReplicationOption.AllCombinations;
    }
}
transaction modelBased "hide BsplineSurface and points9/10"
{
    feature GC.BsplineSurface bsplineSurface02
    {
        Display                     = DisplayOption.Hide;
    }
```

```
    feature GC.Point point9
    {
        Display                              = DisplayOption.Hide;
    }
    feature GC.Point point10
    {
        Display                              = DisplayOption.Hide;
    }
}
transaction modelBased “lacing chevron 1”
{
    feature GC.Shape shape27
    {
        Function                             = function (refPtsA,refPtsB)
                                             {
                                             for (value i = 0; i < refPtsA.Count; i=i+2)
                                             {


                       value shapeRow1 = CreateChildFeature(“Shape”,this);
                       for (value j= 1; j < refPtsA.Count; j=j+2)
                       {
                       CreateChildFeature(“Shape”,shapeRow1).ByVertices({refPtsA[i][j],refPtsA[i+1][
                       j+1],refPtsB[i][j+1],refPtsB[i-1][j]}, true);
                       }
                       }
                       };
        FunctionArguments  = {point10,point9};
    }
}
transaction modelBased “lacing chevron 2”
{
    feature GC.Shape shape28
    {
        Function = function (refPtsA,refPtsB)
                       {
                       for (value i = 0; i < refPtsA.Count; i=i+2)
```

```
                        {
                        value shapeRow1 = CreateChildFeature(“Shape”,this);
                        for (value j= 1; j < refPtsA.Count; j=j+2)
                        {

                        CreateChildFeature(“Shape”,shapeRow1).ByVertices({refPtsA[i][j],refPtsA[i+1][
                        j-1],refPtsB[i][j-1],refPtsB[i-1][j]}, true);
                        }
                        }
            };
        FunctionArguments = {point10,point9};
    }
}
transaction modelBased “lacing chevron 3”
{
    feature GC.Shape shape29
    {
        Function = function (refPtsA,refPtsB)
                        {
                        for (value i = 1; i < refPtsA.Count; i=i+2)
                        {
                        value shapeRow1 = CreateChildFeature(“Shape”,this);
                        for (value j= 1; j < refPtsA.Count; j=j+2)
                        {
                        CreateChildFeature(“Shape”,shapeRow1).ByVertices({refPtsB[i][j],refPtsB[i+1][j
                        -1],refPtsA[i][j-1],refPtsA[i-1][j]}, true);
                        }
                        }
            };
        FunctionArguments = {point10,point9};
    }
}
transaction modelBased “lacing chevron 4”
{
    feature GC.Shape shape30
    {
        Function = function (refPtsA,refPtsB)
```

```
                        {
                        for (value i = 1; i < refPtsA.Count; i=i+2)
                        {
                        value shapeRow1 = CreateChildFeature(“Shape”,this);
                        for (value j= 1; j < refPtsA.Count; j=j+2)
                        {
                        CreateChildFeature(“Shape”,shapeRow1).ByVertices({refPtsB[i][j],refPtsB[i+1][j
                        +1],refPtsA[i][j+1],refPtsA[i-1][j]}, true);
                        }
                        }
            };
        FunctionArguments = {point10,point9};
    }
}
transaction modelBased “Graph changed by user”
{
    feature GC.GraphVariable Offset
    {
        Value                               = 0.289;
        UsesNumericLimits                   = true;
        NumericLowLimit                     = 0.1;
        NumericHighLimit                    = 1.0;
}
transaction modelBased “Graph changed by user”
{
    feature GC.GraphVariable Offset
    {
        Value                               = 0.181;
        UsesNumericLimits                   = true;
        NumericLowLimit                     = 0.1;
        NumericHighLimit                    = 1.0;
    }
}
transaction modelBased “Hide Shapes”
{
    feature GC.Shape shape27
    {
```

```
        Display                    = DisplayOption.Hide;
    }
    feature GC.Shape shape28
    {
        Display                    = DisplayOption.Hide;
    }
    feature GC.Shape shape29
    {
        Display                    = DisplayOption.Hide;
    }
    feature GC.Shape shape30
    {
        Display                    = DisplayOption.Hide;
    }
}
transaction modelBased "Hide shape27"
{
    feature GC.Shape shape27
    {
        Display              = DisplayOption.Display;
    }
}
transaction modelBased "State at which new feature type, GC.chevron_feature01, created"
{
}
}
```

A4. Design Concept #3A - GenerativeComponents Script File for Static Deployment - Application of chevron_feature01 to Variable BsplineSurface

In this example, chevron_feature01 was applied to a BsplineSurface, where movement of the layout points from point01 to point 07 produced the variety of forms displayed in Figure 135 on p116.

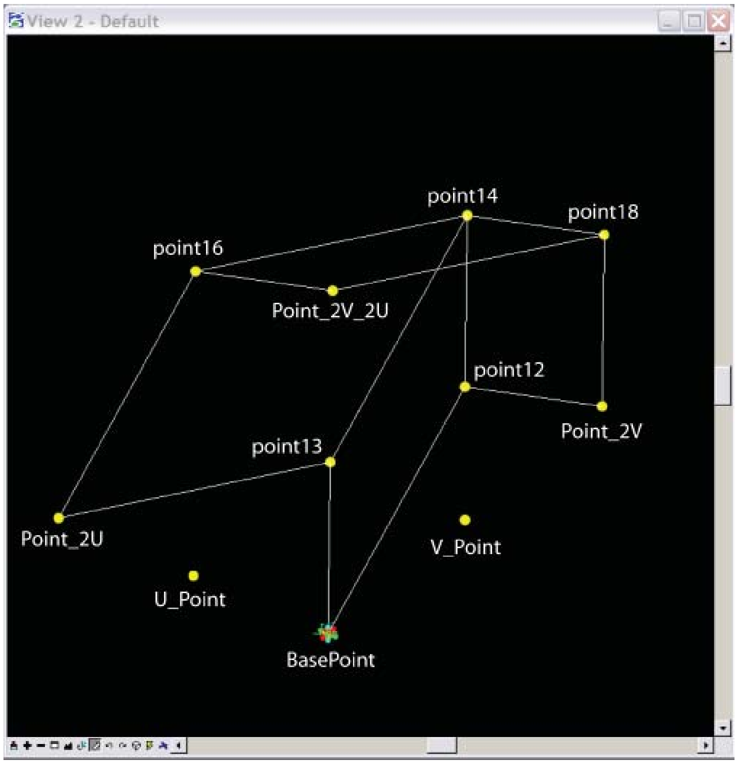
```
transaction modelBased “points”
{
    feature GC.Point point07
    {
        CoordinateSystem            = baseCS;
        Xtranslation                = <free> (4.33763791286761);
        Ytranslation                = <free> (-2.13718670164055);
        Ztranslation                = <free> (6);
        HandleDisplay               = DisplayOption.Display;
    }
    feature GC.Point point06
    {
        CoordinateSystem            = baseCS;
        Xtranslation                = <free> (8.10086460967013);
        Ytranslation                = <free> (-3.89916514844596);
        Ztranslation                = <free> (4);
        HandleDisplay               = DisplayOption.Display;
    }
    feature GC.Point point05
    {
        CoordinateSystem            = baseCS;
        Xtranslation                = <free> (-2.02912063409083);
        Ytranslation                = <free> (18.6724857105255);
        Ztranslation                = <free> (0.0);
        HandleDisplay               = DisplayOption.Display;
    }
    feature GC.Point point04
    {
        CoordinateSystem            = baseCS;
        Xtranslation                = <free> (3.30917495547002);
```

```
        Ytranslation               = <free> (16.9371610512656);
        Ztranslation               = <free> (0.0);
        HandleDisplay              = DisplayOption.Display;
    }
    feature GC.Point point03
    {
        CoordinateSystem           = baseCS;
        Xtranslation               = <free> (5.53435513852012);
        Ytranslation               = <free> (12.8185243895387);
        Ztranslation               = <free> (0.0);
        HandleDisplay              = DisplayOption.Display;
    }
    feature GC.Point point02
    {
        CoordinateSystem           = baseCS;
        Xtranslation               = <free> (4.73560612610537);
        Ytranslation               = <free> (6.9007426939103);
        Ztranslation               = <free> (0.0);
        HandleDisplay              = DisplayOption.Display;
    }
    feature GC.Point point01
    {
        CoordinateSystem           = baseCS;
        Xtranslation               = <free> (0.223273654899217);
        Ytranslation               = <free> (0.240351271830272);
        Ztranslation               = <free> (0.0);
        HandleDisplay              = DisplayOption.Display;
    }
}
transaction modelBased “Move points”
{
    feature GC.Point point04
    {
        Xtranslation               = <free> (1.02425502809233);
        Ztranslation               = <free> (0.499971694588041);
    }
    feature GC.Point point05
```

```
    {
        Xtranslation                  = <free> (4.56675558979121);
        Ytranslation                  = <free> (25.2348995107612);
        Ztranslation                  = <free> (-0.540368485870081);
    }
}
transaction modelBased “layout curves”
{
    feature GC.BsplineCurve bsplineCurve02
    {
        FitPoints                     = {point05,point04,point03,point02,point01};
    }
    feature GC.BsplineCurve bsplineCurve01
    {
        FitPoints                     = {point01,point07,point06};
    }
}
transaction modelBased “BsplineSurface”
{
    feature GC.BsplineSurface bsplineSurface01
    {
        Rail0                         = bsplineCurve02;
        Section0                      = bsplineCurve01;
    }
}
transaction modelBased “Move points”
{
    feature GC.Point point03
    {
        Xtranslation                  = <free> (3.62376586076137);
    }
    feature GC.Point point04
    {
        Xtranslation                  = <free> (1.24573118327263);
        Ztranslation                  = <free> (0.0130934139742951);
    }
    feature GC.Point point05
```

```
    {
        Xtranslation                     = <free> (1.24524186693465);
        Ytranslation                     = <free> (28.8261168662667);
    }
    feature GC.Point point06
    {
        Xtranslation                     = <free> (6.18053087807 35);
        Ytranslation                     = <free> (-5.41945972728544);
    }
    feature GC.Point point07
    {
        Xtranslation                     = <free> (-3.17996928267702);
        Ytranslation                     = <free> (2.41950379753293);
    }
}
transaction modelBased "chevron"
{
    feature GC.chevron_feature chevron_feature01
    {
        bsplineSurface02                 = bsplineSurface01;
        Offset                           = -.5;
        U_Variable                       = 0.05;
        V_Variable                       = 0.05;
    }
}
transaction modelBased "Hide BSplineSurface01"
{
    feature GC.BsplineSurface bsplineSurface01
    {
        Display                          = DisplayOption.Hide;
    }
}
```

A5. Design Concept #3B - GenerativeComponents Script File for Application of Dynamic Deployment



```
transaction modelBased "Create Graph Variables"
{
    feature GC.GraphVariable A_length
    {
        Value                      = 5;
        SymbolXY                   = {92, 106};
    }
    feature GC.Point BasePoint
    {
        CoordinateSystem           = baseCS;
        Xtranslation               = <free> (0);
        Ytranslation               = <free> (0);
        Ztranslation               = <free> (0.0);
        HandleDisplay              = DisplayOption.Display;
        SymbolXY                   = {96, 102};
```

```
}
feature GC.GraphVariable D_phi
{
   Value                         = 39;
    UsesNumericLimits            = true;
    NumericHighLimit             = 180.0;
    SymbolXY                     = {95, 106};
}
feature GC.GraphVariable B_width
{
   Value                         = 5;
    SymbolXY                     = {93, 106};
}
feature GC.GraphVariable E_theta
{
   Value                         = 45;
    UsesNumericLimits            = true;
    NumericHighLimit             = 180.0;
    SymbolXY                     = {96, 106};
}
feature GC.GraphVariable H_height
{
   Value                         = A_length*Sin(D_phi)*Sin(E_theta);
    SymbolXY                     = {95, 107};
}
feature GC.GraphVariable C
{
   Value                         = A_length*Sin(D_phi);
    SymbolXY                     = {94, 106};
}
feature GC.GraphVariable M
{
   Value                         = Atan(1/(Tan(D_phi)*Cos(E_theta)));
    SymbolXY                     = {97, 107};
}
feature GC.GraphVariable F
{
```

```
        Value                           = Asin(Sin(D_phi)*Sin(E_theta));
        SymbolXY                        = {94, 107};
    }
    feature GC.GraphVariable G
    {
        Value                           = B_width*Sin(D_phi);
        SymbolXY                        = {93, 107};
    }
    feature GC.GraphVariable K
    {
        Value                           = Asin(Tan(F)/Tan(D_phi));
        SymbolXY                        = {96, 107};
    }
    feature GC.GraphVariable V
    {
        Value                           = A_length*Cos(K);
        SymbolXY                        = {94, 108};
    }
    feature GC.GraphVariable U
    {
        Value                           = B_width*Cos(M);
        SymbolXY                        = {93, 108};
    }
    feature GC.GraphVariable E_theta
    {
        NumericHighLimit                = 90.0;
    }
    feature GC.GraphVariable D_phi
    {
        Value                           = 45;
        NumericHighLimit                = 90.0;
    }
}
transaction modelBased "Change BaseCS SymbolSize"
{
    feature GC.CoordinateSystem baseCS
    {
```

```
        SymbolSize                  = .25;
        SymbolXY                    = {96, 100};
    }
}
transaction modelBased “V_point”
{
    feature GC.Point V_Point
    {
        Origin                      = BasePoint;
        Direction                   = baseCS.Xdirection;
        Distance                    = V;
        SymbolXY                    = {92, 111};
    }
}
transaction modelBased “U_point”
{
    feature GC.Point U_Point
    {
        Origin                      = BasePoint;
        Direction                   = baseCS.Ydirection;
        Distance                    = U;
        SymbolXY                    = {93, 111};
    }
}
transaction modelBased “cs_01”
{
    feature GC.CoordinateSystem coordinateSystem01
    {
        Origin                      = BasePoint;
        CoordinateSystem            = baseCS;
        RotationAngle               = -K;
        Axis                        = AxisOption.Y;
        SymbolXY                    = {94, 104};
    }
    feature GC.Point point12
    {
        Origin                      = BasePoint;
```

```
        Direction                   = coordinateSystem01.Xdirection;
        Distance                    = A_length;
        SymbolXY                    = {94, 111};
    }
}
transaction modelBased "cs_02"
{
    feature GC.CoordinateSystem coordinateSystem02
    {
        Origin                      = BasePoint;
        CoordinateSystem            = baseCS;
        RotationAngle               = 90-M;
        Axis                        = AxisOption.Z;
        SymbolXY                    = {98, 104};
    }
}
transaction modelBased "point13"
{
    feature GC.Point point13
    {
        Origin                      = BasePoint;
        Direction                   = coordinateSystem02.Xdirection;
        Distance                    = B_width;
        SymbolXY                    = {95, 111};
    }
}
transaction modelBased "point14"
{
    feature GC.Point point14
    {
        Origin                      = point13;
        Direction                   = coordinateSystem01.Xdirection;
        Distance                    = A_length;
        SymbolXY                    = {96, 111};
    }
}
transaction modelBased "Chevron Face shape01"
```

```
{
    feature GC.Shape shape01
    {
        Vertices                    = {BasePoint,point12,point14,point13,};
        Fill                        = true;
        SymbolXY                    = {92, 114};
    }
}
transaction modelBased "Point_2U 2*U"
{
    feature GC.GraphVariable Chevron_Width
    {
        Value                       = 2*U;
        SymbolXY                    = {93, 109};
    }
    feature GC.GraphVariable Chevron_Length
    {
        Value                       = 2*V;
        SymbolXY                    = {94, 109};
    }
    feature GC.Point Point_2U
    {
        Origin                      = BasePoint;
        Direction                   = coordinateSystem01.Ydirection;
        Distance                    = Chevron_Width;
        SymbolXY                    = {97, 111};
    }
}
transaction modelBased "point16 distance A from Point_2U"
{
    feature GC.Point point16
    {
        Origin                      = Point_2U;
        Direction                   = coordinateSystem01.Xdirection;
        Distance                    = A_length;
        SymbolXY                    = {98, 111};
    }
```

```
}
transaction modelBased "Chevron face shape02"
{
    feature GC.Shape shape02
    {
       Vertices                         = {point13,point14,point16,Point_2U};
        Fill                            = true;
        SymbolXY                        = {95, 114};
    }
}
transaction modelBased "Point_2V 2*V"
{
    feature GC.Point Point_2V
    {
        Origin                          = BasePoint;
        Direction                       = baseCS.Xdirection;
        Distance                        = 2*V;
        SymbolXY                        = {99, 111};
    }
}
transaction modelBased "point18 distance B from Point_2V"
{
    feature GC.Point point18
    {
        Origin                          = Point_2V;
        Direction                       = coordinateSystem02.Xdirection;
        Distance                        = B_width;
        SymbolXY                        = {100, 111};
    }
}
transaction modelBased "Point_2V_2U distance 2*V from Point_2U"
{
    feature GC.Point Point_2V_2U
    {
        Origin                          = Point_2U;
        Direction                       = baseCS.Xdirection;
        Distance                        = 2*V;
```

```
        SymbolXY                    = {101, 111};
    }
}
transaction modelBased "Chevron face shape03"
{
    feature GC.Shape shape03
    {
        Vertices                    = {Point_2V,point12,point14,point18};
        Fill                        = true;
        SymbolXY                    = {98, 114};
    }
}
transaction modelBased "Chevron face shape04"
{
    feature GC.Shape shape04
    {
        Vertices                    = {Point_2V_2U,point18,point14,point16};
        Fill                        = true;
        SymbolXY                    = {101, 114};
    }
}
transaction modelBased "Line 2V"
{
    feature GC.Line Line_2Vto2V_2U
    {
        StartPoint                  = Point_2V;
        EndPoint                    = Point_2V_2U;
    }
}
transaction modelBased "State at which new feature type, GC.Chevron4, created"
{
}
transaction modelBased "Second Chevron Added"
{
    feature GC.GraphVariable E_theta
    {
        Value                       = 64.8;
```

```
    }
    feature GC.Chevron4 chevron401
    {
        A_length                    = 5;
        B_width                     = 5;
        BasePoint                   = Point_2U;
        baseCS                      = baseCS;
        D_phi                       = 60;
        E_theta                     = E_theta;
    }
}
transaction modelBased "Third Chevron Added"
{
    feature GC.Chevron4 chevron402
    {
        A_length                    = 5;
        B_width                     = 5;
        BasePoint                   = Point_2V;
        baseCS                      = baseCS;
        D_phi                       = 60;
        E_theta                     = E_theta;
    }
}
transaction modelBased "Fourth Chevron Added"
{
    feature GC.Chevron4 chevron403
    {
        A_length                    = 5;
        B_width                     = 5;
        BasePoint                   = Point_2V_2U;
        baseCS                      = baseCS;
        D_phi                       = 60;
        E_theta                     = E_theta;
    }
}
```

Glossary

Allele

Any one of a number of viable DNA codings occupying a given locus (position) on a chromosome. Usually alleles are DNA sequences that code for a gene, but sometimes the term is used to refer to a non-gene sequence. An individual's genotype for that gene is the set of alleles it happens to possess. In a diploid organism, one that has two copies of each chromosome, two alleles make up the individual's genotype.

Diploid

Containing two sets of homologous chromosomes and hence two copies of each gene or genetic locus.

Enzyme

A protein functioning as a catalyst in living organisms, which promotes specific reactions or groups of reactions.

Genotype

Genetic constitution of an individual cell or organism, in the form of DNA. Together with the environmental variation that influences the individual, it codes for the phenotype of the individual.

Microfilaments

Helical protein filament formed by the polymerization of globular actin molecules. A major constituent of the cytoskeleton of all eucaryotic cells and part of the contractile apparatus of skeletal muscle.

Microtubules

Tubes that are the structural entity for eucaryotic flagella, have a role in maintaining cell shape, and function as mitotic spindle fibers.

Nucleotide

Chemical compound that consists of a heterocyclic base, a sugar, and one or more phosphate groups. In the most common nucleotides the base is a derivative of purine or pyrimidine, and the sugar is the pentose (five-carbon sugar) deoxyribose or ribose.

Nucleotides are the structural units of RNA, DNA, and several cofactors - CoA, FAD, FMN, NAD, and NADP. In the cell they play important roles in energy production, metabolism, and signaling.

Phenotype

The phenotype of an individual organism is either its total physical appearance and constitution or a specific manifestation of a trait, such as size, eye color, or behavior that varies between individuals. Phenotype is determined to some extent by genotype, or by the identity of the alleles that an individual carries at one or more positions on the chromosomes. Many phenotypes are determined by multiple genes and influenced by environmental factors. Thus, the identity of one or a few known alleles does not always enable prediction of the phenotype.

Polypeptide

Linear polymer composed of multiple amino acids. Proteins are large polypeptides, and the two terms can be used interchangeably.

References

Aalbers, Cas, H. Bier, S. Boer, and K. Oosterhuis. 2004. *File to factory and real-time behaviour in ONL-architecture. Paper presented at Fabrication: examining the digital practice of architecture*, Cambridge & Toronto.

Aish, R. 2004. *Generative Components Introduction: Bentley Institute Course Guide*. Bentley Systems, Incorporated.

Alberts, B., D. Bray, J. Lewis, M. Raff, K. Roberts, and J. Watson. 1994. *Molecular Biology of the Cell.* 3rd edition ed. New York, NY: Garland Publishing.

Aldersey-Williams, H. 2003. *New animal architecture.* London: Laurence King Publishing.

Anders, P., and M. Phillips. 2004. *Arch-OS: An operating system for buildings. Paper presented at Fabrication: examining the digital practice of architecture*, Cambridge & Toronto.

Ban, Shigeru. 2000. *Projects in process to japanese pavilion, expo 2000 hannover*. 2nd Edition ed. Tokyo: Mitsuo Kawagoe.

Basily, B. B., and E. A. Elsayed. 2004a. *A continuous folding process for sheet materials.* International Journal of Materials and Product Technology 21, (1/2/3): 217-238.

----------. 2004b. *Dynamic axial crushing of multi-layer core structures of folded chevron patterns.* International Journal of Materials and Product Technology 21, (1/2/3): 169-185.

Batcheler, B., and I. Howell. *Building information modeling two years later - huge potential, some success and several limitations*. in The Laiserin Letter [database online]. 20052006]. Available from http://www.laiserin.com/features/bim/newforma_bim.pdf.

Beesley, P., R. MacHenry, and E. von Michalofski. 2001. *On growth and form - textiles and the engineering of nature*. Toronto: Textile Museum of Canada.

Bell, B., and A. Vrana. 2004. *Digital Tectonics: Structural Patterning of Surface Morphology. Fabrication: examining the digital practice of architecture.* Cambridge & Toronto.

Benyus, J. 1997. *Biomimicry: Innovation inspired by nature*. 1st Edition ed. New York, NY: William Morrow.

Benyus, J. 2004. *Biomimicry.* Pop!Tech Lecture Series. Camden, Maine. Recorded Oct. 22, 2004.

Biomimicry Guild. *Biomimicry methodology*. 2006 [cited 06/21 2005]. Available from http://www.biomimicry.net/essent_resourc.html.

Brock, T., M. Madigan, J. Martinko, and J. Parker. 1994. *Biology of microorganisms*. New Jersey: Prentice Hall.

Burry, Mark. 1999. *Paramorph*. Architectural Design 69, (10): 78-83.

Coffey, J. *Automating thermal plate forming for hull plates*. in The Navy Joining Center [database online]. Columbus, Ohio, 2006]. Available from http://www.ewi.org.

Coop Himmelb(l)au. *Seoul performing arts center*. [cited 09/15 2005]. Available from http://www.emergentarchitecture.com/projects.php?id=8.

Craven, J. *Master Architects: Antoni Placid Guillem Gaudi Cornet*. [cited 05/01 2006]. Available from http://architecture.about.com/library/blgaudi.htm.

DECOI: Goulthorpe, Mark. 1999. *Aegis hyposurface autoplastic to alloplastic*. Architectural Design 69, (10): 60-65.

Denton, M., and C. Marshall. 2001. *Laws of form revisited*. Nature 410, (March 22): 417.

Edwardson, S. 2003. *High-powered laser forming of metallic components.* Dies and Stamping News(April), http://www.lasers.org.uk/paperstore/Forming4.pdf.

EMERGENT Architecture. *Lattice house*. [cited 09/15 2005a]. Available from http://www.emergentarchitecture.com/projects.php?id=9.

----------. *Radiant hydronic house*. [cited 09/15 2005b]. Available from http://www.emergentarchitecture.com/projects.php?id=7.

Faludi, J. *Biomimicry for green design (A how-to)*. 2006 [cited 10/28 2005]. Available from http://www.worldchanging.com/archives/003680.html#more.

Forbes, P. 2000. *Folders find a new wrinkle*. Guardian (Thursday, November 23), http://www.guardian.co.uk.

Forte, C. *Packaging material innovation: 3-D folded structures*. 2005 [cited 01/05 2006]. Available from http://www.iopp.org/files/Forte_Chris_Rutgers_3D_Structures.pdf?pageid=pageid.

Foster, N., D. Sudjic, and S. de Grey. 2001. *Norman foster and the british museum.* New York: Prestel.

Gallaher, M., A. O'Connor, J. Dettbarn, and L. Gilday. 2004. *Cost analysis of inadequate interoperability in the U.S. capital facilities industry*. Gaithersburg, Maryland.: U.S. Department of Commerce Technology Administration. National Institute of Standards and Technology., NIST GCR 04-867.

Godfaurd, J., D. Clements-Croome, and G. Jeronimidis. March 2005. *Sustainable building solutions: A review of lessons from the natural world*. Building and Environment 40, (3): 319.

Goodsell, D. 1996. *Our Molecular Nature: The Body's Motors, Machines, and Messages*. New York: Springer-Verlag.

Haas, F. 1998. *Wing folding in insects: A natural, deployable structure*. Paper presented at IUTAM-IASS Symposium on Deployable Structures: Theory and Applications, Cambridge, UK.

Hall, P. 2003. *Bend the rules of structure: A brooklyn metalworking shop with an unlikely name may hold the key to 21st century shapemaking*. Metropolis.

Jirapong, K., and R. Krawczyk. 2002. *Architectural forms by abstracting nature*. Paper presented at Generative Art Conference, Milan, Italy.

Kilian, A. 2004. *Linking digital hanging chain models to fabrication. Paper presented at Fabrication: examining the digital practice of architecture*, Cambridge & Toronto.

Knecht, B. *Brave new solid-state, carbon fiber world.* [cited 06/15 2005]. Available from http://archrecord.construction.com/innovation/2_Features/0310carbonfiber.asp

Lalvani, H. 1999. *Meta Architecture*. Architectural Design 69, (10): 32-37.

----------. 1977. *Transpolyhedra dual transformations by explosion - implosion*. Papers in theoretical morphology. New York: Haresh Lalvani.

Lang, R. J. 1996. *A computational algorithm for origami design*. Paper presented at Proceedings of the 12th Symposium on Computational Geometry, Philadelphia.

Leslie, T. 2003. *Form as diagram of forces - the equiangular spiral in the work of Pier Luigi Nervi*. Journal of Architectural Education 57, (2): 45.

Lindsey, Bruce. 2001. *Digital Gehry material resistance / digital construction*. The information technology revolution in architecture. Boston: Birkhauser.

Liu, Yu-Tung. 2002. *Defining Digital Architecture*: FEIDAD 2001 Award. Boston: Birkhauser.

Mahadevan, L., and S. Rica. March 2005. *Self-organized origami*. Science 307, (5716): 1740.

Martinelli, Cesar. 1967. *Gaudi: His life, his theories, his work*. Barcelona: Colegio Oficial de Arquitectos de Cataluna y Baleares.

McGraw Hill. 1997. *McGraw Hill Encyclopedia of Science and Technology.* New York: McGraw Hill.

McLendon, W.R. 2005. *Investigation into Dragonfly Wing Structure and Composite Fabrication*. [Cited 05/05, 2006]. Available from http://tiims.tamu.edu/2005summerREU/papers/McLendon.pdf.

Newton, Randall. 2003. *Lessons from London, Hotbed of AEC Innovation*. AEC News. [Cited 06/31,2--6]. Available from http://aecnews.com/articles/695.aspx.

Nonell, Juan Bassegoda. 2000. *Antonio gaudi: Master architect*. 1st Edition ed. New York, NY: Abbeville Press.

O'Connor, John and Robertson, Edmund. 2006. *D'Arcy Wentworth Thompson*. MacTutor History of Mathematics Archive. Available from http://www-history.mcs.st-andrews.ac.uk/Mathematicians/Thompson_D'Arcy.html. Accessed May, 2006.

Pearson, David. 2001. *New Organic Architecture: The Breaking Wave.* University of California Press.

Perez, S. 2004. *The synthetic sublime - notes on physical/digital space and form. Paper presented at Fabrication: examining the digital practice of architecture*, Cambridge & Toronto.

Portoghesi, Paolo. April 30, 2006. *Concrete architecture*. Materia(No. 46), http://www.materia.it/materia.

Pottman, H, and J. Wallner. 2001. *Computational line geometry. Mathematics and visualization.*, eds. Gerlad Farin, Hans-Christian Hege, David Hoffman, Christopher R. Johnson and Konrad Polthier. New York: Spinger-Verlag.

Sanders, K. 2004. *Why building information modeling isn't working...yet*. Architectural Record, http://archrecord.construction.com/features/digital/archives/0409feature-1.asp.

Schlaich, J., H. Schober, and K. Kurschner. 2005. *New trade fair in milan - grid topology and structural behaviour of a free-formed glass-covered surface*. International Journal of Space Structures 20, (1): 1-14.

Schodek, D., M. Bechtold, K. Griggs, K. Martin, and M. Steinberg. 2005. *Digital Design and Manufacturing - CAD/CAM Applications in Architecture and Design.* New Jersey: John Wiley & Sons.

Schrodinger, E. 1944. *What is life? the physical aspect of the living cell.* Cambridge: Cambridge University Press.

Seebohm, T., and W. Wallace. 1998. *Rule-based representation of design in architectural practice*. Automation in Construction 8, : 73.

Spuybroek, Lars. 2004a. *D-tower*. In NOX machining architecture., 158. New York, NY: Thames & Hudson.

----------. 2004b. *ECB*. In NOX machining architecture., 290. New York, NY: Thames & Hudson.

----------. 2004c. *obliqueWTC*. In NOX machining architecture., 260. New York, NY: Thames & Hudson.

----------. 2004d. *OffTheRoad_5Speed.* In NOX machining architecture., 114. New York, NY: Thames & Hudson.

----------. 2004e. *Pompidou two*. In NOX machining architecture., 304. New York, NY: Thames & Hudson.

----------. 2004f. *Soft office*. In NOX machining architecture., 216. New York, NY: Thames & Hudson.

----------. 2004g. *Son-O-house*. In NOX machining architecture., 174. New York, NY: Thames & Hudson.

----------. 2004h. *wetGRID*. In NOX machining architecture., 139. New York, NY: Thames & Hudson.

Srinivasan, A.V. March 1996. *Smart biological systems as models for engineered structures.* Materials Science & Engineering: C 4, (1): 19.

Thompson, D. 1963. *On growth and form*. 2nd Edition, reprinted ed. Vol. 1. Deventer, Holland: Cambridge University Press.

Tsui, E. 1999. *Evolutionary architecture - nature as a basis for design*. New York: John Wiley & sons.

----------. Tsui design & research inc. 2006. Available from http://www.tdrinc.com/. 2006.

Vincent, J. 2002. *Deployable structures*, ed. S. Pellegrino. 1st Edition ed. Vol. Vol. 1. Vienna: Springer.

Vogel, S. 1998. *Cat's paws and Catapults. Mechanical worlds of nature and people.* 1st Edition. New York: W.W. Norton & Company.

Vollers, K. 2001. *Twist & build creating non-orthogonal architecture.* Trans. Lynn de Graaf and Brenda Vollers-King. Rotterdam: 010 Publishers.

Wade, L. G., Jr. 1995. *Organic chemistry.* 3rd Edition ed. New Jersey: Prentice Hall.

Weinstock, M. 2004. *Morphogenesis and the Mathematics of Emergence*. Architectural Design Vol 74; Part 3, p10-17. Great Britain: Academy Group Ltd.

Whitesides, G., Gryboski, B. 2002. *Self Assembly at All Scales*. Science 29 March. Vol. 295, no. 5564. p.2418-2421

Yeang, K. 2002. Reinventing the Skyscraper: A Vertical Theory of Urban Design. New Jersey: John Wiley & Sons.