Design Exploration through Bidirectional Modeling of Constraints

by
Axel Kilian
Master of Science in Architectural Studies,
Massachusetts Institute of Technology, 2000
Diplom Ingenieur, University of the Arts Berlin, 1998

Submitted to the Department of Architecture in Partial Fulfillment of the Requirement for the Degree of

Doctor of Philosophy in Architecture, Design and Computation

at the

Massachusetts Institute of Technology

February 2006

Signature of author

Department of Architecture, MIT
Jan 13th, 2006

Certified by:

Takehiko Nagakura
Associate Professor of Design and Computation
Department of Architecture, MIT
Dissertation Advisor

Accepted by:

Yung Ho Chang
Professor of Architecture
Head of the Department of Architecture,
MIT

Design Exploration through Bidirectional Modeling of Constraints

Abstract

by
Axel Kilian

Submitted to the Department of Architecture on Jan 13, 2006, in Partial Fulfillment of the Requirement for the Degree of Doctor of Philosophy, in Architecture: Design and Computation.

February 2006

Today digital models for design exploration are not used to their full potential. The research efforts in the past decades have placed geometric design representations firmly at the center of digital design environments. In this thesis it is argued that models for design exploration that bridge different representation aid in the discovery of novel designs. Replacing commonly used analytical, uni-directional models for linking representations, with bidirectional ones, further supports design exploration. The key benefit of bidirectional models is the ability to swap the role of driver and driven in the exploration.

The thesis developed around a set of design experiments that tested the integration of bidirectional computational models in domain specific designs. From the experiments three main exploration types emerged. They are: branching explorations for establishing constraints for an undefined design problem; illustrated in the design of a concept car. Circular explorations for the refinement of constraint relationships; illustrated in the design of a chair. Parallel explorations for exercising well-understood constraints; illustrated in a form finding model in architecture. A key contribution of the thesis is the novel use of constraint diagrams developed to construct design explorers for the experiments. The diagrams show the importance of translations between design representations in establishing design drivers from the set of constraints. The incomplete mapping of design features across different representations requires the redescription of the design for each translation. This redescription is a key aspect of exploration and supports design innovation.

Finally, this thesis argues that the development of design specific design explorers favors a shift in software design away from monolithic, integrated software environments and towards open software platforms that support user development.

Takehiko Nagakura
Associate Professor of Design and Computation
Department of Architecture, MIT
Advisor

Dissertation Committee

Takehiko Nagakura
Associate Professor of Design and Computation
Department of Architecture, MIT
Advisor

William J. Mitchell
Professor Architecture and Media Arts and Sciences
Department of Architecture
Reader

Una May O'Reilly
Research Scientist
Computer Science and Artificial Intelligence Laboratory,
Massachusetts Institute of Technology
Reader

Acknowledgments

I am grateful to my dissertation committee, Takehiko Nagakura, William J. Mitchell and Una-May O'Reilly, for their insightful comments and their support throughout my dissertation. I would like to thank William J. Mitchell for providing the research opportunities on concept cars in the Smart Cities Group of the MIT Media Lab. I would like to thank the Department of Architecture and in particular the Computation Group for supporting my research throughout the Ph.D. program. I would like to thank John Ochsendorf, Simon Greenwold, Barbara Cutler, Eric Demaine, and Marty Demaine for their contributions and comments in the context of the hanging chain workshop at MIT. I want to also thank Robert Aish, Lars Hesselgren, and Hugh Whitehead from the smart geometry group for their support and the many inspiring workshops.

The interest in the dissertation topic developed during my Masters thesis on the reformulation of space through a visual language, while being a guest in the aesthetic and computation group by John Maeda. For his support and excitement in computational design I thank him. Also the friendships I made in acg have lasted long past my time there. In particular I would like to thank Ben Fry for his relentless support in all matters. Without Ben Fry's and Casey Reas' development of processing.org much of the work would have not been possible.

Without my colleagues in the computation group it would have not been possible to complete the dissertation. Thank you especially to Joao Rocha, Paul Keel, Mine Ozkar, Sotirios Kotsopoulos, Maria Thompson, Mitchell Joachim, Yanni Loukissas, Stelios Dritsas, Carlos Barrios, Franco Variani and Saeed Arida. Also thank you to all the members of the smart cities group, especially to Patrik Künzler, Peter Schmitt for all the great collaborations. I would also like to thank Michael Fox for his support and collaboration over many years. Also thank you to the computation faculty, especially William L. Porter, Larry Sass, and Terry Knight for their support and the opportunities for research and teaching.

And last but not least thank you to Susanne Schindler, who I want to dedicate this dissertation to.

Axel Kilian

Biography

Axel Kilian came to MIT on a Fulbright scholarship in 1998 after completing his professional architecture degree at the University of the Arts in Berlin, Germany. He holds a Master of Science in Architectural Studies from MIT, which he completed in 2000. In 2001 he began his research toward a Ph.D. in the Design and Computation Group in the Department of Architecture at MIT.
His research has focused on the application of programming in design, in particular on the development of bidirectional design systems and the use of constraints in design exploration. He has collaborated widely in architectural competitions and research, among others with the Aesthetics and Computation Group (acg), the Smart Cities Group, and the Tangible Media Group at the MIT Media Lab; the Kinetic Design Group and the Emergent Design Group at MIT's Department of Architecture; and the Smart Geometry Group, based in the UK.
His work has been exhibited at the 2004 Venice Architecture Biennale, the Digital Salon (2001), and gardenlab, Los Angeles (2004). He has lectured in Europe and the United States, and has published papers for conferences and in journals like ecaade, Acadia, IASS, and IJAC.
Kilian has co-taught workshops and a studio with the focus on generative programming and parametric tools in the context of fabrication at MIT since 1999. During his Ph.D. he was frequently thesis reader for MArch, SMArchS and MAS students. Since 2003 he has been co-teaching workshops with the Smart Geometry Group, in the UK, Canada and the U.S.
Axel Kilian was born in 1971 in Ulm, Germany, where he went to school before moving to Berlin to study architecture.

Method:
Experiments in design exploration

Themes :
Simulation Surface System Search

Key terms:
Constraint:
A condition where one entity enforces a state onto another

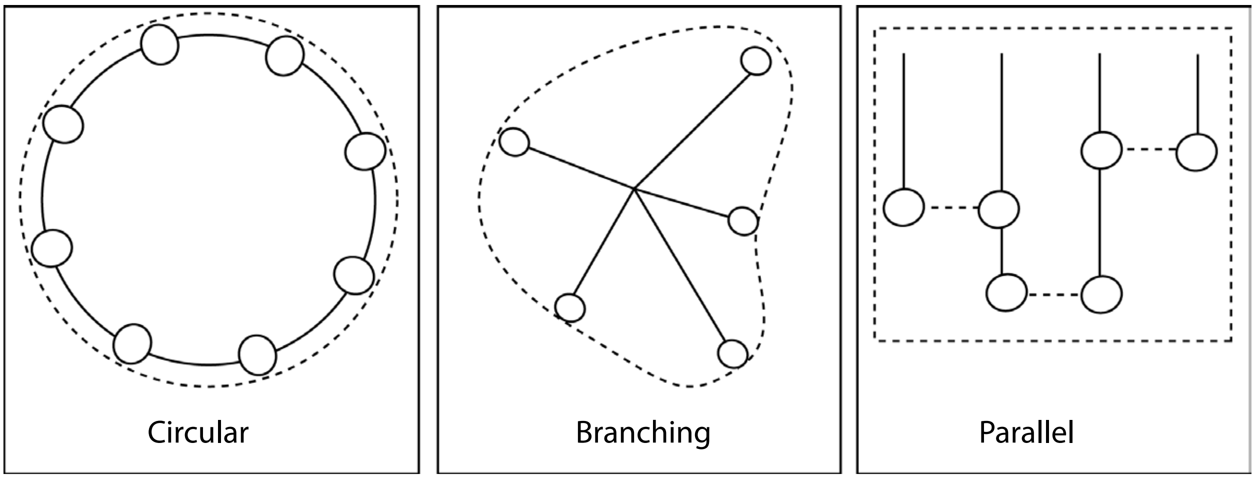
Design Driver:
A constraint with the most weight in a design exploration and therefore the most influence

Bidirectional Links:
A relation between two entities in which the role of driver and driven can switch

Design Exploration:
Variation of design solutions and goals to ultimately reach a better design solution

Design Explorer:
A physical or computational construct that combines design representations and constraints in order to support design exploration within the defined conditions

Types of Explorations:
Circular, Branching, and Parallel



Material constraints
Chair

Function constraints
Concept car

Topology constraints
Hanging model

The three different types of explorations tested in the experiments.

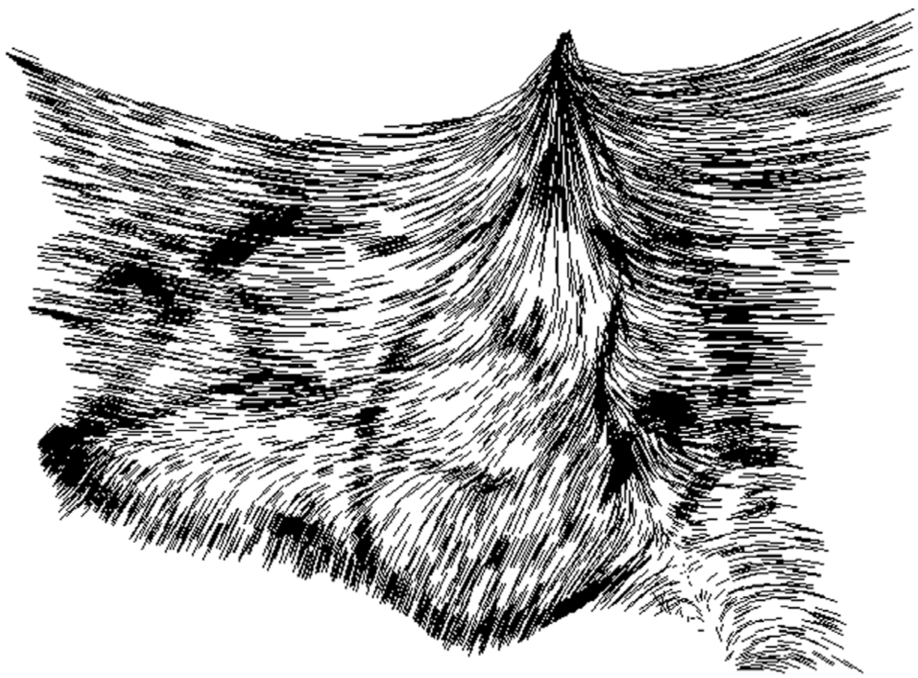
1 Introduction

This thesis explores the hypothesis that constraints can play the role of design drivers in design exploration. The thesis has developed out of a series of design experiments situated in a number of different design domains as a means to emphasize the general applicability of the approaches across design disciplines.

The author claims that significant design innovations occur in the translation between individual design representations, and that design exploration cannot be confined to a singular design representations. Furthermore, the author challenges the dominant role of geometry as the sole design representation at the center of digital design software. The specific experiments diagram the general design exploration from three separate domains and test the hypothesis from a design centered point of view. A key contribution of the thesis is the development of constraint diagrams for design explorations based on a network of constraints and design representations.

1.1 Design Exploration through Constraints

Design can be described as a process of emergence and discovery resulting from the definition of the constraints, their relationships,



A screenshot of an interactive vector cloud that responds to the point of attention of a viewer. Every action changes the overall and the local area, resultant in a fluid but not goal driven interaction that evokes emergent visual constructs.
Master of Science thesis (2000), Axel Kilian.

and the design problem. The constraints that form the boundaries of the problem can also serve as design drivers for possible design solutions. Understanding the constellation of constraints is crucial, and it goes hand in hand with the creation of design solutions. Every design move creates additional constraints and consequently triggers contextual responses.

1.1.1 Constraints as design drivers

Constraints are generally viewed as limiting factors in design. But there is evidence in research (Burrow and Woodbury 1999) (Krzysztof 2003) (Gross 1985) and architectural practice (Shelden 2002) (Schlaich 2005) that constraints can trigger the development of innovative design solutions and are a powerful way to drive solution space.

Constraints can help to focus design exploration through formulating the boundaries of available resources. There are various types of constraints that can be applied to a range of aspects of the design problem. While constraints may initially prove to be a limitation, over the course of the design process they can evolve to become a driver for innovative design solutions. The first step to aid this transition is to externalize the constraints.

1.1.2 Types of constraints

The main constraints used in the experiments are:

- Material: Constraints related to fabrication and material parameters
- Functional: Constraints related to different functional requirements in design definition
- Topologic: Constraints related to geometric variations without changing topologies
- Geometric: Constraints related to dimensional and relational aspects of geometric models

A design explorer is the assembly of constraints that apply to the design problem. The experiments all share such design explorers. The constraints are modeled both implicitly and explicitly. Constraint solvers allow for the explicit modeling of constraints, such as demonstrated in the form finding experiments of the hanging model. Implicit constraints are for instance, modeled through the choice of, geometric primitives that have the constraint embedded in their given properties.

1.1.3 Externalizing the constraints

Externalizing the constraints is a necessary step in constructing a design explorer. The analysis of the design problem through diagrams provides the dependency network between the constraints. In the next step of constructing the exploration framework, the constraints are translated into appropriate design representations. Finally, physical or digital implementation choices are made to complete the design explorer.

This network of representations goes beyond the clean homogenous digital geometric representation of today's CAD environments and can potentially integrate all design relevant forms of representation.

Antonio Gaudi's physical hanging models are examples of an externalized constraint explorer. The physical embodiment of the form-finding model ensures certain robustness as a simulation tool. At the same time, the physical nature of the models also limits the exploration due to time, cost, and conceptual restrictions. A digital hanging model, in contrast, extends the physical model by calculating structural member cross sections in a wireframe model based on forces present. It enriches the exploration by giving direct feedback.

As a central question this thesis asks whether design exploration can generally benefit from the cross referential interfacing of digital and physical design explorers.

1.2 The Experiments

The goal of the experiments conducted in this dissertation was to test computational methods for their potential to support different design exploration approaches related to constraints. The author combined and extended existing computational models from computer science, computer graphics and structural engineering in the experiments. In addition, the experiments test the integration of design drivers that traditionally have not been considered part of the design process, but are typically considered conventions of implementation. For instance languages have developed around common building materials such as brick or steel, that have, over time, become design conventions. Design conventions are powerful as long as they are applied in the context in which they were developed in. With the shifts in architectural forms and program many of the traditional building conventions need to be adjusted or even be reinvented partially to fit the new application context.

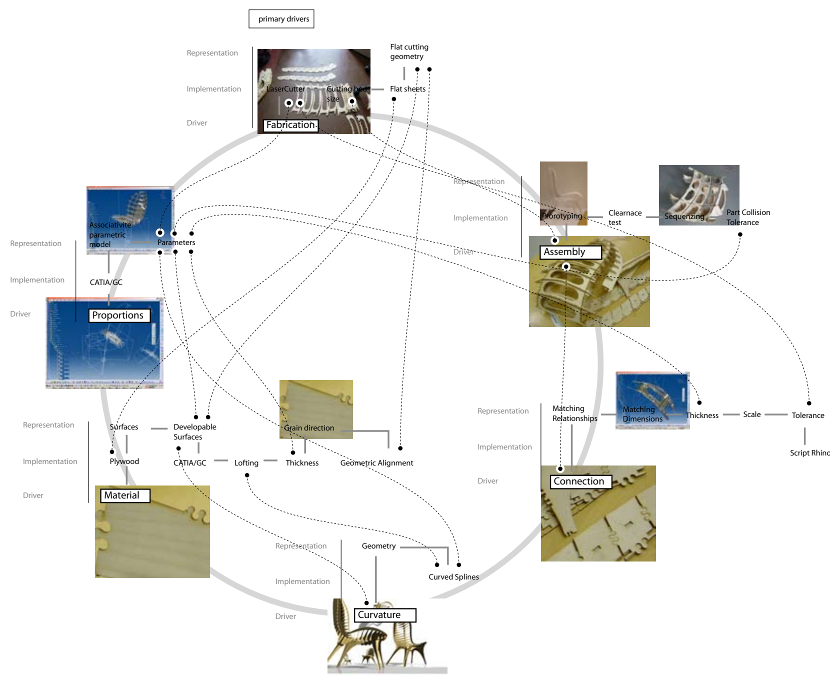
The experiments range a chair design integrating material and fabrication constraints to particle spring system for form finding to the physical prototyping of functional, constraint-driven designs.

The three major experiments in the thesis were:

- The chair experiments
- The concept car experiments
- The hanging model experiments

1.2.1 The chair – surface and material

The chair design experiment at the core of the circular exploration experiment section makes use of a parametric CAD model to integrate and mediate the influence from different design representations and constraints. The experiments sought to correlate material and geometry in curved, surface based forms, and identify how material constraints could be implemented



Circular dependencies in the chair design and fabrication study.

through geometry. The initial set of experiments, leading up to the chair design, served to gather knowledge of the constraints in material and double curvature surfaces. Those constraints were then combined in the main design experiment, which focused on the design of the chair, to refine the geometric design goal while conforming to the constraints of fabrication, curvature and part assemblies.

1.2.2 The concept car – function and design

The concept car experiment at the core of the second experiment section used a host of computational processes primarily focused on establishing the constraints necessary to define a novel car architecture referred to as the H-series. Among them were three dimensional constraint solvers, rigid body simulation models, parametric models and micro controller driven physical servo models.

Preliminary experiments involved functional constraint models and rule based explorations. Overall, the experiments used digital, conceptual and physical exploration techniques for the definition of a design task. The experiments were conducted as contribution to and in collaboration with, other members of the concept car design workshops and members of the smart cities group headed by William J. Mitchell.

1.2.3 The hanging model – form and forces

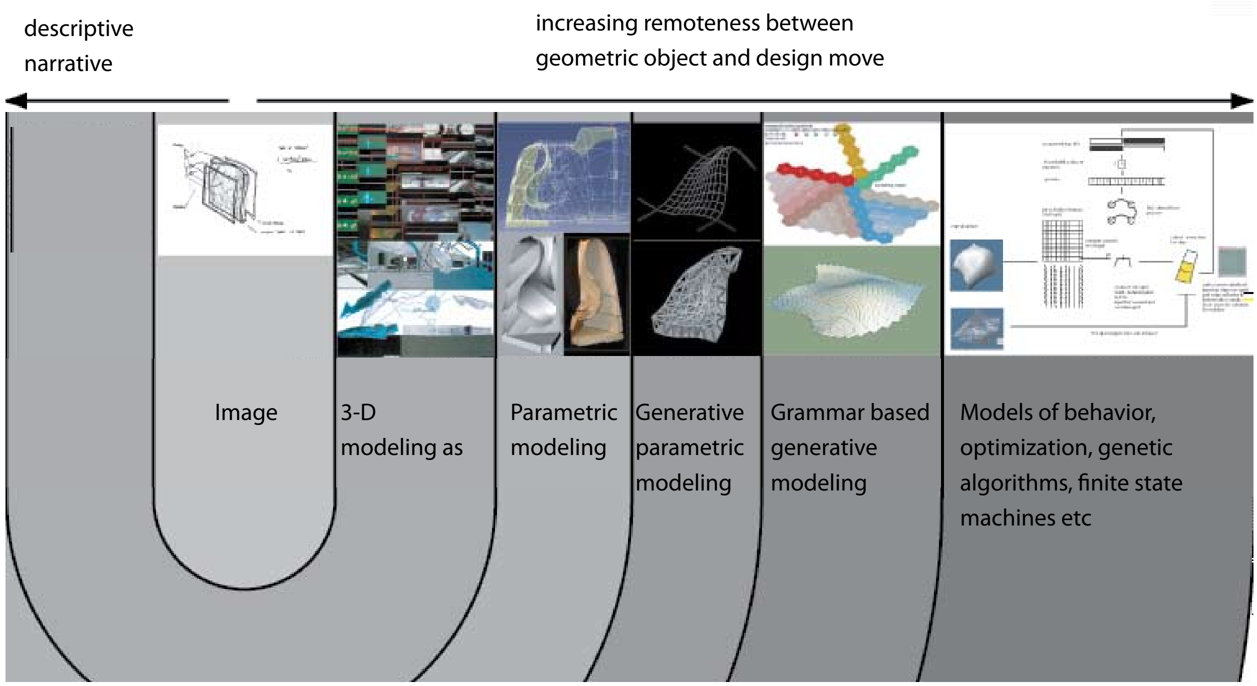
The hanging model experiment constitutes the core of the third experiment section which uses a particle spring system from computer graphics for implementing a digital version of a hanging model in force equilibrium. Preliminary experiments focused on the extension of the pure hanging model through the integration of the additional constraints of material properties. The geometry of a design emerges from a particle spring model that embodies the constraints of a physical hanging model. The particle spring models were initially developed with Megan Galbraighth and Dan Chak using a Particle Spring library by Simon Greenwold for a computer graphics class and further developed in a series of workshops co-instructed by John Ochsendorf, the author, Barbara Cutler, Simon Greenwold, Eric Demaine, and Marty Demaine.

1.3 Thesis Statement and Themes

Design exploration is a powerful process crossing design domains for solving highly constrained design problems, for refining existing design solutions, and for defining entirely new design problems. Modeling *design explorers* both digitally and physically can significantly help the design process and extend the range of solutions. The experiments in this thesis are exploited as a means to explicitly demonstrate the ways in which design innovation can happen at different points during exploration. It can be the innovation of a novel design from the interlinking of constraints. It can be innovation from the refinement of a design from known constraints, or the emergence of a novel design from exercising of a fully implemented constraint network.

1.3.1 Constraints in design exploration

The first key aspect of design exploration is the integration of constraints into the design explorer. A design explorer is defined by the translation of constraints into representations and



The diagram depicts the increasing remoteness between the geometric object and the architect’s design moves.

implementations. A design explorer is the assembly of constraints that apply to the design problem. Constraints become drivers of the design exploration process. Different domains are linked among each other. Every choice influences the overall exploration and possible designs. The mechanics of these cross dependencies between constraints, representations and implementations are best illustrated in detail in the chair experiment.

1.3.2 Bidirectionality in design exploration

The second key aspect of design exploration is the bidirectional nature of the computational models used. Most digital implementations for design evaluation are based on analytical principles that allow a one-directional link between one design representation and another. For instance, most rendering programs provide a way to analyze lighting conditions based on a given geometry, but only very few research projects (Schoeneman 1993) (Mahdavi 1997) allow the reversal of the process to derive a possible light source from a given light situation. This has to do with the non-deterministic nature of reverse mapping, as there are infinitely many possible solutions even for the simplest condition. If the relationship between driver and driven is more closely defined, however, it becomes possible to also obtain meaningful results from reversing the process. In the remainder of the thesis, this ability to reverse the link between design representations will be referred to as bidirectionality.

Bidirectional design exploration, then, is a domain independent part of the design process. It can be enhanced through computational and physical constructs, so-called design explorers. By adding bidirectional properties to the translation between different design representations, design exploration can support even complex constraint dependencies. As a result, novel designs can emerge by discovery.

1.3.3 Conceptual building blocks

The thesis introduces four major themes as the conceptual building blocks of digital design based on the historic and conceptual development of the field.

The four major themes of the thesis are:

- Simulation
- Surface
- System
- Search

The themes are developed in order to introduce both the vocabulary and the concepts that reoccur in the field of digital computation. Of course there are many more themes such as recursion and parameterization, yet most of these terms fall in some form or another within the categories of simulation or systems in general. The themes listed here are a reflection of the experience of the author in the appropriation of the field and its history and it is not intended to serve as an inclusive perspective of computation as a whole.

Excellent accounts of the history of the development of computational sub fields such as computer aided geometric design, computer graphics, and histories of modern computing currently exist that provide a more complete overview of the developments, (Farin 2001) (Ceruzzi 1998). The significance of the simulation, surface, system and search is their influence on each other and on the emergent field of computation, especially on its digital design implementations. Many of the frustrations of digital design can be traced back to choices and limitations in early digital design systems.

1.3.4 Choices of abstraction

The mind of an architect or designer is trained to be receptive to the contextual shifts and cross-dependencies that drive the creative process. Abstraction plays a crucial role in reducing complexity in a meaningful way so that it becomes possible for the designer to process and interact with the evolving problem. But abstraction can be counter productive in those design problems that rely on the interdependency of multiple design constraints. In fact, abstraction might compromise some particular features critical to design.

Where the constraint relationships rely on precise, computationally intensive dependencies, a digital implementation has advantages over an abstracted one. This is where digital models can make a difference in design exploration. They can help to overcome the complexity barrier in design problems that can only be explored further by taking into account all cross dependencies and that would suffer from a loss design relevant features through abstraction.

Computation can aid in the discovery of design solutions by modeling the constraints and the design representation into design explorers. Design explorers offer an alternative form of abstraction. Through the computationally based externalization of networks of dependency relations, design explorers capture and store states of the process for the designer to interact with. Through the interaction, additional layers of the constructed design model may be revealed. This externalized, computational representation offers a distinctly separate set of explorations aside from abstraction alone.

Once again, the hanging models can serve as an example. While the choice of abstracting a structure into point nodes and string connections is useful, the interaction of the entire node system relies on detailed computational simulations. Therefore, the mesh abstraction of the hanging model works as it allows for interactive design exploration through interaction with the nodes. In contrast, abstraction of the precise mesh form would render the result useless as it relies on the geometric detail to work structurally. The example is a well chosen abstraction for an interactive design explorer that offers the chance of discovery of design variation within a defined set of constraints.

In summary, abstraction choices are crucial in supporting design exploration and have to be carefully chosen to avoid established design conventions. In order to maintain the possibility for innovative design to occur it is important to create design abstraction from first principles and design specific for every design.

1.3.5 Incremental improvements versus innovation

Design constraints in architectural design are becoming more complex in terms of the overlay of functional, aesthetic, and performative demands. At the same time, traditionally evolved solutions that are localized and long-refined rarely exist for today's building projects. Even where precedents exist, the adaptation to new material and production techniques requires explorative techniques. With increased complexity, design problems can become over constrained, for instance in car or airplane design. There is a tendency to then resort to a strategy of incremental improvements. This is the result of the pressure of constraints from a large number of domains onto the design solution; introducing fundamental changes at the implementation level becomes virtually impossible. Here conceptual design exploration can make a difference. By developing a method for the generation of program descriptions driven by functional constraints, conceptual design exploration can help to resolve novel design approaches in highly constrained domains. In the thesis, the section on concept car design experiment (Chapter five) exemplifies this type of exploration.

1.3.6 Constructing expertise

In practice, knowledge of design constraints and how to work with them is referred to as expertise. It refers to a depth of knowledge that goes beyond convention. Expertise allows novel designs to emerge while drawing on previous experience. Experience is accumulated by humans or by organizations and teams, and gets formalized into procedures and eventually traditions.

Is it possible to externalize this expertise and capture it for design exploration? In practices such as Foster and Partners, expertise is turned into computational procedures. These tools are distributed to designers. The embedded expertise ensures, for instance, that certain geometrical requirements needed for fabrication are met. This sort of capturing expertise in applications or tools has a long history, spanning from sets of ship curves all the way to the first in-house CAD applications developed by architecture offices for their own use and eventually the curve and surface algorithms developed by Bezier and Casteljau (Bezier 1966) (Casteljau 1963) The problem is the lack of integration between these isolated instances of expertise. For exploration purposes, the isolated entities need to be able to connect and adapt quickly as the projects change course or new demands require different expertise. The disconnect between isolated applications is not a new one but reaches all the way back to the beginning of CAD systems. (Mitchell 1975)

Each of the experiments in this thesis connects areas of expertise through the design exploration in order to push the design forward. The integration of the findings is accomplished through a parametric model in the case of the chair, selective prototyping in the car experiment, and through a particle spring implementation that embodies the structural behavior in the case of the digital hanging model.

1.3.7 Design exploration in the context of computation

In "What is design?" Stiny gives an account of what constitutes design from the perspective of architectural design. According to Stiny, design is "an element in an n-ary relation among drawings, other kinds of descriptions, and correlative devices as needed" (Stiny 1990). Design rarely relies on a single type of representation or description but rather on a network of geometric and non-geometric representations. Stiny extended shape grammars by introducing such non geometric entities with Weights and label algebras (Stiny 1992). Other non-geometric generative rule systems, for instance color grammars, were developed for design exploration in parallel to geometry (Knight 1994). Constraints in the context of design exploration were addressed earlier in component based design explorers (Gross 1985), and more recently in design space exploration (Burrow et al. 1999).

Reinhold Martin refers to the role of architecture "…as a node in the communication network within and between disciplines and regimes of knowledge and production" (Martin 2003).and architects as being able to "…lay claim to a function within this web, relating to the spatial articulation of the network as such" (Martin 2003).. Spatial form is described as a response to an intangible network of knowledge and production, and the exploration of translations of such networks into built form as being a major question in architectural design in the 20th century. (Martin 2003).

In a different context, Norbert Wiener foresaw many critical developments related to the problem of exploration and prediction. Beginning with the control problem between humans and machines in anti-aircraft guns, observed an increasing merging of disciplines of mathematics, electrical engineering and others into a field he named "Cybernetics". Cybernetics is based on the Greek word for steering (Webster 2005). With this new term, Wiener refers to the challenge of balancing goals and feedback within the complex systems emerging in the human machine interaction at the time (Wiener 1948). The field of Cybernetics is similar to design exploration as the interaction between design goals and design exploration follow a similar pattern of feedback loops and face similar challenges in human-machine interaction. Essentially, design exploration could be described as steering the design using a "vehicle", the design explorer, through a solution space, where the target of the design coevolves with the explorer, the design and the design space itself.

Another inspiration for design exploration is D'Arcy Thompson, who in "On Growth and Form" laid the groundwork of relating forms found in nature to the diagram of forces that shape them (Thompson 1942). Nature is described as one big design explorer producing endless variations of species and forms from similar building blocks and generative principles. D'Arcy Thompson's model is fascinating in its descriptions of the emergence of beehives from the connection of material properties, bee behavior, and population density and probably more closely related to the work of Wolfram on cellular automata in a "New Kind of Science" (Wolfram 2002) than to Darwin's origin of species, as it explains form first and foremost as the result of generative rules on the level of chemistry and cell geometry, and only second as result of evolutionary selective pressures acting on what can be generated in nature.

This dependency emphasizes the importance of the exploration setup in relation to the possible outcomes, as selection can only dismiss what can initially be created. For instance, the complexity of a sea shell emerges from the intricacy of the growth based process, not from a top down design move. In a similar fashion, design explorers can aid in projecting the implications of constraints onto a design space. But as D'Arcy Thompson explains in great detail for the study of eggs, these processes do not rely on geometric relationships alone but on a range of shaping processes from pressure to material strength and the environment (Thompson 1942). Architecture responds to a far greater range of influences and the complexity can be overwhelming, both in the challenge of how to externalize concepts such as space or site and the translation of such concepts into design drivers. The right abstraction and translation of the core ideas is crucial and often the diagram of the design problem may not have a quantifiable implementation. However the externalization and mapping into a diagram that leads to a design explorer is a crucial step. Alexander summarizes the problem of design as: "What does make a design problem in the real world cases is that we are trying to make a diagram for forces whose field we do not understand." (Alexander 1964)

An expansion of design representation is necessary as geometry at the core of design exploration and generation may be insufficient as the vehicle. Geometry will not be substituted in design representation, but design exploration needs to move beyond the description of form. The power of diagrammatic design descriptions needs to be integrated into design exploration as a high level, a pre-formal design stage, as it offers the most flexibility in early design stages. This avenue is explored in the studies of writing devices in the experiment chapter.

The most immediate context for this thesis is, of course, the prevalence freeform building design in recent years, made possible by advances in digital design tools. This widening of the possibilities of built forms should be leveraged to respond to challenges posed by material constraints, building performance and aesthetics to create new formal and structural possibilities. In this sense, freeform buildings may end up representing a larger shift in design, less in what meets the eye and more in the general approach to design. The approach may be described as digital craft: as the integration of experience based parameters into a larger whole, serving as a design explorer. Eventually, the potential of freeform may satisfy many more constraints than the aesthetic one, as other constraints get folded into the exploration of form beyond the conventions of today's buildings. The results may even be similar in shape, but just as, for instance, clay shaped like a seashell may conform to the original geometrically, but it is likely to falls short on aspects of structure, form and aesthetics.

Design exploration can help to create form in response to and as the embodiment of constraints, contextualized in non-geometric aspects of design.

1.4 Precedent Work

The thesis builds on a number of precedents from architecture, engineering and computer science that use design explorers for cross-domain design. In particular, the thesis draws on the graphic static approach; the development of geometric constraint systems for construction; models from computer science such as Genetic Algorithms; and physical design explorers such as hanging models by Antonio Gaudi and Frei Otto. These approaches all allow defining and subsequently exercising a constraint network for the exploration of design variations with multiple competing goals.

To date, the precedents of digitally implemented design explorers are mostly based in engineering. This has partly to do with the quantifiable nature of engineering models, which makes them easier to implement in digital medium. In engineering and science, significant economic benefits are reaped from the development of analytical tools. The case is harder to make for design centered design explorers due to the lack of a clear theoretical basis for understanding the creative design process. Nonetheless, design centered design exploration constitutes an active research field involving cognitive science, computation, artificial intelligence and the design disciplines. The thesis contributes to this research through a series of experiments and their evaluation.

The precedents and how they relate to this thesis are explained in more detail in Chapter two.

1.5 Results of the Experiments

Three basic design exploration processes evolved from the design experiments in this thesis: a circular approach, a branching approach and a parallel approach.

- Circular — Refining the constraint relationships
- Branching — Establishing constraint relationships
- Parallel — Exercising the constraints

The categories emerged from the experiments, but reflect general trends in computation. They help to identify areas of study and missing links in the computational design exploration. They are also themes prevalent in design practice and education.

1.5.1 Refining constraint relationships – chair experiment

In the circular case, the main goal of the exploration is design refinement. Design refinement compares in many respects to the tuning of an instrument with the additional challenge of circular dependencies. This means that any change in one design representation may affect its neighboring representation. This can create a feedback loop with the changes rippling through the dependency chain.

A good small example of this codependency is the IPod© music player by apple©. If analyzed by features, there is little that distinguishes the design from its direct competitors. Its value goes beyond the actual product into the perception of the brand, as well as supporting services such as iTunes, the online music store. Still, as a design refinement challenge, it shows how the exact balancing of all features can make a difference. Those differences range from design choices to technical details, from materials to user interfaces, from pricing to marketing, a complex network of interdependent factors that need to be balanced to make a successful product. This is already a complex network for a digital music system, and it is much more complex for an architectural project. But the challenge remains similar no matter what scale the project. The task is to identify and prioritize the key design factors in a project.

Can a digital design explorer – modeling the different design factors using appropriate design representations and linking them through a constraint network – make a difference? The chair design experiment set out to test this question using a much less complex set of design features and their corresponding representations. The relevance of the experiment is less the end product of the chair itself. Rather, it is the insight gained into a tuning process that uses parametric models and digital fabrication in achieving a design goal that fulfills all requirements.

1.5.2 Establishing constraint relationships – car experiments

The most challenging design exploration is the one where the design constraints are not known at the start. At the beginning of an open ended design project, the role of digital design explorers is unclear. With the understanding of the design problem the design exploration evolves in parallel. The formation of the design problem creates many of the constraints from the emerging design features. This back and forth between design features and constraints may form and reformulate the design explorer many times. The car experiment illustrates this in the dissertation. Many of the design exercises aim at the reinvention of the architecture of the car without a complete programmatic definition at the starting point of the design. This might seem useless and certainly is not immediately goal driven. But in fact the openness of the approach allows for the emergence of design features that might become the building blocks of an emergent goal description. Therefore the exploration is aimed at the creation of the design problem itself.
What models exists to support such a process? There are rule based systems such as shape grammars that fulfill the requirement for an emergent design goal through the application of the rules that can co-evolve along with the exploration of the design.
. In order to overcome the conventions for car design, the exploration focused on alternative representation than form. Over the course of the experiments different sub component of the established car architecture were subjected to design variations in order to form the vocabulary for an emergent novel car architecture. The components studied by the author were ingress and egress, seating, chassis and skin. Some of those explorations developed

to complete vehicle designs such as the "H-Series" or "Athlete" in collaboration with other members of the smart cities group as listed in chapter three.

It became clear that there was no model that encompassed all the different exploration criteria or would adapt to the rapid changes in concept and object. But a number of digital models proved helpful in expanding the exploration where manual or physical based design processes would fail. They were: Rigid body libraries for driving simulations, parametric models in connection with three dimensional constraint solvers for the complex interdependencies of the articulated parts and digital fabrication for translating design ideas into physical prototypes.

The concept car experiments were the most complex and time intensive of the thesis. The design explorers developed are also the least well developed due to the complexity and open-endedness of the design exercise.

1.5.3 Exercising constraints – hanging model experiments

The digital hanging experiments suggest that if constraints are well understood and their relationships can be defined computationally, design exploration benefits greatly from interactive digital tools for exploration. In this case the exploration does not address the constraints, the design problem or the constellation of the constraints but purely the balance between the different design drivers. This shifting balance lets novel designs emerge within the boundaries of the exploration. This approach is particularly useful for complex interactions of constraints such as the hanging model where partial abstraction do not render any useful results. Here the articulated interactions make the difference between an innovative design and one that fails. The digital supported exploration makes it possible to keep track of the interactions and in addition translate the finding in secondary design features such as member dimensions of the hanging structure. The hanging model experiment is best described by the term "design by discovery". Design solutions emerge from carefully crafted design explorations, design focuses on steering the design constraints and monitor the developing results.

1.5.4 The experiments in overview

The experiments are heterogeneous and not limited to digital models but a mix of design representations both digital and physical. The claim is that digital models can help in integrating design constraints into design explorers. These design explorers can help to improve the design results and also more importantly help to access the unused potential of integrated digitally enhanced processes such as CNC fabrication. The indication is that the changes go beyond the increased efficiency or increased complexity of the product and may influence the design process itself. One change may be the departure from geometry at the center of digital design representation in favor of more integrated codependent design representations that might use geometry merely as an output

1.6 Introduction to the Chapters

1.6.1 Chapter two

In chapter two, the author reviews precedent work that has established the concept of design explorers with integrated constraints from the fields of structural design, architecture, artificial intelligence and computer science.

1.6.2 Chapter three

Chapter three discusses the conceptual building blocks underlying the thesis experiments: simulation, surface, system and search. Each concept is analyzed in its relevance to current design practice, and accompanied with visual examples. The dependencies between the concepts are explained with regard to the experiments. A brief history of the development of digital computation is developed following the occurrence of the four major themes of simulation, surface, system, and search.

1.6.3 Chapter four

Chapter four is an in depth study of the role of constraints in design exploration. It introduces the thesis design experiments and contextualizes them in the context of design research. Chapter four was published in parallel with the thesis in the International Journal of Architectural Computing and is reproduced her with permission of the publisher multi science, Brentworth, UK.

1.6.4 Chapter five

In chapter five, the author explains and evaluates the design experiments conducted in the course of the thesis in depth. The focus is on the three main experiments: the chair design (the circular exploration study); the concept car design (the branching exploration study); and the hanging model design (the parallel exploration study). In addition, the chapter contains teaching examples of each of the exploration models. Each of the main experiments is preceded by a number of preliminary experiments that establish the constraints of the domain. They are surface control and material constraint experiments for the chair, functional driver experiments for the car experiments and graphic statics experiments for the hanging model.

1.6.5 Chapter six

Chapter six concludes the thesis. The author discusses the role of constraints as design drivers in the light of the experiments. The author claims that it is necessary to go beyond the geometry-based, form-descriptive approach of today's digital design environments, and to choose instead a process oriented approach where design emerges from the modeling of the constraints and the analysis of the design problem.

1.7 Beyond Geometry

D'Arcy Thompson describes form as a diagram of forces (Thompson 1942), Christopher Alexander goes further in defining a design as the attempt to "make a diagram for forces whose fields we do not understand" (Alexander 1964). In this thesis, experiments were conducted to research the emergence of an understanding of the forces involved in design and design exploration and how design can benefit from it.

The focus of the experiments has been on the creation of innovative design, not on conforming to conventional design solutions in the experiments' respective design domains. Innovation is either in the idea (in the case of a novel design problem) or in finding a new constellations of established components and constraints (in the case of a known design problem), or in the emergent design

discovered from exercising a constraint network.
In contrast, the strategy of incremental improvements is firmly established in engineering and a powerful design method in the highly constrained design problems. Its disadvantages, however, are in the resistance to conceptual change due to the many dependencies.
With improved, integrated and digitally supported exploration methods for design, it will be possible to provide a competitive design approach that can stand its ground against the more conservative approach of incremental improvements.
By integrating the conceptual exploration more closely with the exploration of design implementation, as demonstrated in this thesis, expertise can be generated in parallel with the development of the design. At the same time, this expertise can be captured for future, open-ended design exploration should another design iteration be necessary.
This has implications not only for digital tools, but for the concept and formulation of design exploration in general. The thesis does only offer a glimpse at the range of design and many of core concepts of architectural design remain unaddressed. There is no claim for the universal applicability of the concepts presented. Rather a call for rethinking the use of computational tools and models as tools for design specific exploration.

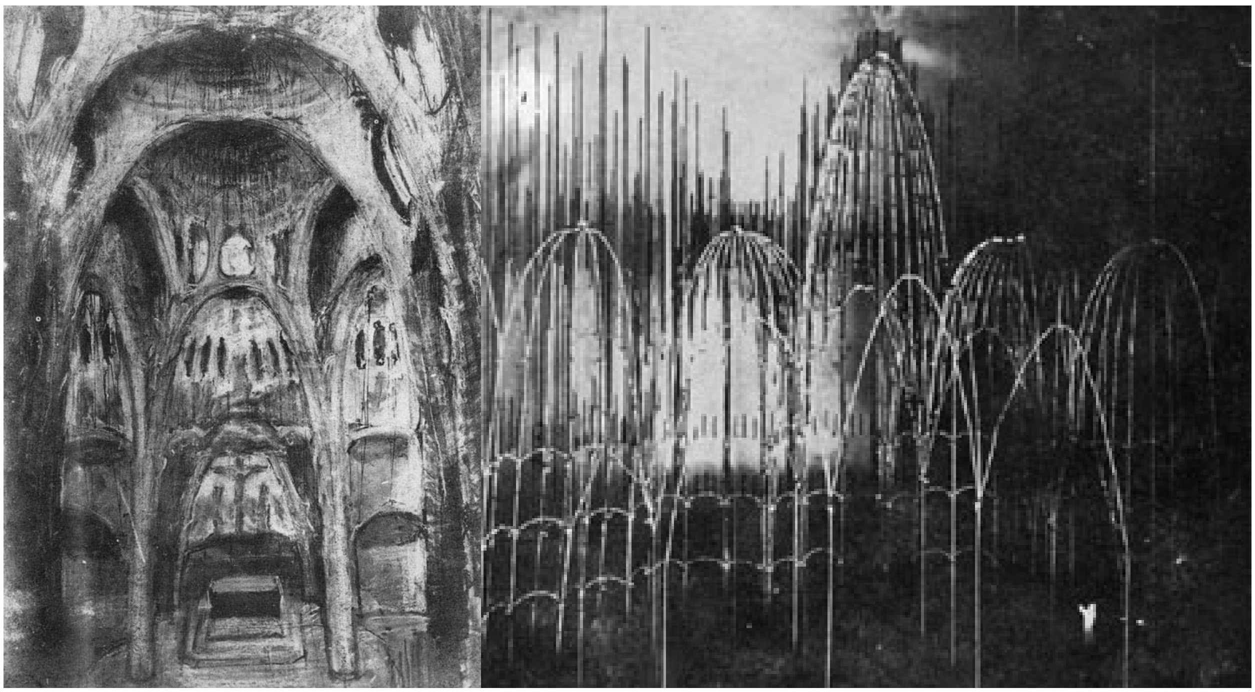


Fig 5: Gaudi' Colonia Guell hanging model on the right. On the left a photograph of the model with surfaces painted over the strings to capture the interior space.
Image (IL 34 Das Modell ‚Tomlow", J. et al)..

2 Precedents

Previous work on design explorers that integrate design specific constraints is focused mainly on two areas: physical modeling and computational modeling. Physical design explorers in the traditions of Antonio Gaudi and Frei Otto compute form based on forces present. Computational models do so using dynamic relaxation techniques or constraint solvers. Of related interest to the question of constraint-driven design explorers is a line of research in building technology that focuses on bidirectional constraint modeling in building performance such as daylight, thermal performance and energy use. Similarly, there is a large body of research in engineering focused on constraints in part assemblies. This includes the robust description of part geometries, taking into account the context of the parts in a parametric environment.

A thesis that addresses design exploration in the context of constraints must address the precedent work of architects using physical form finding. Usually this work is presented in the context of structural optimization. I would like to focus instead on its relevance as physical computation for integrative design exploration: the exploration of design not just in a formal sense, but rather as an integrative method with a design explorer modeled to respond to multiple constraints.

The precedents in engineering serve as examples of how to integrate analytical methods into generative tools for design exploration. The section on software architectures addresses possible new models for design specific software developments to replace the current monolithic CAD design software application.

The following sections address precedents of design explorers or design exploration processes in theses areas.

2.1 Precedents in Architecture

This section will discuss precedent work in architecture starting with Antonio Gaudi's hanging model in the context of design exploration. It will lay out material specific geometric design methods such as those employed in the offices of Frank O. Gehry and Partners or Foster and Partners. Christopher Alexander's Synthesis of Form is used to put the results of the thesis experiments into the larger context of design methodology.

2.1.1 Gaudi's physical based design exploration

The physical models of Antonio Gaudi in particular fit this notion of a design explorer as they address both structural and formal considerations and present the designer at all times with a status of the design that reflects the cumulative changes applied to the model. In addition to reflecting the intentions of the designer through multiple constraints, the hanging models are collaborative tools. They allow for a number of people to simultaneously view and interact with the design representation. All of these characteristics make hanging chain models a convincing example of a design explorer as developed in this thesis. Gaudi's work has received extensive attention in recent years due to the rise of freeform architecture and the associated frustration in uninformed shape making made possible by today's digital tools. Gaudi's form finding principles seem to offer an alternative to uncontextualized geometric sculpting in promising a rationale for freeform design. The hanging chain model reflects Gaudi's design intent in the form of an explorative device within the constraints of a physical object. Certainly, Gaudi would have pushed this device further and might have even deviated from the pure hanging model if there had been alternatives to the time consuming physical method of tuning the model. His renderings of the interior spaces, accomplished by painting on the wireframe photos of the hanging structure, provided him with surfaces to visualize the interior spaces more accurately, and are examples of his search for alternative design methods.

In addition, it is important to keep in mind that the models were subject to extensive changes due to translation from the zero volume string mesh to the material envelope of the built form. This translation is not a formalized conversion that offers a complete mapping of the string model to a model with the volumetric brick information contained. This step was based on experience and knowledge about the material and structure, but only partially represented in the string model in form of the weights. This is an important design step the translation of the abstract spatial diagram into a materialized volumetric structure.

Only for one project, the Colonia Guell, Gaudi was using the hanging model (Tomlow 1989) The Sagrada Familia, for instance, was not modeled using a hanging model, but instead by stacking plaster pieces. Here the relationships between structure, form and the building components are more complex. The columns' swept surfaces are constraint both by fabrication constraints and the overall structural skeleton governed by the force equilibrium. Fabrication constraints on the component level can potentially compete with global constraints governing the overall structure. In the case of the building block its mass in comparison to the overall structure is negligible, but it is indeed difficult to find the fitting swept surfaces for the components in all cases (Burry 2001).

The use of distinct design representation for the different design indicates on the one hand Gaudi's ability to experiment and expand the tools at his disposal but on the other hand also the necessity to develop design specific design explorers.

2.1.2 Christopher Alexander's Design Diagramming

Christopher Alexander's "On Synthesis of Form" (Alexander 1964) lays out a design methodology which, he stresses, is not a design methodology per se, but to be understood as an integral part of the process of making design. His argument for rationality in design and the need for formalism is to advance design beyond guessing. He argues that the big challenge in design is to "make a diagram for forces whose fields we do not understand". (Alexander 1964)]. He refers to the search for a design form as a process of finding a good fit or as a process where the solution emerges as those that do not fulfill the design goals are dismissed.

He contrasts self-conscious and unselfconscious processes of building, contrasting a tradition of craftsmanship reliant on incremental improvement yet unable to respond to unfamiliar tasks, with that of self-conscious design that produces forms of bad fit through the increasing distance between design process and execution, but capable to adapt to change.

Demands on design are complex and changes to existing solutions pose a highly complex challenge to the designer. Alexander identifies the shortcoming of diagramming in the design process based on concepts currently in use rather than on the analysis of the problem that is being investigated. He proposes a process of decomposing the problem into its subsystems in order to describe *a program* for the designer.

The thesis describes a similar process in the creation of function chains for design generation for the writing device study in chapter five.

2.1.3 Gehry's paper based design exploration

Developable surface in Gehry's architecture. It is clearly visible how the geometric constraint of straight lines of ruling is directly linked to the construction constraint.
Image: (Shelden 2002)

Gehry and Partners in general follow a sculpturally driven design process where the translation of form into buildable components is developed after establishing the form. This has been referred to a post rationalizing the design. An important design vehicle in the office is the use of paper at the sketch stage, the modeling stage and even at the construction stage in form of paper like materials like sheet metal. Dennis Shelden from Gehry Technologies developed computational tools for the Rhino modeling platform and for CATIA© that allow designers to model surfaces and be notified, when they leave the range of developable surfaces within the

Dennis Shelden developed a computational developable surface tool programmed in rhino that allows the user to explore a range of single curvature surfaces. A color warning is given if the developability constraint is violated.
One could describe this tool as a design explorer with embedded material constraints, which are enforced geometrically.
Image: (Shelden 2002)

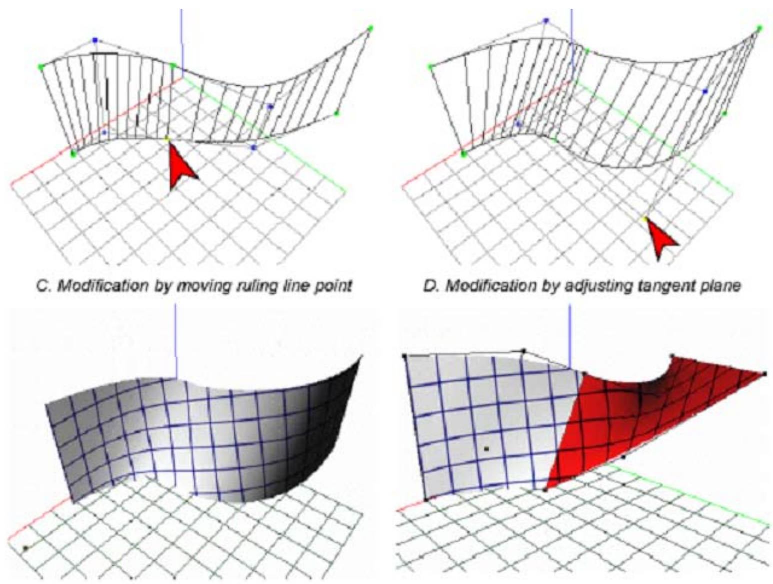
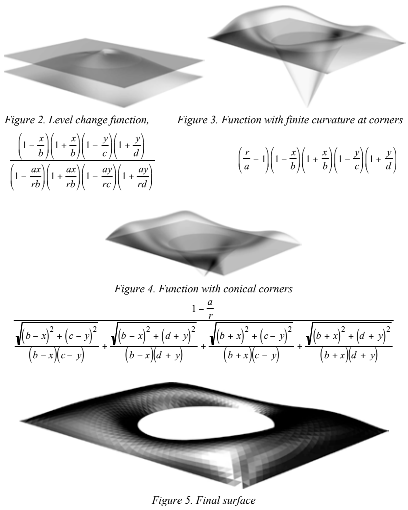


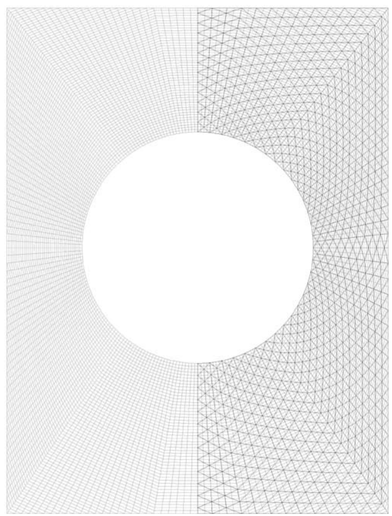
Figure VI-23: Developable surface approach based on (1,2) weighted Bézier patches

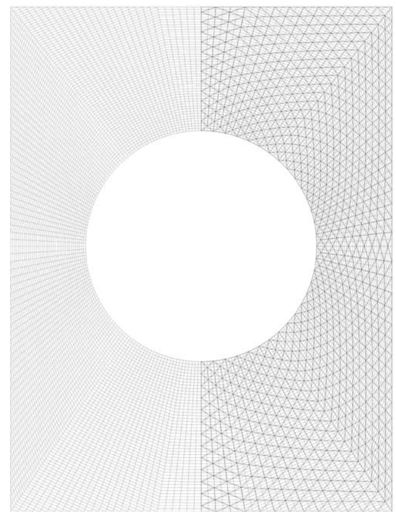
surface handling interface (Shelden 2002). This was made possible by taking material constraints, tested first in physical modeling, through paper and cardboard based models, and translating these into a digital representation. This gives the designer more flexibility in terms of dimensions and geometric interaction, as well as the ability to analyze the properties of the resultant surface. This is an example of how a constraint has been turned into a design driver through the use of a design explorer that embodies the constraint, in this case the developable surface tool. More generally, the use of paper as a design generator as well as an abstraction of form is a powerful instance of a shared design representation in a collaborative design environment.

2.1.4 Foster and Partner’s British Museum courtyard roof

Foster and Partners may be categorized to take the other end of the pre and post rationalized spectrum in comparison to Gehry. The design process in the office regularly works with building systems in mind or the parallel development of construction principle and design development and they are thereby pre-rationalizing the designs. The British Museum courtyard roof design goes further in the refinement of the geometry. It was designed by Foster and Partners together with structural consultant Chris Williams for Buro Happold. The roof is remarkable in achieving structural and architectural clarity within the highly constrained context of the inner courtyard of the British Museum in London. Not only do the spanning distances vary widely, but the placement of the inner cylinder is not centered in the courtyard. A further complication was being able to load the existing structure above the façade cornice only. This highly challenging and irregular context was resolved very elegantly with a seemingly floating triangulated grid shell without additional structural members or supports. The result was only possible in this clarity through the use of computational







Chris Williams from Bath University generated the above grid distributions for the British Museum courtyard roof. The structure is subject to multiple constraints: Structural, aesthetical, economical, fabrication and assembly related ones.
The position of each node was determined through dynamic relaxation. The starting condition is a discontinuous grid, the end condition exhibits continuity in the flow of lines across all parts of the roof, while keeping element lengths similar. The structural constraints are enforced through a design surface principle. The aesthetic appearance emerges from the dynamic relaxation distribution. This is an instance of a design explorer that relies only both optimization and geometric principles to simultaneously enforce constraints. Variations for explorations are easily possible by rerunning the program. Image: Chris Williams http://www.bath.ac.uk/~abscjkw/

techniques that combined the constraints of the geometry of structural shell geometry with the economy and aesthetic of self similar structural members. The technique used, and demonstrated by the consulting engineer Chris Williams in his presentations is one of dynamic relaxation of a initial simplistic straight line solution spanning from the courtyard edge to the rotunda. There are many abrupt transitions in the corner diagonals between members that would not be acceptable visually in such an exposed roof shell. The dynamic relaxation technique operates on the intersection nodes moving them incrementally around step by step measuring with the goal of minimizing the energy present in the system. Energy is represented by how long the members stretch between the nodes the longer the distance the higher the energy stored in this particular member. This will cause the nodes connected to that member to move closer and thereby reducing the energy. Eventually the grid system settles into minimal energy equilibrium. This state will exhibit the desired even distribution of length and smooth transitions even in the more complicated corners of the roof.

Image of the courtyard structure showing the even distribution and the visual continuity despite the irregular courtyard layout.

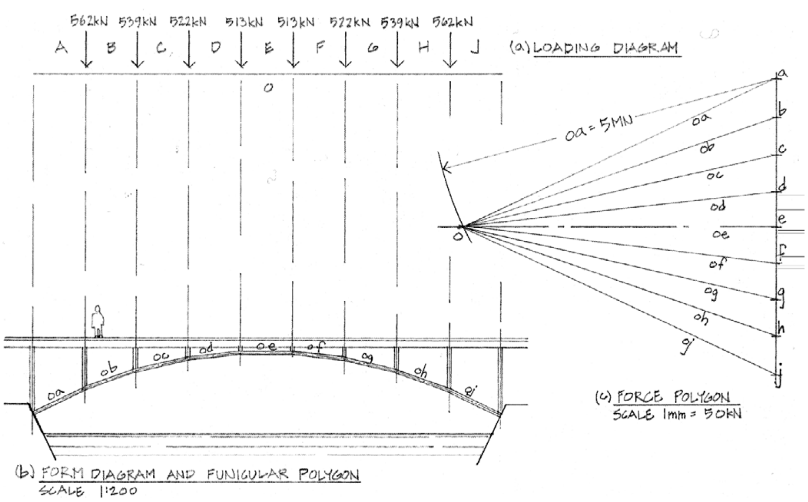
The principle of dynamic relaxation is related to the ordinary differential equation (ODE) solvers used by the author in the hanging chain modeler and the driving simulators. Dynamic relaxation techniques are simpler algorithmically but offer a wide range of application in engineering and as demonstrated here also in the optimization within the context of engineering and design constraints.

2.2 Precedents in Structural Design

This section will discuss structural design tools with a focus on exploration of form and forces. Starting from graphic statics and Active Statics it will touch upon Schlaich Bergermann's glass domes, Heinz Isler's shell structures, and topology optimization. These examples are relevant for the thesis as instances of performance based analytical tools. They support design exploration and are implemented in most cases to allow for bidirectional explorations back and forth between the different design domains of structure and form.

2.2.1 Graphic statics

One example of a bi-directional system that integrates multiple domains is graphic statics. The geometry of the force polygon is directly linked to the form diagram through simple geometric constraints. Graphic statics serves as a starting point for the integration of other domains besides form and structure, for instance fabrication procedures. This allows for an iterative design process in both structural design space and the geometric configuration of the design (Zalewski and Allen 1998)

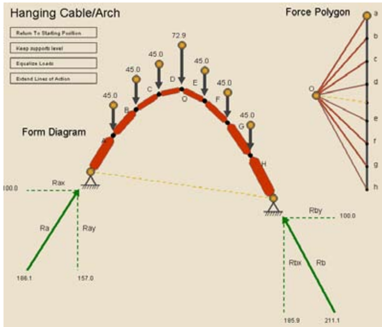


Graphic statics work sheet for a bridge design exercise. Form and force polygon are linked through geometric constraints such as that of being parallel.
The design can take place bidirectionally, either by driving the design from defining the forces or from specifying the starting geometry. Image: (Zalewski and Allen 1998)

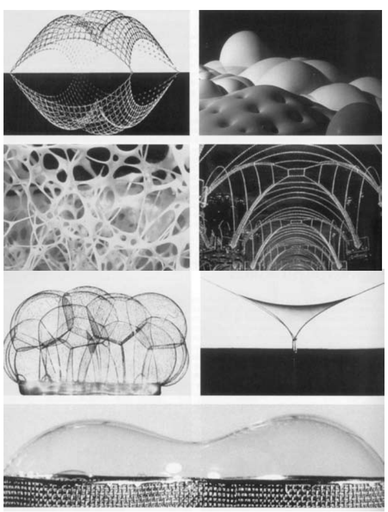
Graphic statics is an elegant structural analysis method that has gained some popularity recently after being the status quo in structural engineering around 1900. Its simplicity and graphic based calculation make it ideal for both manual and digital constraint based design development.

2.2.2 Active statics

Active Statics is a digital implementation of graphic statics developed by Simon Greenwold and Ed Allen (Greenwold and Allen 2005). This interactive version of Graphic Statics allows for a much better exploration of the different scenarios for a given load or the variation of the starting conditions than a hand drawn version. It has a high educational value and has been used widely in courses on structure at MIT and beyond. However there are some limitations. The user can not build new structures at run



Active statics - an interactive structure and form explorer by Simon Greenwold and Ed Allen. It allows to vary visually force and form polygon through a graphical interface. It is only partially bidirectional due to the lack of a solver architecture, but very educational through the immediate response to changes on either polygon.
Image: Screenshot Active Statics http://acg.media.mit.edu/people/simong/statics/data/index.html Simon Greenwold



The images show only a fraction of the physical based form finding methods developed by the Institute of Lightstructures at the University of Stuttgart under the guidance of Frei Otto.
Image: ILEK, Stuttgart. http://www.uni-stuttgart.de/ilek/Fotoarchiv/Fotoarchiv.html

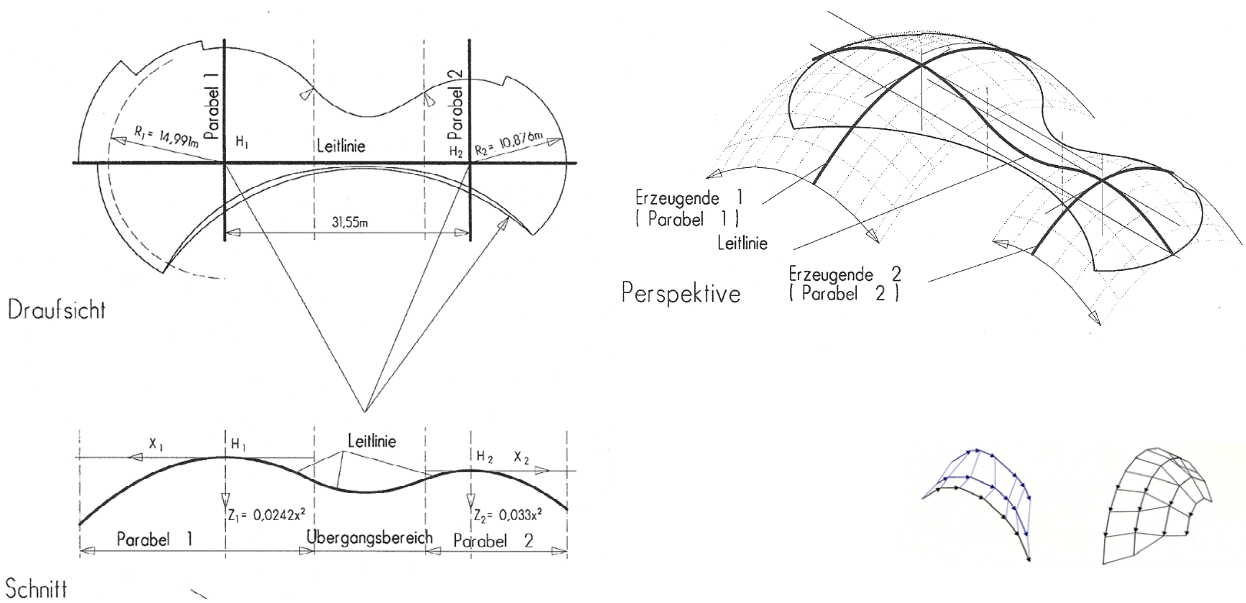
time. Another shortcoming is that it is not fully bidirectional, which means not every element in the implemented examples can be used to drive the rest of the graphic static assembly. This is a small detail but makes a significant computational difference. The Active Statics version relies on explicit calculations of all values through equations, whereas a fully bidirectional system would requires bidirectional solvers to find a solution with the given constraints. The digital hanging model project in the Chapter five Experiments allows both for user constructed structures at run time and uses a solver architecture that allows the interaction with any element in the design as a driver.

2.2.3 Frei Otto and the ILEK Institute, Stuttgart

The Institute of Light Weight Structures at the University of Stuttgart developed a number of physical forms finding techniques based on soap bubbles, hanging chain models and foam principles to aid in structural form finding. Most of these processes were pre-computational. They were based on the behavior of physical material that was then measured or photographed in order to scale the results to real structures. The Institute was founded by Frei Otto in 1964 (http://www.freiotto.com), and was next taken over by Jörg Schlaich in the 1990's and most recently in 2001 by Werner Sobek. Frei Otto's work and teaching initiated a world wide following in organic buildings that derive their structural form from the interaction of forces. A key aspect of his teaching is to stress that the geometry alone, even if derived through form finding by physical or computational means, does not constitute the solution. Rather, this geometry needs an expert to interpret and translate the results into details appropriate to material and scale in order to be a successful design. The work of ILEK is well documented in the literature. It will not be discussed in detail, but referenced as an early instance of research into design and the generation of form driven by the overlay of multiple constraints. Frei Otto reproduced had many of the physical building models at the IL where possible reproduced as digital models. The first digital model was created in 1966 by Klaus Linkwitz (Serebryakova 2006). Frei Otto does not believe teaching architectural design is actually possible beyond the foundations. He prefers the empirical approach: "I do not design, I search." (Serebryakova 2006). Otto's research, one could argue, constitutes a series of design explorers following his belief that design is a search for the shape of things.

2.2.4 Planar panel solutions for freeform surfaces

In the framework of their engineering practice, Jörg Schlaich and Hans Schober developed these principles of form finding as related to fabrication. This approach is discussed in the following

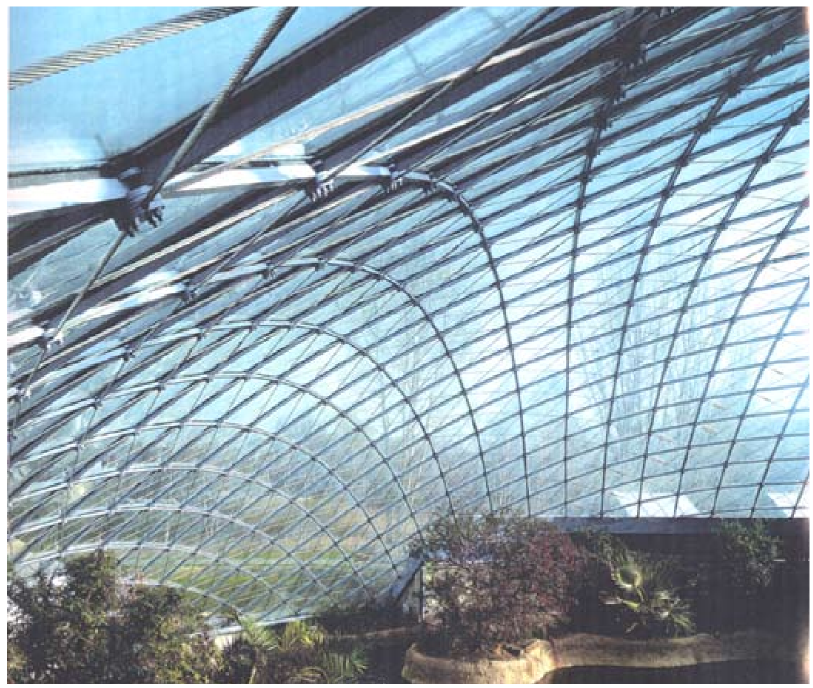


section. The use of translation surfaces allows for the fabrication of double curved surfaces with quadrangular, planar elements and constitutes a further geometric constraint modeling principle. It has been perfected by the engineering office of Schlaich Bergermann und Partner since the early 1990s on a number of freeform glass roofs. The challenge in the construction of freeform glass roofs is to keep the number of connections as small as possible, to avoid sharp angles in the glass pieces, to minimize material use during cutting, and to avoid stress fractures during handling, installation and use A robust fabrication constraint strategy is a necessity to allow for the geometric adjustments to achieve buckling resistance

Schlaich and Schober developed a series of triangular and quadrangular light weight glass roofs following a geometric constraint principle called translation surfaces.
The approach allows for free form surface approximation and robust quadrangular tiling solutions.
Image: (Schlaich and Schober 2005)

in the very thin glass shell domes. Translation surfaces procedure allows for the use of planar glazing and plane formwork elements, a major prerequisite for the economic realization of glazed, opaque or concrete spatial structures (Schober 2002).

The engineers' response to architects' freeform geometries was a geometric construction system that enforces the constraints while allowing variability within the constraints. While the geometric



The hippo house in Berlin by Schlaich Bergermann and Partner. The tensile grid interacts with the glass structure. The range of forms is not confined to spherical or toroidal shapes but the structural demands on the shell limits the range structural load bearing glass roofs.
Image: (Schlaich and Schober 2005)

principle of the translation surfaces is not a novel development in its own right, its application for freeform roof surfaces was.

2.2.5 Hans Isler's physical form finding

Another instance of the use of physical hanging models is by the Swiss engineer Hans Isler. He has derived an impressive set of light concrete shells from very precisely built and measured hanging models. His technique is one of externalizing a design idea in form of a model, which he then meticulously fine tunes and adapts before measuring it. The measurements are then scaled up and adopted to the load conditions as well as material and construction specific constraints. Isler does not stop when his shells are built, however. Rather, he continues to monitor and measure the structures after they have been completed. This is an important aspect of Isler's work, not only since he feels responsible for the safety of the structures, but also to provide feedback to earlier design assumptions and how they have played out in practice and over time. In a way, Isler is working with a circular dependencies network of designing, testing and building his structures. A shell design that works reliably, is elegant and minimal in material usage will, dimensionally, not be far away from a design that might fail, use excessive material or lack the refinement of Isler's shells. The type of design exploration Isler uses is focusing on fine tuning a proven design method to perfection, as even small deviations from a balanced design can affect the quality of the final result. In the thesis experiments this type of exploration is most closely followed in the chair example, where the set of constraints are known and the design goal as well and the exploration aims at finding a balance between them. The demands on architecture in terms of building performance increase making the sort of fine tuning of a design a very relevant challenge in architectural design exploration.

Isler: "[T]his is only the first step to finding the form. Afterwards one has to do the exact structural analysis, the modeling investigation, the layout of reinforcement and pre-stressing elements, and the support details. Also, the designer needs to think out the construction problems and finally to observe carefully the structure while it is in use." (Chilton 2000)

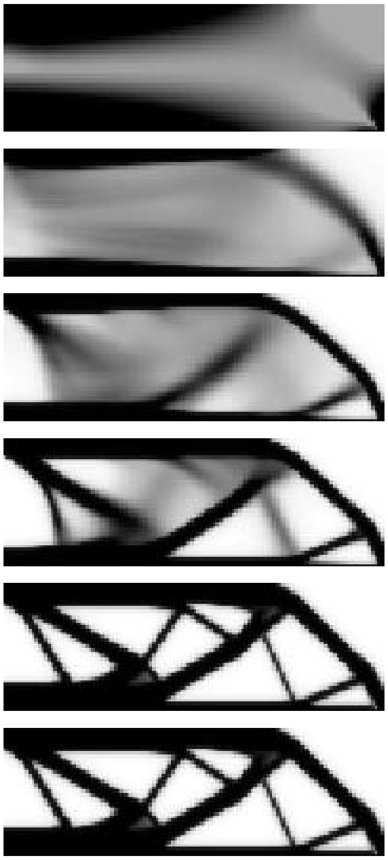
Since the 1950s, Swiss engineer Heinz Isler has pioneered shell structures with minimal thickness. The lines of thrust lie exactly within the thin shell cross-section. Isler extracts the geometry in a very tedious translation process from the physical scale model and scales it to full size.

Isler himself on a number of occasions came across what he thought were mistakes in the models. He commented that it was a sort of non-correctness in his ideas at first, a mistake. He was unhappy that this experiment did not succeed but finally he realized that it was giving him the solution for three problems that he had not thought of. (Chilton 2000)

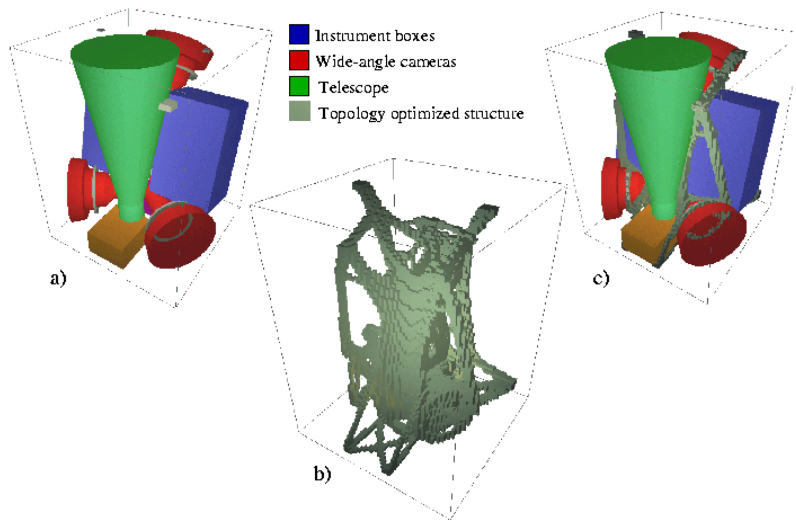
An image of a scaled model of a thin concrete shell by Heinz Isler from the exhibition "The Art of Swiss Engineering" 2005
Image: Heinz Isler
http://web.mit.edu/museum/exhibitions/galleries/swisslegacy/1.html

2.2.6 Topology optimization

Topology optimization is an example of a structurally driven



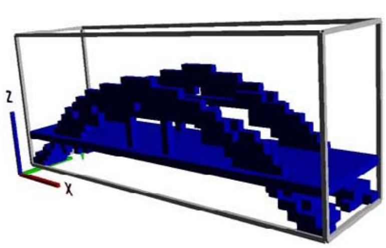
Sequence of a truss topology optimization over time from a set of forces. in two dimensions. (top)



Optimization of a satellite assembly: in three dimensions. (right)

design process that can be inversed: to create topology from forces present while taking into account constraints of limited area and material. One can also start by drawing the desired envelope of material and let the structure evolve within it. The leaders in the field of topology optimization are M.P. Bendsoe and O. Sigmund with their implementation TopOpt. (Sigmund 1994).

TopOpt focuses on optimized use of material within a designated design space in response to the presence of forces. The novel approach is the optimization of the structural topology in combination with the geometric dimensions of the structure. The approach makes use of Finite Element Modeling (FEM). But in contrast to conventional FEM analysis the FEM is used to generate a design geometry and topology. The authors caution that the process produces extreme solutions, meaning that the solution is not robust to slight adjustments of the starting conditions. Still, TopOpt is a very promising instance of the use of analytical engineering principles in a generative manner for design exploration.



Example of Topology Optimizer in three dimension
All images this page: www.topopt.dtu.dk by M.P. Bendsoe and O. Sigmund, TOPOPT.

2.3 Precedents in Artificial Intelligence

Constraint solving systems are common in artificial intelligence for robotics and expert systems. Constraint resolution is a common approach to modeling artificial intelligence. Many expert systems are based on logic using constraint resolution as in: "All birds are animals", "Some birds can fly", – "Is every animal that can fly a bird?" The answer is no, as the first statement would have to be "All animals are birds" in order to give a certain answer. The goal of this type of constraint resolution is to build up very large databases and gradually arrive at a notion of common sense within the system, which is essential for any intelligent system to operate and reason within the physical world and to interact with humans. While these constraint networks are very powerful, they are not very good at dealing with the sort of ambiguity common in design. However for functional descriptions of design and non geometric design generation logic solvers are important. The thesis does not contain any implementations of the functional explorations though beyond the diagramming studies of the writing devices.

Other models from Artificial Intelligence that were explored for this thesis in course work were Genetic Algorithms (GA) Genetic Programming (GP) and Artificial Chemistry (AC). GAs provide a means to search for design solutions from a defined solution set using constraints in the form of fitness functions. Although fitness functions are not technically constraints, the selection pressure over time enforces the constraints indirectly in the solution set. The GA examples are shown in the constraints for design exploration section. They were used for driving parametric models to achieve quantitative goals such as building volumes or geometric properties such as certain surface properties for fabrication.

Rodney Brooks, a strong advocate for embodied intelligence in AI over the construction of abstracted worlds in AI research warns in his paper "Intelligence without Representation" of abstraction. Many of the problems in AI seem to apply to computational design as well. He argues that the wrong kind of abstractions factor out all the aspects of perception and motor skills that he says are the hard problem to solve Brooks argues that the abstraction of a problem is the essence of intelligence and the hard part of the problem to be solved (Brooks 1991).

For design explorers this means that the definition of the constraints for a design problem to be explored is already a major part of the exploration itself and might in fact be the most challenging part. To construct a design explorer to explore possible design problems themselves will be even harder. The branching exploration discussed in chapter 5 for the concept car design falls into this category.

2.4 Software Architecture for Exploration

The thesis attempts to invert these approaches and develop computational and conceptual approaches that allow constraints to become design drivers in the exploration of a design problem. The premise is to model the constraints in a way that they can be exercised interactively.

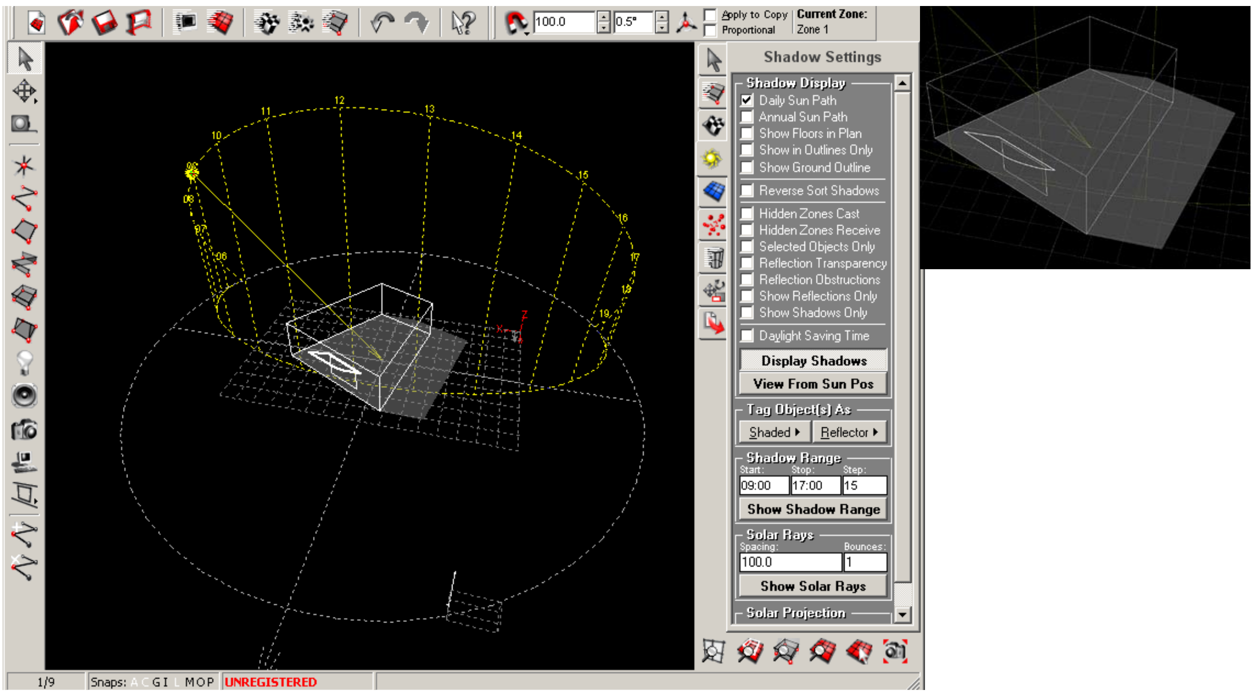
2.4.1 Analytical approaches in engineering software

Integrated engineering software packages offer parallel software components for different design representations and design domains. For instance, CATIA© a software package for air and space and well as automotive technology offers a Finite Element Modeling (FEM) package that can be directly linked to design geometry from the solid modeling module. This is a very powerful feature, which allows the iterative testing of designs both in terms of form factors and in terms of structural performance.

The shortcoming lies in the type of analysis. It is a one directional analytical translation of geometry into a force distribution. It does not provide the inverse approach of creating a structure for a set of forces.

2.4.2 Building performance evaluation - Ecotect

Ecotect provides performance analysis lighting, acoustics, and thermal performance evaluation with a focus on the integration with design. It is unique in its ease of use and accessibility. In addition it has examples where performance measures can be used in a generative principle. The software allows for the creation of simple design geometry directly within the analysis environment.



It also has features that allow the reversal of the design direction – essentially a bidirectional design process. For instance, it is possible to calculate the accumulative shadow created by a window shade over the course of a year, given a specific building site. In reverse it is also possible to generate such a shading device from the constraint to cast shadow on a specified window opening.

This reversal might seem trivial at first, but it illustrates a fundamental shift in the design process: the reversal of driver and driven in design exploration. The design problem is much better understood if cause and effect can be reversed. This is especially true in cases

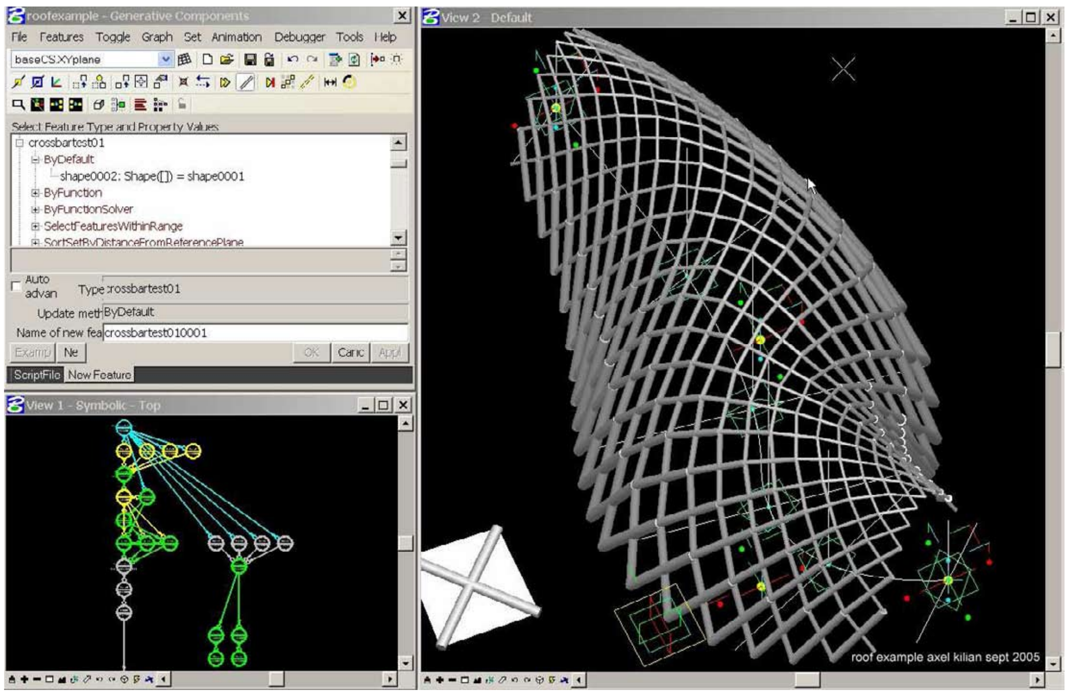
Ecotect by Andrew Marsh. The tool is mostly an analytical performance based tool for lighting, acoustics and energy analysis. But its simple interface and the ability to model and vary as well as script geometry gives it much bigger potential for design exploration. It is also possible to generate geometry from lighting conditions, such as a shading device that will keep a given window out of the sun year round. Design environment like ecotect that integrate analysis and generation are very promising developments toward integrated design explorers..

Image: Screenshot Ecotect

where the artifacts that are created defy our expectations, as it was the case for the oddly shaped year round shading device from the example above.

2.4.3 Parametric and associative parametric software

In recent years there has been an increasing interest and investment into parametric and associative parametric software in educational contexts and in professional practice. It is a marginal, but noticeable trend. Parametric software, riding on the success of the use of CATIA in the building projects of Frank O. Gehry and others, has the reputation of making otherwise impossible projects a reality. But the question remains: What enabled what? It was it the architectural imagination that made it possible to appropriate engineering software package for architectural use, despite all its shortcomings? Or was it the software development

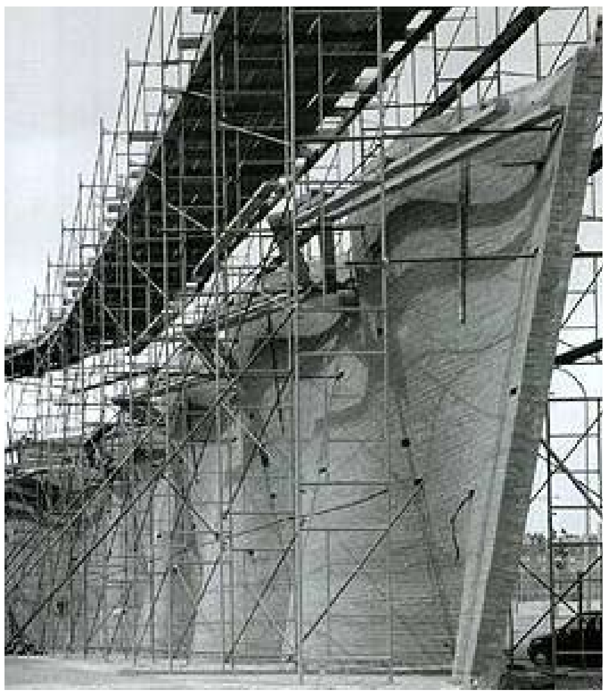


Generative Components developed by Robert Aish for Bentley. Generative Components is a unique variation of a parametric and associative modeling software. It allows for the extension of the tool set by the user from any of the four levels of interacting with the system:

- Manually modeling a component..
- Editing the topology and associativity.
- Scripting a geometry.
- Programming a component in C#.

This parallel approach allows to gradually introduce programming principles at different levels of abstraction of the task.
Image: Screenshot Generative Components by the author.

and engineering that went into the tool that brought the Gehry sketches into reality? It is not one or the other. The imagination has always played a driving role in the creation of new tools. In reverse, the availability of new processes and tools can inspire new ideas.
The parametric associative modeling of a design problem helps greatly in exploring dimensional and proportional variations. The integration of solvers for part assemblies and two dimensional



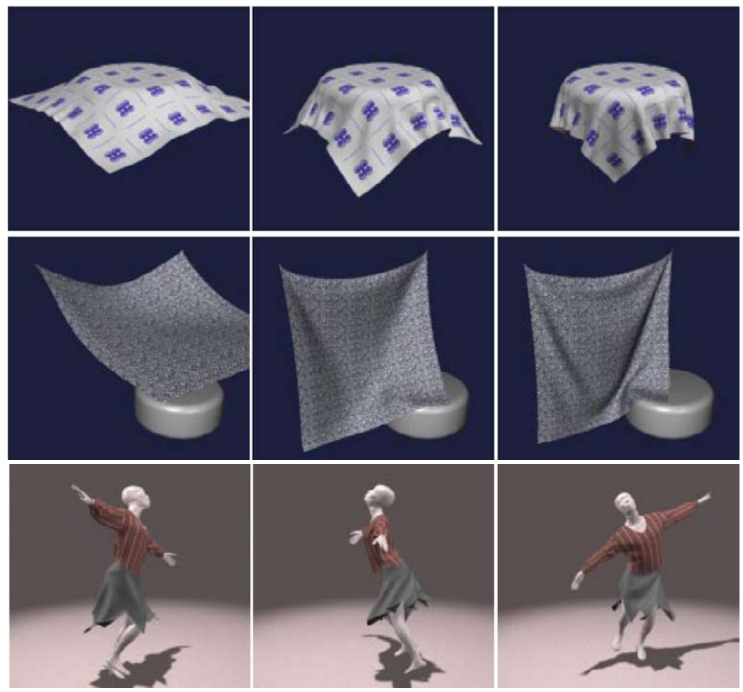
Eladio Dieste's church in Atlantido during construction. Dieste used simple construction principles to achieve his complex curved structural reinforced brick shells and walls.
The image shows the physical use of lines of ruling for the construction of a brick wall. This example is an interesting variation on the paper based surfaces of Gehry. Brick allows for non developable ruled surfaces to be built both share straight lines of ruling. Dieste literally built the control polygon of the surface physically.
Image: Eladio Dieste

constrained drawings offer some level of bidirectional exploration in the digital realm.

2.5 How other Methods Fail

The main problem in setting up a design exploration is capturing a design problem with its constraints and translating it into a design explorer. The goal of a design explorer is to aid in finding novel solutions or to make adjustments in existing approaches. The solution should fulfill all the constraints while negotiating the user's input.
For open-ended problem descriptions, creating and defining the design problem is the hardest part. In the case of the concept car design, the problem description initially is very broad, the

starting point being questions of mobility in the urban context. The solutions that developed over time were as much about defining the design task at hand as they were design solutions for it. If constraints don't exist or can be easily adjusted, the design problem becomes much harder to pin down. The thesis illustrates the approach taken by the author to defining the design problem and finding solutions to it, through selective prototyping: a way



Real time simulation of cloth. The improvement of solvers and the particle spring model for cloth simulation made real time realistic cloth simulations possible. Many other types of simulation can benefit from this fast and robust solver system. The hanging model uses a particle spring system library written by Simon Greenwold that is based on cloth simulation precedent.
Image: (Baraff and Witkin 1998.) Digital reprint ©1998 by CMU. The original printed paper is ©1998 by the ACM.

of capturing subsets of constraints to test their interplay and the emergent design solutions.

Grammatical approaches, which were initially studied, fail in the context of this open design problem due to their reliance on generative rules. These are best derived from a set of existing design solutions. For the concept car studio, the design problems were so far removed from existing solutions, both in vehicle architecture and design response, that a rule based approach drawing from precedent designs did not prove to be useful.

Physical design explorers in the realm of form finding and structural evaluation are very powerful collaborative and interactive design explorers. They allow for tangible spatial interaction with the design that allows people to work on it simultaneously. However, their biggest strength, their physical presence, is also their biggest disadvantage. Their physical presence and size makes them very difficult to construct and maintain, let alone to share them across larger distances.

Designing with constraints poses a challenge of representing both the design problem and the constraint influence accurately enough to obtain meaningful results, but not too specifically to prevent any evolution of the design.

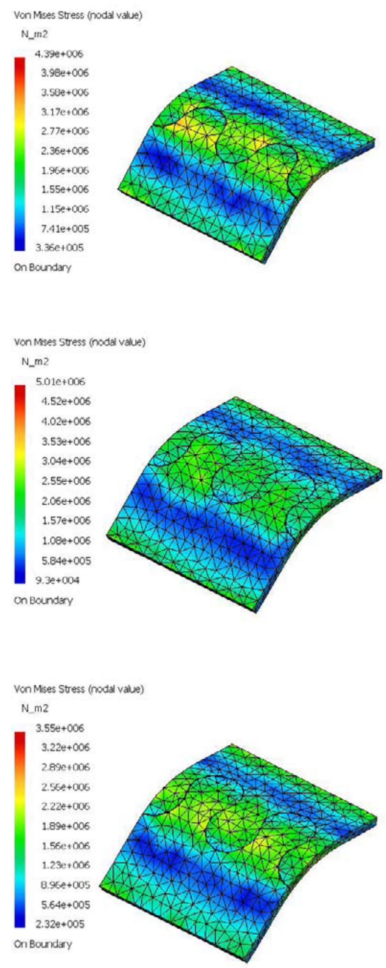
Modeling material constraints in are a good example for this dilemma. On the one hand, the structural and material resistance could be precisely modeled through a Finite Element Model of the volumetric object. On the other hand, one could choose a different material representation that is design relevant rather than realistic such as developability (Shelden 2002). The two approaches could be called holistic modeling and selective or subjective modeling respectively. Both are simplified, computational models of a much more complex physical artifact, one aims at simulation, and the other focuses on a design-specific representation.

Creating a design representation has to go beyond a simulation of conventions and focus on the aspects that could lead to an innovative use of the material. It is the difference between an analytical and a generative approach to design.

The thesis argues for a generative approach to design exploration. This means that alternatives to purely analytical simulation have to be found to support the generative aspects in design exploration.

2.5.1 Limitations of parametric associative modeling

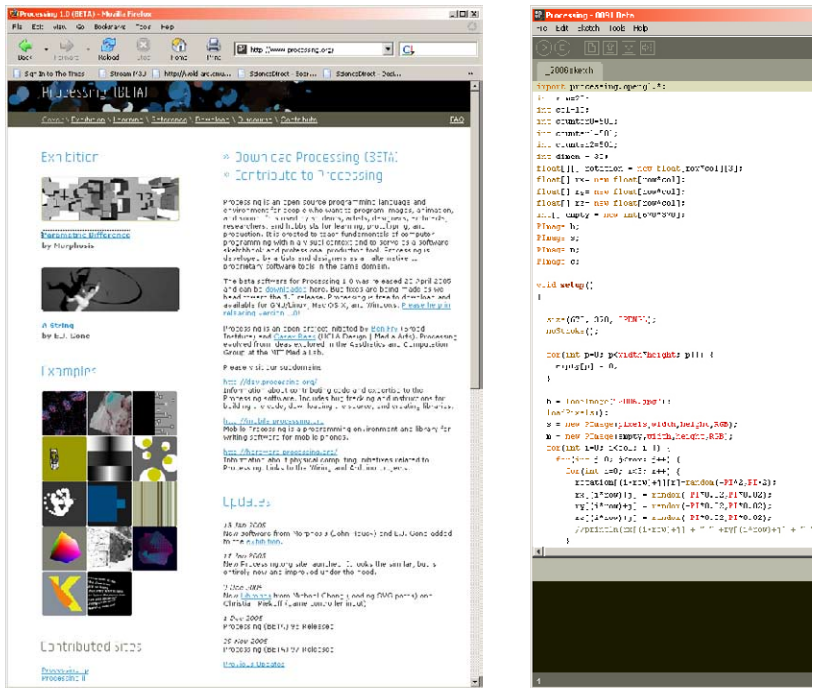
It is safe to say that the current set of parametric and parametric associative tools suffers from limitations in the definition of design. One of those limitations is the hierarchical structure of the dependency chains in basic parametric definitions. These require structuring the design approach early on in the design process and offer little flexibility once a model has been created. Associative parametric models represent design through parametric geometry. Although multiple ways may exist to describe any particular state of design geometry, translations to alternative ways of seeing and describing the design geometry is cumbersome and limited. The parametric redescriptions cannot keep up with the rapid changes in



Finite Element Analysis evaluation of different geometric variants of the puzzle joint by the author.

design interpretations by the designer. This withholds the benefits of parametric modeling from the early brainstorming sessions in design. In the later phases, it tends to prematurely freeze design due to the investments made into the parametric associative design description, rather than to support exploration at an elevated level of detail through an exercisable model.

Example of an inversal of the typical analytical lighting problem. The after a desired lightening effect is marked on the floor the program approximates the lighting conditions necessary to achieve the desired effect.
Image: (Schoeneman et al 1993)"Painting with Light", copyright ACM, copied with permission of the

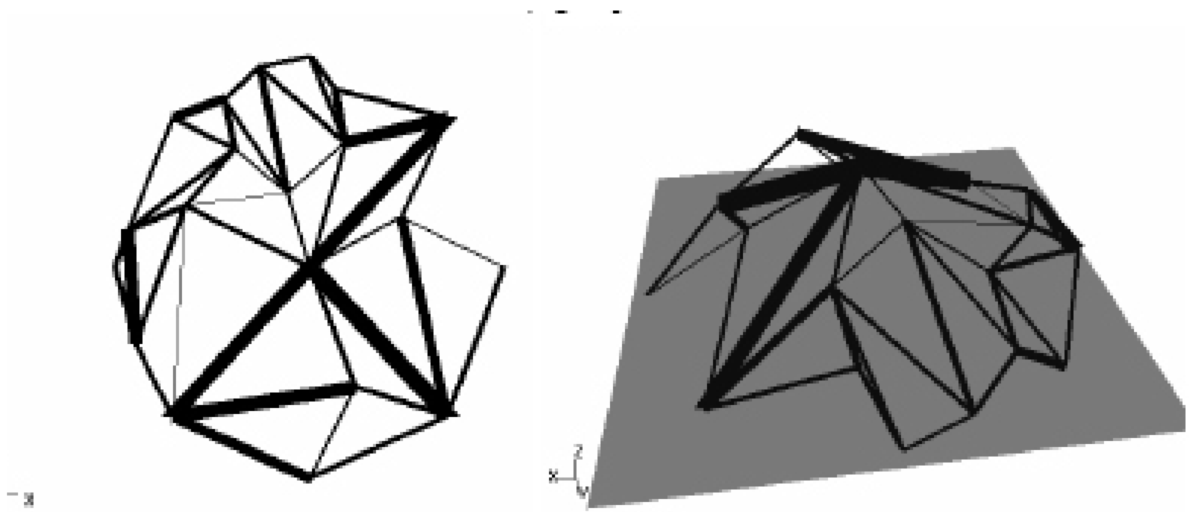


Alternatives to closed software environments are community based design platforms such as Ben Fry's and Casey Reas' processing environment. With a user community of several thousand designers it has grown into a true exchange platform of computational design. The platform allows to capture and share computational ideas in JAVA through a series of elegant methods that lower the bar for beginners and make higher functionality easier accessible to expert users.
Image and program:
(Fry and Reas 2006)

2.5.2 Limitations of Finite Element Modeling (FEM)

FEM is a very powerful structural and dynamic load analysis method that allows for a wide range of applications in a number of domains ranging from structural analysis to frequency analysis in electronics.

The core limitation of FEM methods for design exploration is their one-directional, analytical nature. The result of a FEM analysis of an architectural structure offers little in terms of how to change the geometry to overcome any problems detected in the analysis. This makes the approach of little use for an exploration based design approach, unless it is combined with an optimization technique that runs through a large number of design iterations using a fitness function to measure the progress. A promising extension of the use of FEM models is topology optimization. By integrating constraints into the FEM calculations, it is possible to iterate back and forth between a desired topology, structural optimization and material distribution within the specified design envelope. Another successful example of the use of FEM in a generative, rule based system is the EifForm application developed by Kristina Shea (Shea

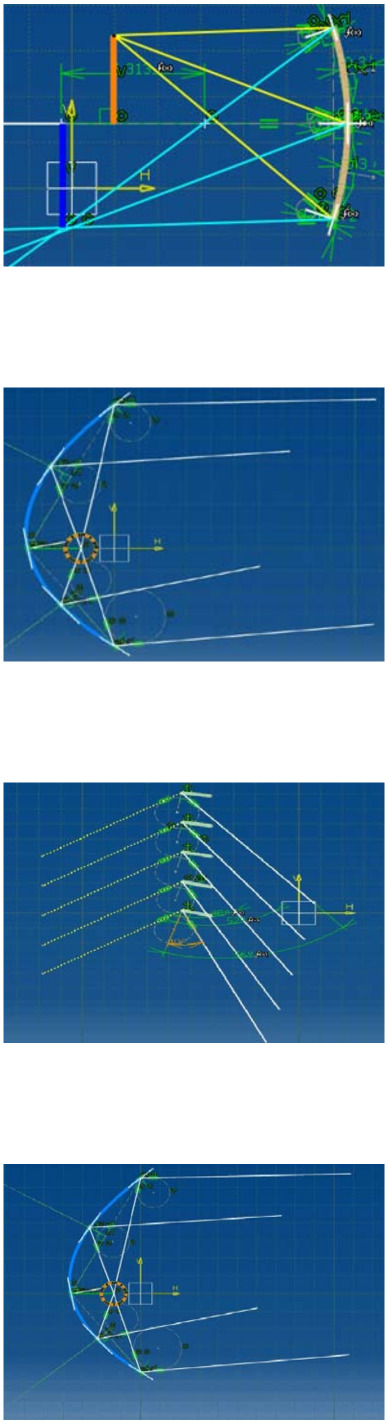


1996).

Eifform by Kristina Shea. From Essays of Discrete Structures: Purposeful Design of Grammatical Structures by Directed Stochastic Search', Ph.D. Dissertation, Kristina Shea, Carnegie Mellon University, Pittsburgh, PA (USA). 1997.
The combination of search and ruled based generative techniques and analytical methods is remarkable as an integrative approach.
Image: (Shea 1997)

2.5.3 Limitations of interface based digital design tools

Each computer aided design program relies on a specific set of core geometries. Accordingly, the designs produced with a specific program share a similar aesthetic. Diversions from the easily accessible geometries are possible, but this requires a conscious design intent and considerable effort on the user's part. In addition,



Series of geometric design studies for modeling lines of reflection based on the bidirectional solver environment in the two dimensional sketching environment of CATIA©.
Image: Light ray studies by the author.

software packages tend to create biotopes of users fueled by influential teachers and practices. There is certainly a feedback loop between user community and software features, most observable in the surface use of MAYA, or the parametric façade objects of ArchiCad in the 1990's in Berlin.

Of course, it is essential to provide tools and interfaces to make complex geometric entities approachable to novice users, as it is the case with NURBS surfaces in most current modeling environments. The choice of interfaces defines the reach of the average user, and makes similar procedures available to a very large group of users simultaneously with their qualities and their shortcomings. Other similarly approachable interfaces to interact with the geometry may not make it on the toolbar therefore the interface choices distort the exposure of users to the computational geometry universe. This is unavoidable with any method, but nonetheless often missed in teaching these tools. Recent developments of increasingly exposing the inner workings of software to users through plug-in and scripting environments level the playing field somewhat. While programming libraries and languages suffer from similar effects of exposing certain concepts and hiding others, they certainly open up a wider designer defined toolbox, given some preliminary knowledge of programming is taught.

Eventually the conceptual limitation of scripting and programming environments of software are too limiting for the bigger challenges of design exploration as well. A partial reinvention of software tools is necessary. The monolithic software giants of today are suffocating from their own legacy, the relative inertia of their user base aging together with the software platform, and the protocols and procedures that are resistant to change.

2.6 Conclusion

In conclusion of the precedents the author proposes an alternative approach to design software that focuses on an on the fly assembly of component, functionality, computational and conceptual models around a design task at hand. Such software architecture would rely most and foremost on interfaces and links between heterogeneous modules. Such an approach would face many challenges of supporting mappings between components that never worked together before. But if software would evolve parallel to the understanding of a design problem, if the exploration of a novel design would be mirrored in the creation of a novel design environment, it could be a far more design process oriented interaction with the digital medium than in today's skill based, commercial territory based software landscape. This process needs to be carried by many, can hardly be forced upon the industry by a few, and might find its precedents in the open source movement. But the hard question about interoperability and core design representations in such an open and evolving system are not answered by this comparison. The software industry gets routinely stuck in its attempts to define the next industry standard for interoperable, three dimensional, building information modeling, parametric, file standard. But maybe this approach is exactly the reverse of how it should be. Maybe the design projects should be the programs, developed by the designers to carry the complexity and to specify the problems to be solved and addressed by the vendors in an open source market place of design challenges triggered by design problems. The software should evolve around the design task, not the other way around, this could even involve the people contributing to it, in a sort of free design agent environment where design problems trigger the assembly of teams of people and computational models and expertise on the fly and temporary, maybe even in a sort of eBay© like based market place. The analogy is not targeted at the business model so much as at the posting of challenges. Why should talented developers all be tied up in competing parallel software organization essentially producing very similar products averaging their products based on perceived market needs rather then as free agents joining selectively the development of specific novel computational challenges in micro teams digitally negotiated and possibly remotely? Such a field maybe initially chaotic but over time may develop a far more adaptive library of approaches and methods constantly evolving with the design world then monolithic conservative software players. IN micro steps this is already happening in the academic environment with students developing their project specific tools from a platform of core software and languages, initiatives like digital tooling tried to act as a gathering place for such efforts and in moderating and connecting interested parties.

3 Constraints for Design Exploration

In this thesis it is described how constraint modeling can support design innovation. Furthermore, it is laid out how constraints are employed in the construction and exploration of a model's design space. The approach is placed within the context of design exploration using computational and conceptual representations of design. A review of the literature reveals that geometric, topologic, functional, and quantitative constraints are those most commonly used. For each constraint type, an example is presented drawing from several workshops and research conducted by the author. The examples range from product design, to structural design, to fabrication issues in freeform geometry. Based on the case studies, it is described how the different types of constraints can be used as design drivers and help in the exploration of solution spaces. In conclusion, the need for bidirectional exercising of constraints as the next challenge in design exploration is identified and its relevance for cross domain design is discussed.

3.1 Constraints in the Context of Computational Modeling

Constraints are generally viewed as limiting factors in design.

But there is evidence in research (Burrow and Woodbury 1999) (Kryzysztof2003) (Gross 1985) and architectural practice (Shelden 2002) (Schlaich and Schober 2005) that constraints can trigger the development of innovative design solutions and are a powerful way to drive solution space.

Constraints can help to focus design exploration and to work within the boundaries of available resources. There are different types of constraints and they can be applied to a range of aspects of the design problem. It is suggested that initially, a constraint may prove to be a limitation, but over the course of a design process it may evolve to become a driver for innovative design solutions to the problem. For design exploration, models that are not over-constrained are necessary. How, then, can a constraint be modeled in a favorable way for design exploration? The type of constraint is less important than how it is modeled. For design exploration, models that are not over constrained are necessary.

To start, a number of core types of constraints and the models that can use them can be identified. In general, four types of constraints can be identified from the literature (Krzysztof 2003) (Gross 1985) relevant to the examples presented here;

- Functional constraints; Requirements for what a design solution must accomplish functionally.
- Topologic constraints; Relationships between entities that form a topology.
- Geometric constraints; Geometric dimensions as well as relationships.
- Quantitative constraints; Quantifiable measures such as volume, thickness or length.

An example is given for each of the constraint types in research conducted by the author.

First, an example is presented where functional constraints are used as design drivers in a product design study looking at writing devices. The particular study was conducted in a concept car design studio headed by William J. Mitchell in 2004 at the MIT Media Lab. Second, an example of topologic constraints in a structural form-finding implementation is presented, based on force equilibrium using particles and spring systems. This model enforces axial forces in structural members for a given topology and it was developed in a workshop co-taught by John Ochsendorf, Axel Kilian, Barbara Cutler, Eric Demaine, Marty Demaine, and Simon Greenwold in 2004 at MIT. Third, an example of geometric constraints is presented. This research study was conducted by the author for enforcing developable surface properties for the fabrication of double-curved surface approximations. Finally, an example of quantitative constraints is presented. Two projects are shown, both based on searching the solution space using genetic algorithms and three dimensional constraint solvers to match parametric models to a target value. One deals with an eight-degrees-of-freedom articulated car design, the other with a family of vase designs.

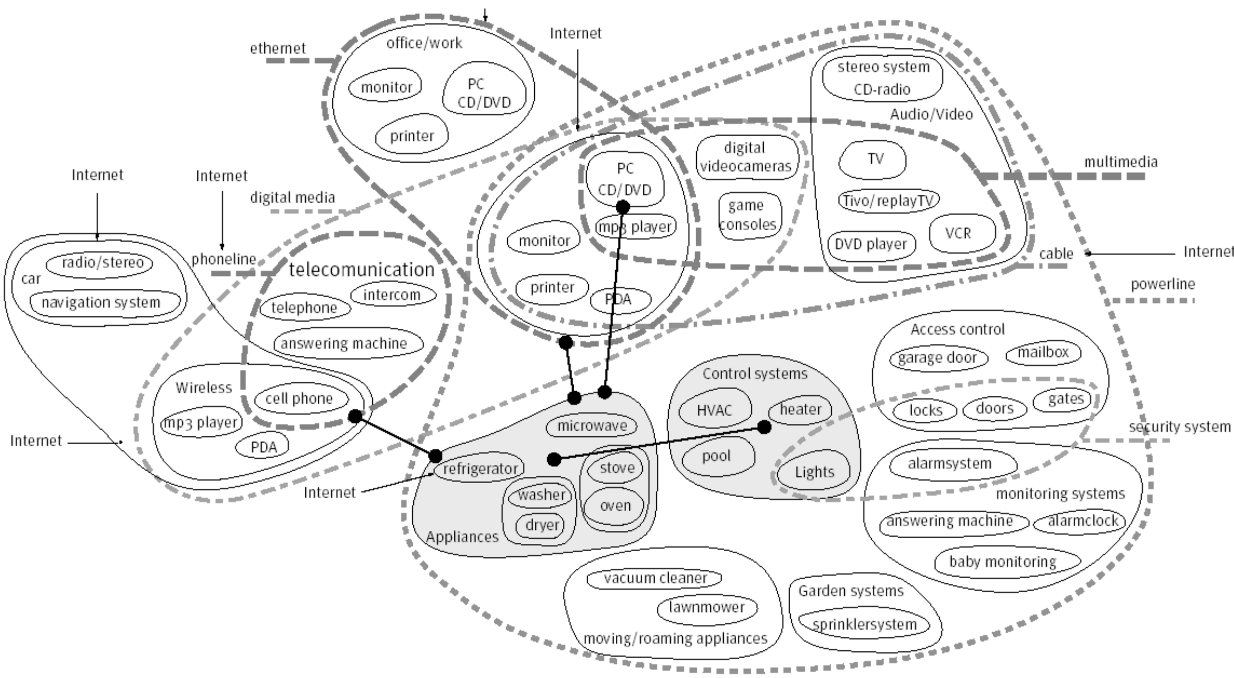
Although isolated constraint types can be identified, real world design problems are never isolated instances of one constraint or another. However, for the scope of this thesis, it is helpful to first look at individual constraint types and to suggest possible strategies in modeling them, before moving into how they might interact and overlap. The larger challenge is to implement robust constraint models that allow for linking different constraint types and develop computational models for digitally supporting design exploration.

Constraints in the context of a design exploration can become *design drivers*. A design driver is a prominent constraint in design exploration that is not easily changed and provides the strongest influence for directing the exploration. The claim is made in this thesis that constraints can play a key role as design drivers in triggering design innovation.

The chapter is structured as follows: First, the computational modeling of constraints and how these are identified in a design problem is discussed, how their analysis takes place, how constraints are translated into design drivers, how constraints are applied to a design domain, and how one may explore a solution space for a given set of constraints.

Second, the use of constraint modeling in design exploration is discussed. Different types of constraints are distinguished, bidirectionality is introduced, the translation between design representations, and the relationship between design domain and representation. Further translation models, bidirectional models, the circular design dependencies, the role of geometry and the use of programming in design exploration are examined.

Modeling constraints computationally requires both a robust theoretical model as well as a working digital implementation. Some examples of successful implementations and several diagrammatic models are shown. Solvers play a crucial role in constraint resolution and more specifically in the recent development of realistic cloth simulation in animations. The need for cloth simulation has pushed the development of robust and efficient solver techniques (Baraff and Witkin 1998).These solvers work well for problems that can be expressed as ordinary differential equations (ODE). However, not all design constraint problems translate into ODE's. Constraints exist in many forms, and constraint solvers are developed domain specific. For instance in contrast to ODE based solvers many constraint solvers in artificial intelligence are based on unification algorithms, a cornerstone of automated theorem proving (Krzysztof 2003), a class of constraints not addressed in this thesis. Therefore an important first step is identifying the design problem and its constraints and finding an appropriate constraint implementation. It is suggested that a useful technique is mapping out the components involved and identifying adjacencies between them. The next step is to identify the constraints and their relationships in order to turn the analytical approach into one that leads to novel design solutions.



Home networking analysis - mapping a design space.
The accumulation of information on the design problem helps to form the design space and possible approaches for exploring it. Formatting the information in the diagram potentially reveals more than what was put into it by exposing patterns and tendencies on the visual level that would not have been apparent in the data alone. Any diagram reflects a position and certain judgment due to its speculative nature. This reduces its value as an objective evaluation instrument while at the same time opening potential avenues to design. The thesis lays out computational approaches for design exploration that aim to capture some of the qualities of the diagram for digitally represented design problems.

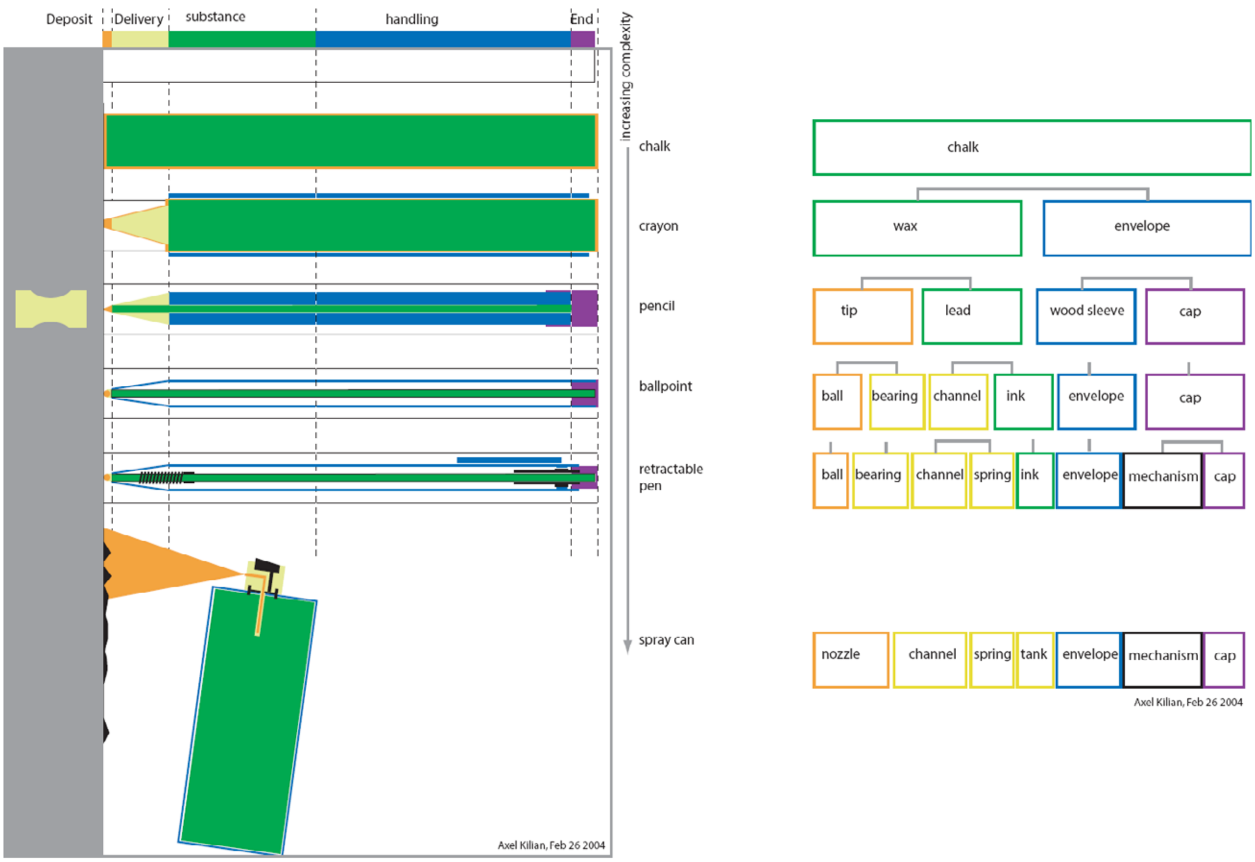
3.2 Identifying the Design Problem. Mapping the Domain

The definition of the design problem itself is a major step in setting up any constraint analysis. In order to understand the contributing factors and the cross dependencies a first step is to map the domain. An example for the mapping of a design domain is given in the diagram example below. The task is to identify the potential for new networking standards for the home networking market from analyzing the existing home infrastructure. Rather than using a matrix-based approach, the common devices in the home were positioned in relative networking proximity to each other to serve as a basis for identifying emerging network proximities. Proximity in the context of the diagram refers to both physical proximity of the devices in the home as well as proximity in terms of their being connected to the same or adjacent existing networks such as the power grid, an Ethernet or a wireless network for example. In addition emerging proximities in usage are taken into account, such as the increasing intermeshing of entertainment and computer equipment. Here the graphing of the design problem reveals emerging properties on a visual level. The mapping of design-relevant relationships, features, and properties can reveal proximities between elements of the design problem that were previously overlooked. It is a first step to externalizing the design problem and visualizing it. The specific diagramming technique evolves with the design problem that is being pictured and is part of the design approach. This example reflects the recording process of the gathered information in a spatial and visual manner in the course of many iterations throughout the gathering of information.

The accumulation of information on the design problem helps to form the design space and possible approaches for exploring it. Formatting the information in the diagram potentially reveals more than what was put into it by exposing patterns and tendencies on the visual level that would not have been apparent in the data alone. Diagramming the design problem is a form of evaluation and helps in formulating position for design additions. The example of mapping a domain demonstrates the first step in structuring a design problem and in identifying the constraints.

3.3 *Analysis of the Constraints in the Design Problem*

The description of a design problem involves analyzing the problem for existing constraints and degrees of freedom. One possible approach to do so is to identify the commonalities between a sample of existing design solutions to the problem. The example shown in Rather than choosing an approach based on the decomposition of

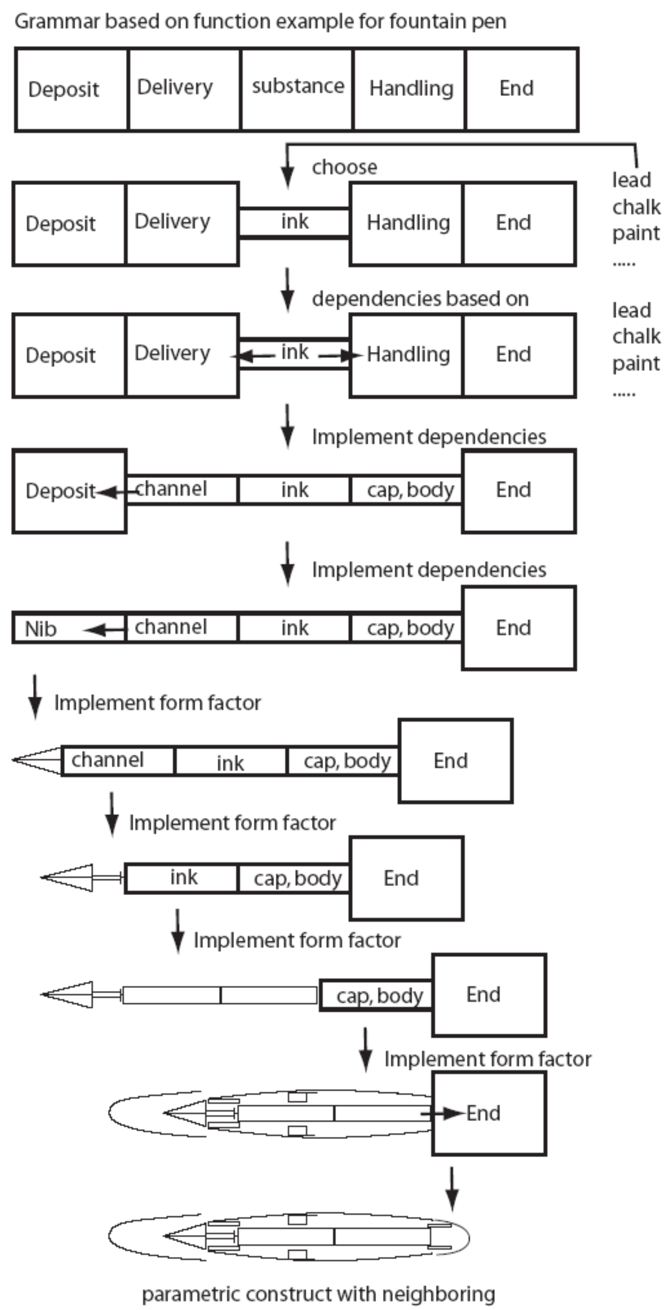


Writing device analysis exercise in design workshop, instructor Mitchell, W. J., Media Lab, MIT. Development of the writing device analysis Axel Kilian.

components, a function-based approach was chosen. The function-based analysis allows for more flexibility in the description and a more general description across solutions with very different form factors. In contrast, a grammatical subdivision based on the parts of existing designs would limit the exploration to dimensional variations of existing designs, or at best combinatorial variations of parametric parts. In the example the examples range from a basic piece of chalk to a mechanized retractable pencil.

It is suggested to go beyond part decomposition and instead abstract the higher level common functional description of the set of design solutions. The functional description then is the starting point for a component independent design implementation that expands the studied set of precedents.
The pen diagram shows the different writing device functions coded by color. The first column shows the analysis of the existing

Development of *function chains* for capturing a design problem and generating novel design solutions for it.



devices, from a piece of chalk to a spray can. The analysis identifies key functions in the writing process across the example set. For instance, leaving a mark with a substance is a core function of all devices, at least the physical ones studied here. Given the function of leaving a mark with a substance and a target surface, a choice for delivering the substance is necessary. From the example set there seems to be great variety in how the substance is deposited, calling for a deposit function as a separate node from delivery. Last, the handling question triggers a design response for packaging. The substitution of functions with implementation choices creates functional demands. The difference between a function and the functional demand is that once a function has been substituted with an implementation choice that choice triggers a functional demand for its neighbors. A match between functions and components is not guaranteed for all the devices, which is a key difference of a functional analysis to a component-based analysis. But the absence of a one-to-one mapping provides the looseness in the design goal description that allows for novel designs. In fact, an implementation choice might bundle several functions into a single component such as it is the case for the chalk piece that implements leaving a mark, delivery and handling all in one step. Such bundling is one approach to lead to innovative designs not seen in the sample set.

The second column describes the devices in a tree-like fashion with increasing subdivision and complexity of the parts. The part hierarchy is appropriate for a descriptive approach but less so when aiming for design innovation.

The third column does exactly that. It provides a functional description of the family of writing devices and is based on five core functions of the set of studied writing instruments. Those functions are chained together based on interdependencies into a so-called function chain. The function chain works much like a design checklist prompting design choices without suggesting explicitly any of the studied precedents. When using a function chain its tokens are replaced with design choices one by one. For instance, the substance token may be replaced with ink. This choice triggers neighboring tokens to be implemented. Step by step, the abstract function chain is turned into a topology of functional demands. These demands form the basis for the next

step, which is form implementation. Functional demands are translated into geometrically variable components. Based on their neighboring conditions, they have to conform to adjacent tokens both in dimensions and propagation of functions. When following through this process a new design instance is created.
The example is kept abstract. The importance of the analysis of the design problem is emphasized before setting up a constraint network in terms of parametric topologies or functions in order to allow for novel solutions. Parametric variations alone can only cover a very small spectrum of possible designs within dimensional and compositional variation of parts, rather than providing conceptual variety. These limitations of parametric descriptions are often neglected in light of the current interest in parametric modeling in design.
Diagrams help in the exploration of function chain constraints. An analysis leads to a formal description of a solution space. That formal description is translated into a function chain that can be used to generate new designs within the writing device family. The functional constraints define the functional demands which in turn give rise to the implementation constraints. Because they apply to an earlier design stage, they have the potential to influence the design much more dramatically than the parametric variation of parts alone.

3.4 Translating Constraints into Design Drivers.

For design exploration it is necessary to translate constraints into design drivers. This can be accomplished by setting up an exploration system that incorporates the constraints but allows for exploration of the unconstrained aspects of a possible design. The hanging model implementation is an example of this approach and demonstrates the use of a solver-driven architecture based on particle and springs for tension-only form finding (Kilian and Ochsendorf 2005) (Kilian 2004). The hanging model is the exploration of geometry within the constraints of a given topology and structural behavior. In addition to adjusting the geometry to meet the structural constraints, the environment can visualize the approximate material envelope in proportion to the forces in the geometric members. This ties in an additional constraint, the load bearing capacity of a given material. The material factor defines the minimal allowable cross section for the beam members under load according to local buckling length and in the implementation it provides the designer with a live feedback of member volumes in the current design configuration under the given material constraint.

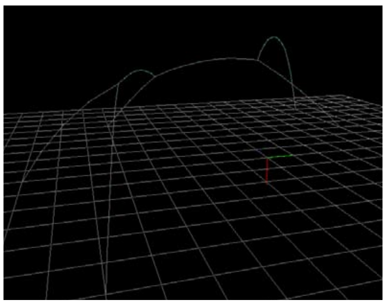
In the implementation, an interactive model uses solvers to negotiate between a fixed topology and a slightly adjustable geometry based on simulated spring and point masses. Properties are embedded in the behavioral model through those primitives. The network of primitives is solved for equilibrium of forces through a solver. The digital implementation allows the user to shift the design driver. One way is through editing the topology, another through editing the geometry by changing the lengths of the members, and last by adjusting the material parameters which control the allowable cross sections of the beam extrusions based on the forces present in the members.

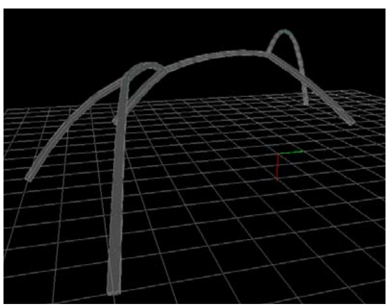
3.4.1 Applying constraints to a problem domain

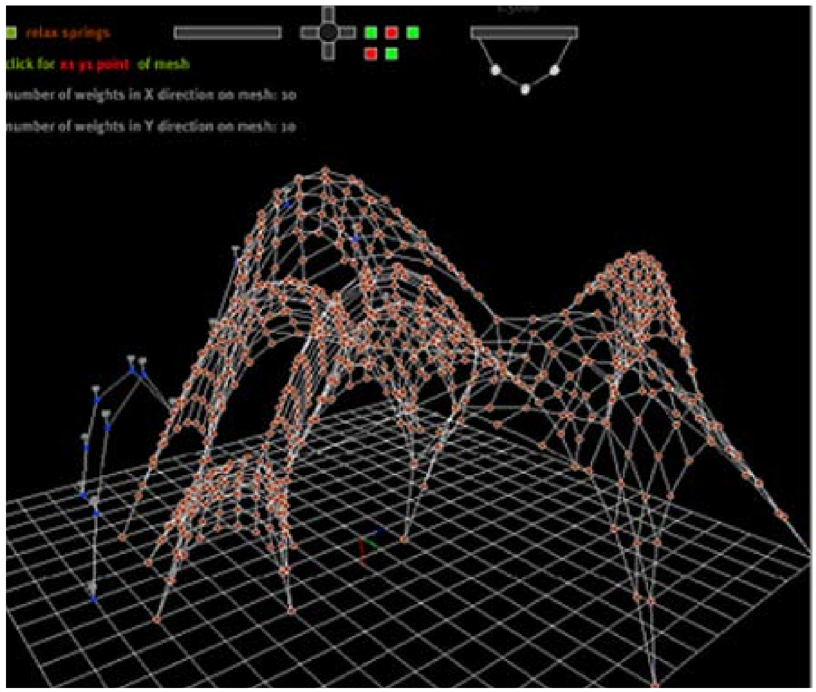
New constraints may also be applied to an existing design domain. For instance, this is the case with the increasing use of digital generative design in combination with CNC fabrication techniques in architectural design (Aranda and Lasch 2005) (Kilian 2003) (Loukissas 2003) (Kolarevic 2003) (Leach 2004). The fabrication techniques as well as the digital generation of the modeling information have specific constraints that apply to the design artifact. Examples are fabrication techniques based on flat sheet material. Flat-sheet fabrication and prototyping has increased due to the availability of relatively inexpensive CNC machinery for cutting, such as laser and plasma cutters. These machines are now available in design studios as well. The developability constraint imposed by the flat sheet material and the fabrication process has led to the development of a fabrication-specific design language in the construction industry similar to developments decades earlier in product, ship, and airplane design (Shelden 2002) (Coons 1964) (Casteljau 1963) (Liming 1944).

At the core of this example is the translation of free-form shapes into developable surfaces using the developable primitive of the cone. The application of this primitive can produce interesting new approaches to low cost fabrication of free-form surfaces. The example below is that of a cone-based translation of a free-form surface into developable cone-based parts.

In a more sophisticated version, the degree of curvature could control the spacing of the cones to achieve a consistent approximation error. Alternatively, a growth-based approach could be taken similar to the process in MoSS (Testa et al 2000) where



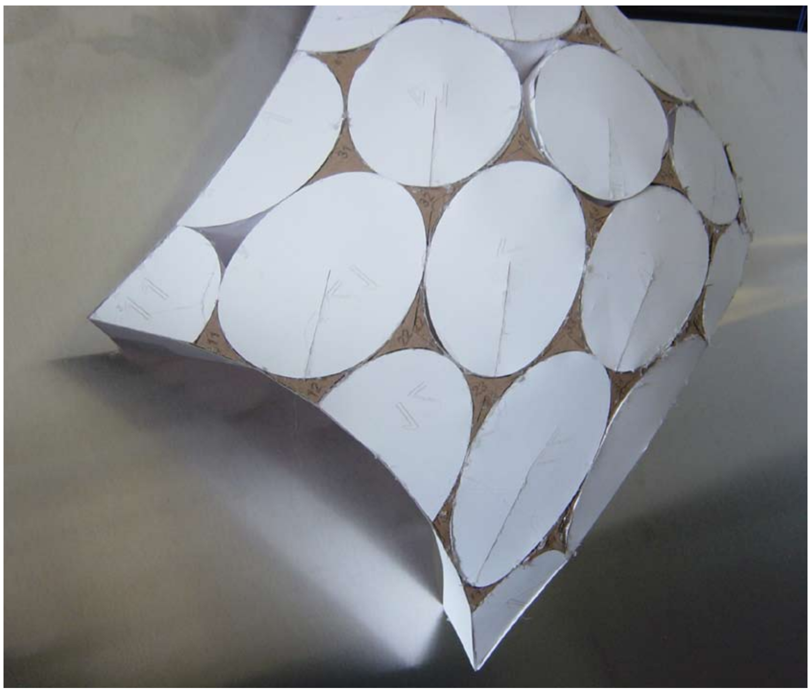




Top Left: Topology of the hanging model.
Bottom left: Linking the resulting forces to the structure envelope based on the material constraint.
Right: Mesh topology of the same environment. The constraint topology results in a geometric form.

an L-system guides the growth of a surface. In the example shown here, a simple close-packing-circle approach is used to produce the basis of the circular cones and the circle center points form the tip. The circular cone base is of interest as a spatial curve that coincides with the curvature of the surface while the resulting cone surface is still strictly developable. The fabrication constraint is therefore embedded in the geometric property of the chosen primitive, which makes it a very robust approach.

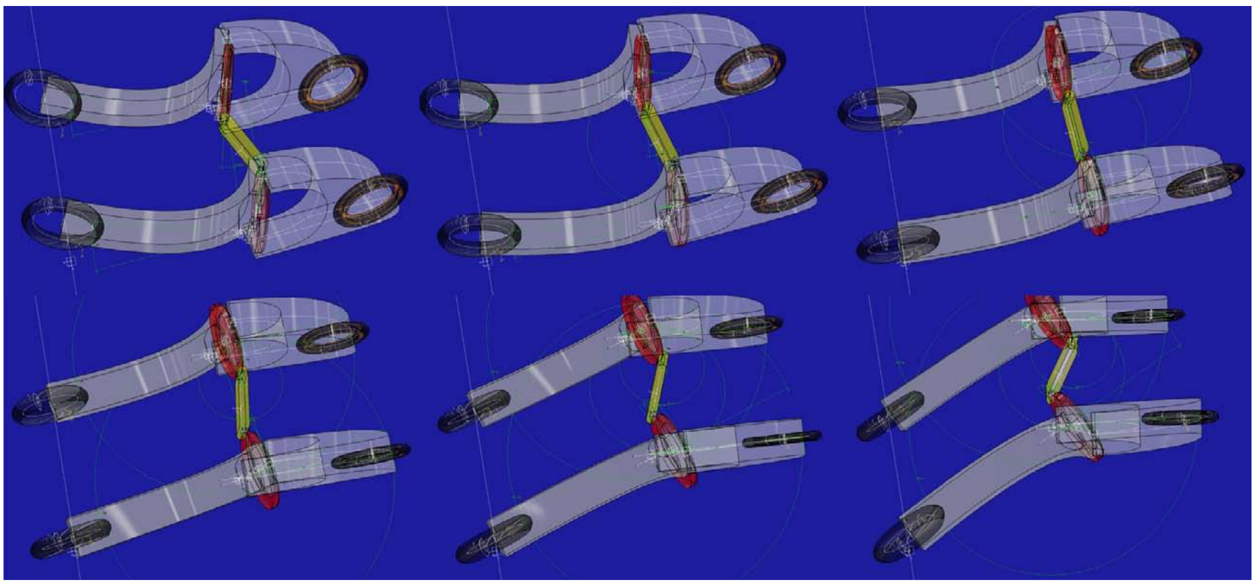
The next example of turning constraints into design drivers is taken from the Concept Car design studio mentioned above, headed by William J. Mitchell. The studio explored novel car architectures using a fairly open approach. One of the outcomes was the "athlete", a car that has many more degrees of freedom for steering than the two degrees of freedom required to steer on a planar surface. Articulation became the design driver for this concept car design study. The additional degrees of freedom provide the opportunity to explore alternatives to rigid car chassis designs. The challenge was to solve the constraints for a particular movement and to determine the relationships between the different degrees of freedom for a particular movement. To do so, the three-



Cone-based approximation of a double curved surface using developable surfaces only.

dimensional constraint solver in the CATIA© engineering package was used. The frame structure was modeled three-dimensionally and the components were jointed through geometric constraints. Relationships of movement between the different joints were then defined through equations. Finally, one of the joints was animated, which propagated the movement through the entire constraint system of the frame. This constraint solving study was used to validate the complex interdependencies of the six degrees of freedom. The study led to adjustments in the relationships of the constraints and to the addition of two constraints to compensate for non-planar wheel positions in turns. The problem had previously been studied in physical models, but only the constraint-solving approach provided results precise enough to detect the necessary adjustments in the joint layout.

Here constraint solving helps to find a working parameter configuration for driving the vehicle. In general it offers a direct way of resolving competing constraints in three dimensions.

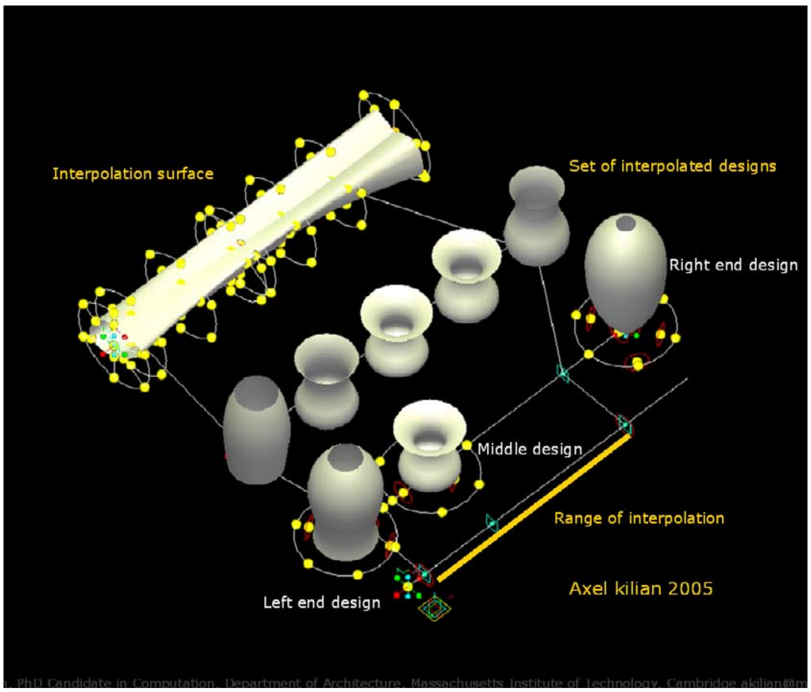


A three dimensional assembly for an articulated car with eight degrees of freedom driven by one parameter, the leaning angle. The remaining parameters are resolved through a constraint solver.
The "Athlete" concept, developed in the concept car design studio, Media Lab, MIT.
Team: Axel Kilian, Mitchell Joachim, Peter Schmidt, Patrik Künzler, Kate Tan, Luis Berrios-Negron, Lorene Gates-Spears, Timocin Pervane.

3.4.2 Exploring solution space for a set of constraints

Setting up a design problem through modeling its constraints is a first step to finding possible solutions. However, once a possible solution is found it is not guaranteed to be the only one or necessarily the best one. Genetic algorithms have been extensively used to optimize parametric objects for a given fitness function (Bentley 1999). The fitness function can be viewed as an implementation for enforcing a constraint in form of the fitness measure. The resulting objects conform to the constraint by selection although strictly speaking the fitness function itself is not a constraint.

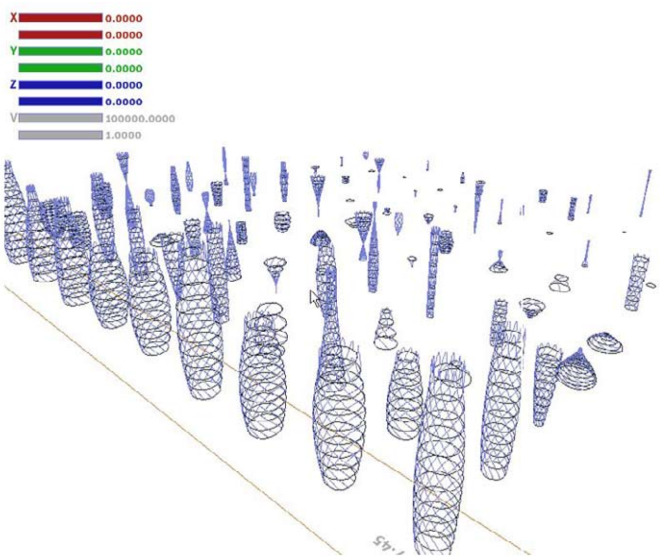
For the design exploration of lower dimensional parametric constructs, genetic algorithms often prove to be too cumbersome to set up or provide too little opportunity for user intervention in order to guide the process. An alternative, at least for lower dimensional parametric searches, is mapping parametric values to geometric control objects. These objects make it possible to capture desirable parametric settings and explore the neighborhood for small variations of the chosen values without losing the context of the previous result.

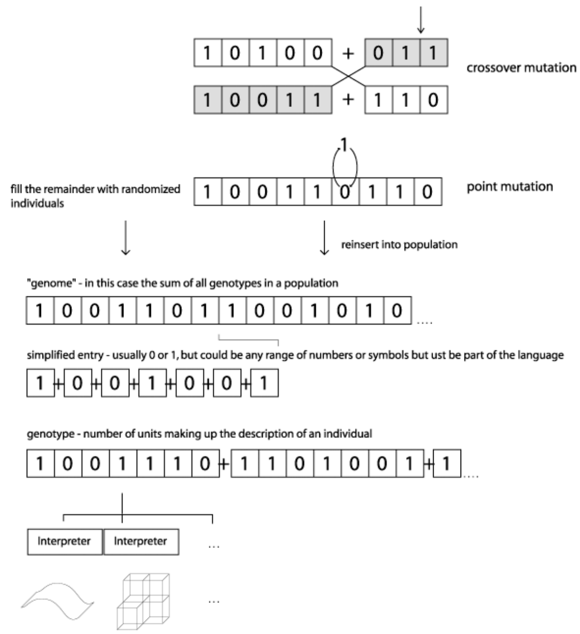


A vase controlled by six parameters. Besides their ability to support dimensional variations, parametric models also model topological constraints through the embedded part association. While navigating the solution of a parameterized object, one can record the different settings accordingly. This approach allows for interpolation of the intermitting parameters. A vase controlled by six parameters demonstrates a geometric control object to provide parameter interpolation and the possibility to record and memorize states of the parametric settings for the object. Designing parametrically poses the challenge of evaluating the range of possible outcomes presented by a parametric construct set up for the exploration. Higher numbers of parameters make it less intuitive to interact with the design construct.

Besides their ability to support dimensional variations, parametric models also model topologic constraints through embedded part association. While navigating the solution of a parameterized object, one can record the different settings accordingly. This approach allows for interpolation of the intermitting parameters. A vase controlled by six parameters demonstrates a geometric control object to provide parameter interpolation and the possibility to record and memorize states of the parametric settings for the object. Designing parametrically poses the challenge of evaluating the range of possible outcomes presented by a parametric construct set up for the exploration. Higher numbers of parameters make it less intuitive to interact with the design construct. The parametric control object also allows for another instance of a bidirectional

link as the control object is both generated by the sample design but also can by direct manipulation drive the set of designs.
A further example shows a family of objects whose parameters are mapped to one point's XYZ value in a grid that samples a surface based on parametric UV spacing. Moving the points of the UV-based grid changes the points XYZ coordinates which are mapped to the parameters and therefore regenerates the object family. By

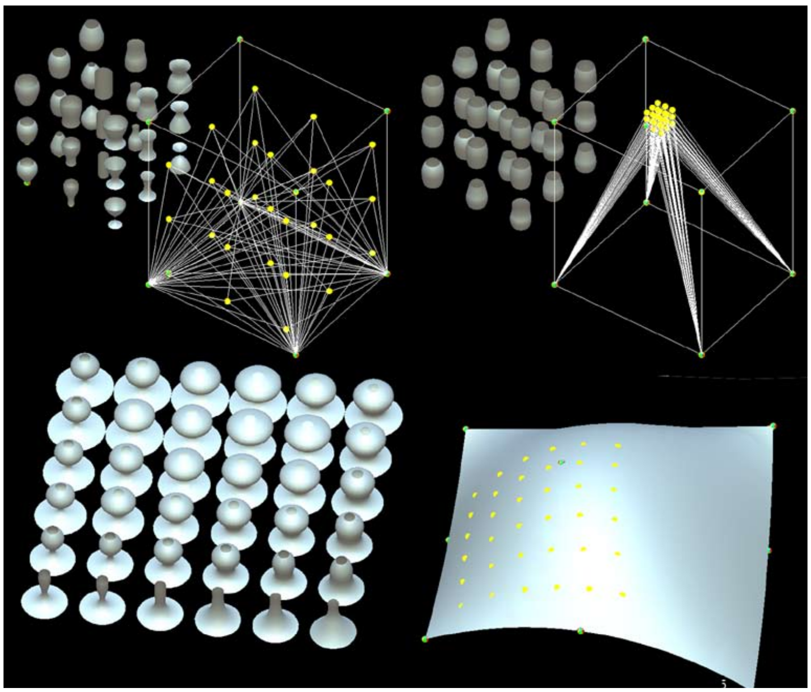




Genetic algorithm driven search for diagrid towers given an overall volume constraint.

increasing or decreasing the sampling rate around the points of interest, one can explore parametric variations in more detail where needed. As an alternative to a design surface, one can use a design volume for the exploration. Each point is one sample within the exploration range. In order to increase the resolution of variations around a particular sample, the sampling can be bundled around one parameter The mapping of preferable design instances in the solution space creates spatial traces, which can suggest additional members of the design family by interpolation of or through spatial proximity to the existing samples in the solution space.
Exploration is a central aspect of design. The process of defining the boundaries and constraints of an exploration helps to define the problem itself. The exploration of a design problem makes

the designer aware of the limitations and constraints of a design problem. Computational methods to support design exploration potentially enhance the designers reach and allow for the modeling of complex interdependencies that cannot be easily abstracted.



Mapping the solution space onto a parametric object. Exploration through sampling of the solution range via a point grid,

3.4.3 Types of constraints

As discussed above, constraints play a central role in the definition of any instance of design exploration. Constraints set the limits and the metric of evaluation for an exploration. Although they are a limitation, they do help to externalize the issues present in a design problem and thereby can turn from constraints into design drivers. The constraints illustrated in the examples above are only a fraction of the constraints present in design problems. The examples are meant to be suggestive of the possibilities that come about with the integration of constraints in computationally supported exploration. This is not necessarily applicable to all design problems and not all constraints can be readily translated into quantifiable representations. Those constraints require more sophisticated computational models to support them and are not addressed in the scope of this thesis.

3.5 The Role of Constraints as Design drivers

Constraints in the context of a design exploration can turn into *design drivers*. Their role as design drivers is determined by their attributed weight in the design problem. A design driver is a constraint that remains unchanged through out the exploration even if there are competing contradicting constraints present. Design drivers are constraints that are challenging enough to require a design innovation to satisfy them. The developability of the paper cone in the double curved roof study is an example of design driver. One could have explored other ways of approximating a double curved design surface but the developability constraint was chosen as the focus of the exploration. For the vehicle design study the design driver or main constraint was the idea of an articulated body. Most other design development was derived from this constraint

3.5.1 Bidirectionality in constraint modeling

To be truly supportive of design exploration, constraint solvers need to be bidirectional (Mahdavi et al 1997). This means the constraint network cannot be implemented as a hierarchically structured dependency tree that allows only propagation of effects towards the tree leaves. A graph requires a more general definition of part relationships since the possibility of cyclic dependencies exists. The circular nature of the network and the possible reversal of the propagation direction along the links requires the use of a bidirectional solver.

Constraint explorations are often only analytical in nature, meaning that a change in a parameter will produce a result but the result can in turn not be used as the driver to continue the exploration. In contrast, a classic example of bidirectional exploration of a constraint network is graphic statics (Zalewski and Allen 1997), where a force polygon is linked with a form polygon through geometric constraints and change can occur in both the form polygon and in the force polygon. This allows for the exploration of either form or force while each change in one representation affects the results in the other through the graphics statics constraints. When the form and force network becomes more complex, the mappings between the two representations are not fully determined as multiple solutions may exist for the form polygon for a given state of the force polygon. To explore the possibilities it is necessary to use a bidirectional solver. Several such tests were modeled using large graphic statics system modeled in CATIA®'s sketcher environment, which features a bidirectional solver.

3.5.2 Translation between design representations

The translation between different modes of design representation may involve a domain shift. As a result, the translation can be an incomplete mapping between the different representations due to the only partial overlap of their domain-specific information. This loss in translation requires a reformulation and a completion of the missing information in the new domain and is a key trigger for innovation. The reformulation forces the designer to re-describe the approach in different terms; often he or she discovers the core qualities of the project in the process.

3.5.3 Translation model

The translation model is an externalized intellectual, computational or physical construct that can relate design constraints in one representation to design constraints in the target representation. The majority of engineering models are analytical in nature. Their mapping occurs in a one-directional manner from one design representation to another. A classic example is structural analysis where the analysis of the form of a design produces a representation of the design in terms of forces. The translation is unidirectional. The force representation in isolation does not allow the reconstruction of the form if the process is inversed. The translation only works in one way and makes design exploration involving the different domains difficult. In contrast, the ideal translation model allows for bidirectional mappings between the design representations. It is suggested to instead define a translation model as generating a new design representation by carrying over the constraints rather than translating the representation.

3.5.4 Bi-directional models

The condition of bidirectionality is a reappearing challenge throughout the examples in the thesis. It is often mistaken as a lesser version of multi-directionality. However, it does not refer to connectivity in a design topology, but rather to the property of the translation model. Bidirectional stands for the translation in both directions without preference for one over the other. The network of interlinked design representations can go beyond that of a pair of domains described above. In fact, most design problems are circular in their cross dependencies, meaning the chain of influences wraps back on itself.

3.5.5 Circular dependencies in design problems

Circular dependencies are far harder to explore than non-circular ones due to the feedback loops among the different representations. Most designs rely on a number of interlinked representations linked by constraints shows how the different design representation in a chair design can be split up into design driver, implementation and representation and how these are interlinked though constraints. They create a complex construct even for relatively simple design problems such as the chair shown here. Externalizing and describing these constructs is a major part of the design process. The exploration of variants is based on this externalized construct. This construct is being referred to as design explorer, which differs from the definition of design explorer given in Gross (Gross 1985)

3.5.6 Geometry and design exploration

Geometry is the main vehicle for design representation. In many cases, geometric design representations work successfully across different domains, and the ability to represent design in multiple domains is crucial to allow for design exploration.

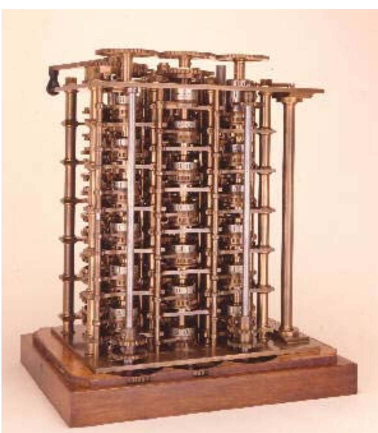
Geometry is a very powerful representation format. It covers a wide range of design representation from descriptive geometry to abstract visual shape, but it does not cover all design representations. The translation of other design representations into a geometry format is not *lossless*. Geometry's central role is further strengthened by the support of digital design environments that are based on geometry. The underlying math is robust and straight forward to implement and acts as the backbone of geometric design representation. Geometric procedures in the form of tools in design environments can influence the choices designers make in their design representation. The integration of non-geometric design components into a central geometric design representation requires models of translation. Developing such models by extending the digital design environments is becoming increasingly popular among designers. One reason seems to be that many students and practitioners have reached certain fluency with the existing tool sets and strive to distinguish themselves through customized design processes. These can be created through scripting or programming extensions of commercial programs. Programming therefore is becoming an important skill to create one's own, design-specific digital context. Geometry may still serve as the output of the design, but the design representation shifts, at least partially, into the realm of programming.

3.6 Conclusion

The case studies show the combination of a number of established computational principles such as genetic algorithms, parametric modeling, and graphs in support of design exploration. Ideally, the modules that implement such computational models would be more easily accessible and combinable in the design environments in use today. Some computational design environments are developing in this direction, most notably processing by Ben Fry and Casey Reas Fry and Reas 2006) (Fry 2004).

A bigger challenge for future work lies in the development of solver architectures that support constraint resolution for non-geometric constraints reflecting the heterogeneous nature of design problems. Constraints exist in many forms and there is no master model that can incorporate them all, but improving the bridges between different isolated constraint models could improve the availability in design.

4 Simulation, Surface, System, and Search



A fragment of the Difference Engine No. 1 (1832)
Image: © Science Museum, London. Science & Society Picture Library

This chapter explores four concepts fundamental not only to the work in this thesis, but to digital computation in general. Their evolution and current use are outlined to provide a background for the experiments discussed in the following chapters. To begin, it is worth reconsidering the evolution of the digital medium.

The digital medium has gone through several stages of development in maturing into what it is today, and it continues to evolve. In its early phases, its mainstay was that of scientific and military calculations, very much perceived as the industrialized version of the human "computers" doing the labor-intensive calculations needed for a society that rapidly developed commerce, shipping and industry. Tide charts, navigation tables, or accounting were all tasks early digital pioneers envisaged machines to take over and many of the organizational principles of reducing complex calculation to addition and subtraction that could be performed by common people laid the foundation for modern computing architectures.

But all early computation devices were purpose-specific machines, and all were analog, mechanical machines closely related to their mechanical cousins, the mechanical looms and the steam engines.

Where was the transition to digital computation? Technological development was not the only hurdle. There was also a cultural hurdle in recognizing available technology for a new use. Despite early pioneers like Charles Babbage there was little development in mechanical calculators. The theory underlying Boolean logic and eventually all digital processes had been defined long before the computation field realized its significance for general purpose machines. This shows that significant breakthroughs do not rely on a single factor, and might be delayed for decades, even a century, despite the presence of all the essential technical components, due to lack of understanding or appreciation of the culture the processes take place. Charles Babbage's visions of computational machines and his first prototype of the Difference Engine in 1832 and the Analytical Engine in 1871 are such examples (Hyman 1982). This phenomenon does not seem to be particular to modern times, but seems to run through the history of civilization in the development of most aspect of modern live starting from agriculture to industry (Diamond 1999).

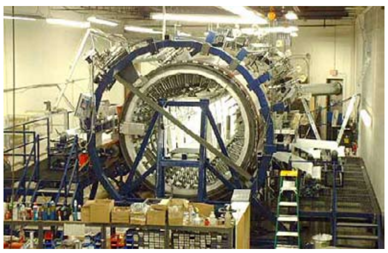
So what catapulted digital computation devices from marginalized, cumbersome, and immensely expensive research devices into the digital computation mainstream of today?

Ivan Sutherland operating his application Sketchpad.
Image: © Sun Microsystems

4.1 Simulation

Accounting and business applications, as well as word processing certainly played a mayor role in fueling the development of the early generations of hardware and software. But at that point after a long slow development period from the 1940's through the war, computers were not yet a mass phenomena and far from their ubiquitous presence of today. What seems to have had the biggest impact in their later explosion was the development of digitally based simulations. These were made possible by the intersection of several factors: the development of the scientific principles required to construct the theory and underlying models of simulation; the development of hardware enabling real time simulations; and, most importantly, a society with problems where simulations could make a significant difference in predicting or testing complex scenarios, including weapon development, starting in WWII, air plane design, the automotive industry and science. The computer game industry followed with a slight delay. Today, its economic power is a major

factor in pushing the hardware development for a mass market.
The field of architecture, once at the forefront of the development of simulation with perspective depiction in the Renaissance, today is largely riding along, benefiting from the developments in computer hardware and software development, driven mostly by engineering and science. Still, significant contributions to the development of computer aided design have come from architects. In 1963, Ivan Sutherland's first parametric software foreshadowed most of the principles of interaction of today's computer applications, namely parametrics and vector-based design representations on visual displays (Sutherland 1963).



A computer controlled sail making drum by ©Northsails using computer numerically controlled arms to lay down strands of carbon fiber along the lines of force. The technique is called 3dsl.
Image: © Northsails

The main point in this section, however, is to point out the significance of the concept of simulation for working with a disembodied digital medium, which posed and continues to pose as much a technological as a cultural challenge for the culture of design. To embrace simulation and drive it beyond its role in replicating what surrounds us, is a challenge that has not been met.
For design, especially in the architectural context, the early beginnings were very promising as they coincided with an immense optimism about the potential of the emergent medium in all disciplines. The early adopters were few but capable and faced little precedents to influence their choices in the young medium at least in their domain. The influences therefore came from other fields more than from within in terms of computational models. Expert systems, developed in the young discipline of artificial intelligence, were adopted by many in guiding design systems in digital forms. Simulations focused mostly on the abstract design process, fittingly to the performance ceilings of the computers at the time. In a way, this was a good thing as it focused the efforts on what turned out to be the hardest problem of all, both in artificial intelligence and in the design disciplines: on how the mind works.
In the following decade, with the increase in computing power, the attention shifted towards the visual, and simulation became more and more synonymous with the recreation of perspectival space as a stand-in for reality.
Simulation in architectural design became increasingly synonymous with producing an image of three-dimensional space. The image was created from a geometric model of the three dimensional

space, geometry, used to capture and represent design. These dependencies became the main driving force of architectural design in the digital medium. In connection with the increasing performance of computational hardware, which seemed to feed off of and at the same time feed an ever increasing demand for speed, architectural design became driven by the pursuit of photorealism as the ultimate goal. Needless to say, this goal was achieved at the visual level, but at the expense of the development of simulation in other domains. Performance-based simulations may be the big exception, since they were catered to by the engineering disciplines, driven by the pursuit of perfection of numerical data and its visualization. The reign of visual simulation came largely at the expense of theoretical developments, and at the expense of software development catering to the simulation and implementation of the design process as a computational process.

To summarize, the dominance of equating simulation with image production has left geometry as almost the sole design representation in the digital medium. While being immensely powerful as a representational format, it is not versatile enough to capture all aspects of design. Alternative design representations have developed at the periphery of geometry at best, but do not play a major role in digital design systems. Of course, these shortcomings go beyond the development history of computer hardware and software. Images are treated as product but not as input, despite there importance in design. The shortcomings relate as much to the theory of image recognition and processing, as to the lack of understanding how our own visual perception exactly works and computes.

Closely linked to the rise of geometry, fueled by the allure of the image, is the role of Surface.

4.2 Surface

A surface is a geometric concept as abstract a geometric entity as a point or line, the basis for simulation as discussed above. Where does the concept of a surface originate?

The concept of surface became important in the development of representing and rendering in computational geometry. In computational geometry, everything that relates to visualization operates through surfaces. Design representations are altered by the need to visualize them, as in the tessellation of geometric objects for the sake of calculating ray surface intersections and surface incident angles for visualization. Most importantly surface functions as the carrier of textures in light simulations. The topic has been discussed extensively, both in computer graphics and in design culture. But a relatively recent trend in representation has been digital fabrication, which has taken over the spotlight of interest from visualization. Curiously enough, the surface continues to play a major role. In fact, the role of geometry, and the role of surfaces in particular, have strengthened in fabrication, as the demands on precision and on the control of geometric features has dramatically increased in comparison to visualization.

The rendering pipeline with its homogenous digital, internal, predictable and repeatable process chain ending in a pixilated, digitally fixed resolution image is far from the messiness of the digital fabrication process. The introduction of material and computer controlled machines breaks out of the fully enclosed, digital-only environment. Fabrication challenges the current role of geometry as a static design representation due to the simple fact that the digital model does not equate the model fabricated on its basis. Many problems remain unresolved by necessity: the prediction of the behavior of material is difficult. This addresses the issue of design translation, necessary when moving from one design representation, the digital- geometric, to another, the physical-material. The concept of surface plays a major role in the understanding of, description of and prediction of both.

Some might point out that this argument is failing to address the development of solid modeling in the last decade or so. In fact, even solids are represented through surfaces, not in all cases computationally, but certainly visually.

Surfaces in design representation and in the digital tools that operate on those representations are prone to misunderstanding from the fascinating ambiguity of the concept of surface. It is an abstract construct, but through the development of rendering it has acquired a quasi realism as an object, because it can be seen and be dealt with although it has zero thickness or encloses an imaginary volume in form of a solid. This perception of surface as object is at the core of the many challenges of translating geometric surface representations, which includes geometric solids, into physical space. It is hard to believe otherwise that the computer controlled manufacturing had evaded architects for decades as the next big thing if it had not been for the fascination with surfaces in relation with images driven by the longing for simulation of a visual reality. Only now that the visualization run is coming to a level of maturity does the focus shift. But interestingly enough it is not shifting the notion of design representation, it is still geometry at the core and more than ever surfaces that carry the main load of design representation. Even the occurrence of scripting and programming is almost exclusively focusing on a descriptive of geometry generation, little different from the manual process of creating it through the interfaces. This is understandably since most designers have been introduced to the medium from that perspective, know the medium as a black box and judge the accomplishments of their peers through what appears hard for them. A script that generates thousands of varied elements in seconds is overwhelming when one judges it from the manual investment necessary even to create a fraction of it by hand. Computer graphics approaches the medium and the generation of geometry very differently. Programming is used to create geometry from first principles; it is generative from a procedural approach and does not tend to occur through the interface of tools conceived of to be operated manually. For instance a irregular triangulated mesh is not only inconceivable to generate manually efficiently but also a simple additive process would make little progress in creating whereas a procedural algorithm using recursion based on a mesh creation principle can produce infinite variations of in the most general way.

But these abstract computational principles are hard to match with physical based constraints of materials and fabrication procedures. They require once again models of translation. Interestingly fabrication spurs the interest in these models of translation and alternative design representation to static descriptive geometric one much more than rendering ever did.

What are the surface conditions that occur in design and are of interest at the borderline between design representation and the physical and also responsible for many of the frustration in designing with today’s digital tools and possible the motivation for a designer to expand their skills and understanding of the problem and change it.

4.2.1 Surface themes

Surface as a theme in design has many roots in everyday experiences and other disciplines besides design. Some of these themes are expanded here based on past experiences and frustrations with surfaces and also based on the tension between surface as

"You cannot escape from a submarine"- surface as object example from a design studio project by the author. The surface describes a non object which evokes volume where there is only self intersection.

phenomenon and surface as artifact.

4.2.1.1 Surface as object

A surface, in the geometric sense, is an infinitely thin, continuous construct in space. The visualization in digital modeling applications presents the surface as object. This leads designers to use surface as object, in the literal sense, and to believe in its objectivity because it is visible as an object in the simulations. While offering a powerful array of surface-based design processes, the objectification of surface is responsible for much of the frustration we see in digitally based design today, as it disrupts the translation of abstract surface into physical implementation. The notion of surface as object also partially explains the love affair of architecture with visual media, since it allows for an extended honeymoon between the surface

object and designer. Images are simply more forgiving than physical space.

4.2.1.2 Surface as image

Every image is based on a surface of some kind, be it paper or a screen. Surfaces are carrier of images. Any portrait or depiction relies on the collapse of its higher-dimensional information into

Perspective construction of images in two dimensions is only one of many models to reduce higher dimensional space to a two dimensional image. Surfaces act as the carrier of images. Self Portrait 2002, Axel Kilian, composite of 14 image slices.

Cuttlefish are, despite their relatively small size of about 10 inches, masters of disguise. Texture in some animals like the cuttlefish show variability both in color texture and bump textures as well as animated texture changes. This level of surface as environment mirror has not been reached in man made objects. Responsible for these changes are the chromapores in the skin of the cuttlefish.

two dimensions.

4.2.1.3 Surface as texture

Through the dominance of rendering as a visualization technique, texture mapping has become the prime process of applying images to surfaces. Texture mapping refers to the spatial positioning of a desired image on a surface, a sophisticated form of photo montage using surface bodies rather than flat cut outs as a carrier. This process has lead to the widespread representation of design objects as surface constructs, rather then as lines or solids. In a design culture focused on the production of images, surface constructs offer the fastest and most flexible return. The examples in animals are far superior in their embodiment of texture in the skin. The example of the cuttlefish proves this

4.2.1.4 Surface as craft

There is a long history of surface based craft. Surfaces function as the carrier of ornament in an interesting variant to the texture of animals. Particularly armor and clothing triggered this craft long before a technological driven understanding of material and

geometry. In a way, this history lays out the physical precedents of digital texture mapping. The craftsmanship that went into the construction and decoration of armor can compete with much of today's industrial design, and is far superior to any façade details found in architecture. The sophistication of expression and the fusion of form and function within the constraints of the time is a great example of surface as craft. There is no technological determinism in the development of sophistication but a parallel of multiple lines of investigation and expertise, some technological driven some not. The maturity of a field of study can have an equally important part in the sophistication of its solution as its technological development. In armor the refinement of surface detail and response to the complex curvature of the body, in combination with its protective function, makes armor a

Armor is an early design example of surface craft that responds to a multitude of competing constraints. In the armor one can read the form of the body, the constraints of the material, the constraints of articulation pieces and the desire for ornament. There is no technological determinism in the development of craft. Skills are lost or go out of style.

formidable example of surface-based craft. This is not at the scale of architecture, but the example of the evolution of armor surfaces stands as a reminder that progress is not certain through the use of novel tools but also through the constant refinement of what is there. Also the principle of design surfaces as the underlying guide for the creation of geometric components is closely related to the production of ornament, which is also carried by the underlying

Surface as solid in the example of a Mayan sculpture. The surface is only an intermediate state in the continued removal of material.

surface. Its digital version is the natural extension of texture mapping in the visual to component mapping in the spatial constructs and through fabrication the physical.

4.2.1.5 Surface as solid

The surface that is revealed in the subtractive process of sculptures is a different one from the objectified surfaces of digital modeling. The surface of a solid stone is the temporary boundary between

The surface acts as a unifying factor in heterogeneous assemblies such as that of an airliner. Homogenous surface allows for unifying paint layers that reinforces the impression of the assembly as a whole.
Image: © Christophe Ena/AP

material and air certain only in the fact that nothing can be added to it anymore once it is removed, which makes sculpture so absolute

4.2.1.6 Surface as assembly

The use of a wide variety of materials and the increasing engineering optimization of high performance constructs like planes and cars leads to collages of materials and assemblies that are unified by little more then their structural integrity and the continuity of their perceived geometric surface whole. In the case of the material patchwork of a modern plane the paint coating proves essential

In a shifting medium such as sand or snow or water, surface is a intermediate state of boundary between different substances. Surface is most abstract here as it is temporary and constantly evolving. In fact each surface change affects the forces shaping the surface itself.

not just for functional reasons but as a unifying mode actuating the geometric continuity over the material discontinuity.

4.2.1.7 Surface as boundary

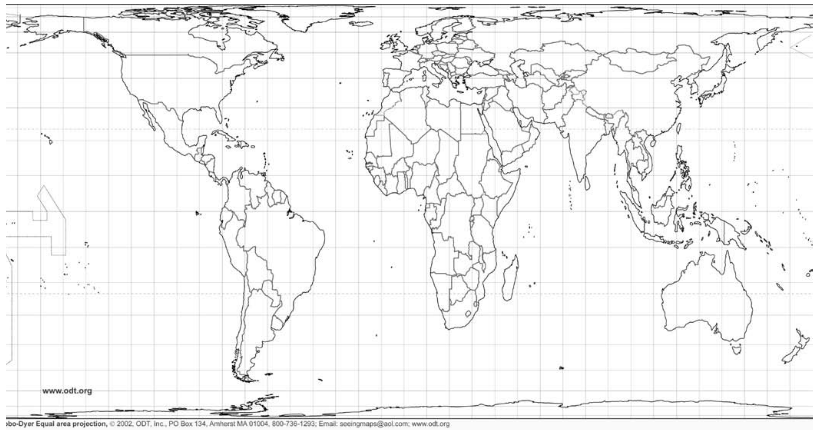
The image of a snow drift in the courtyard of a building exemplifies surface condition as a boundary between to shifting aggregate in this case water in the form of snow and air and how the forces of the wind create ever changing constellations and boundary surfaces between the two. Here surface is a constantly forming and disappearing condition neither object nor state but that what is at the foremost boundary at the division between the two substances. This condition makes the topologic description of the surface far more complex as it is an emergent property rather then

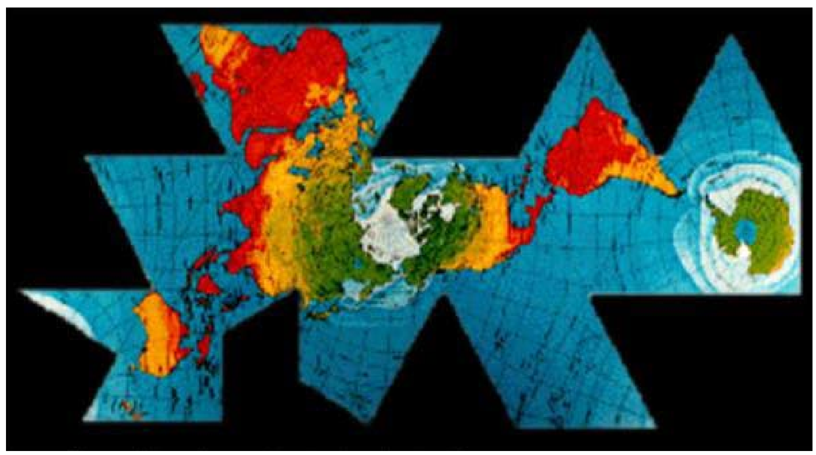
one that could be described and therefore object based modeling approaches are likely to fail in capturing the phenomena.

A mesh is a variant of an assembly based surfaces. Meshes are fascinating as they embody surface through the constant deviation from the ideal surface. Surface emerges as the average of the sum of the strands of a mesh or weave.

The act of unfolding and unrolling of geometric objects into planar surfaces are conscious acts. Surface choices can reflect politics as it is the case in the difference between a Hobo-Dyer equal area projection of the globe and Buckminster Fuller's dymaxion world map. One preserves areas the other one distances and local connections.
Fuller created the dymaxion maps so they can be reconfigured in many ways by moving the triangles around. The shifting of the view point supports his one ocean theory of the disappearance of political boundaries and artificial divisions of the worlds oceans when the world is viewed from space.
The classic world map such as the Hobo-Dyer projection or the Mercator projection stem from a Europe centric culture of viewing the world.

Image: http://www.odt.org/freehdp.htm
Image: Buckminster Fuller Institute http://www.bfi.org/



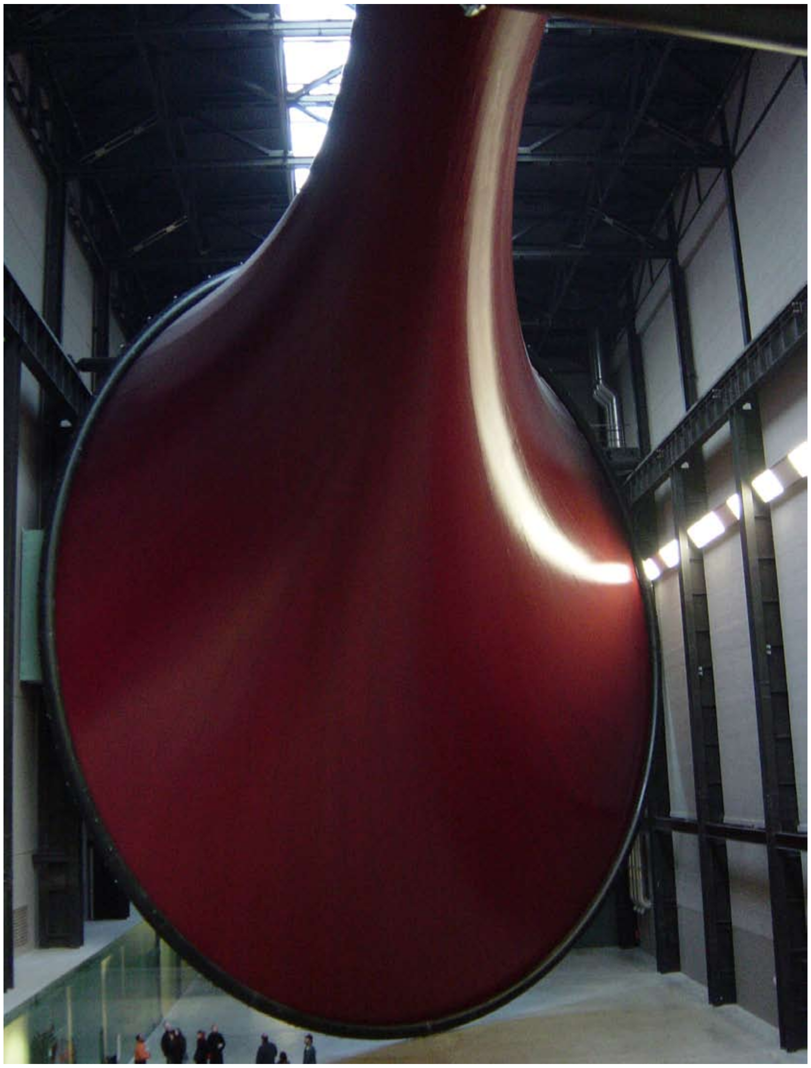


4.2.1.8 Surface as mesh

The accumulation of parts in close formation can evoke surface like readings like in chain mail or fences. Here it is the emergent reading of the overall trend in the parts that signal the surface character

4.2.1.9 Surface as politics

Choice of surface panelization conveys a message and in the case of maps has always reflected the political view of the map maker. The dimaxion map by Buckminster Fuller was his attempt to provide a more realistic view of the geographic landscape in the



Surface can act as a structural member as in the Anish Kapoor installation at the Tate Modern in London in 2003 (In collaboration with Cecil Balmond) and Arup. The tensile membrane was engineered to be in constant tension

light of the cold war threat in placing the north pole as the shared neighborhood in the center of his triangulated globe unfolding as opposed to the Hobo-Didier projection with its maximal distance based on the equatorial distance between the then super powers USSR and USA.

4.2.1.10 Surface as structure

Curvature in surfaces provides a structural potential. But the form is constraint to the interdependency of material resistance to tension and compression, weight and local and global geometry of the surface.
Structural shells have a rich heritage in engineering and architecture but their formal vocabulary has been relatively limited and maybe

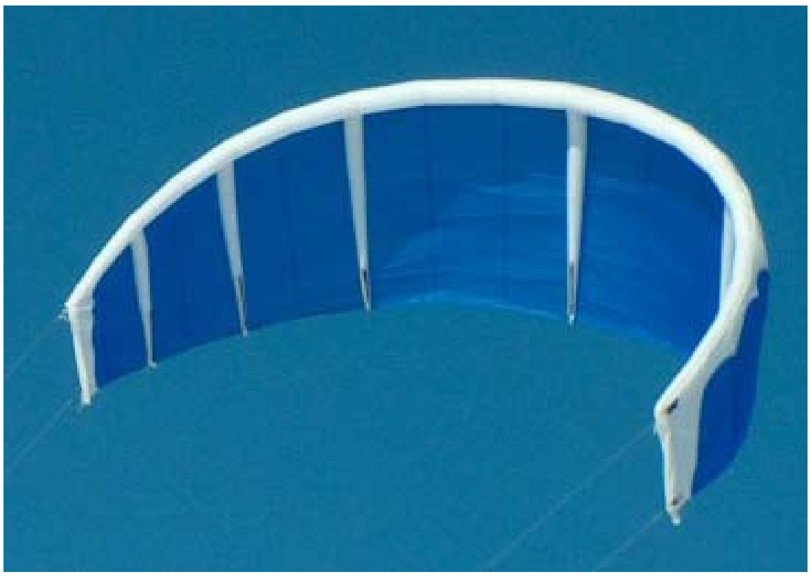
overly emphasizing structural optimization over architectural expressiveness and variation. With integrated digital design tools this potential could be tapped even in less pure forms and surface geometry.

4.2.1.11 Puzzle surface

One way to tap the potential of surfaces is in overcoming the

The puzzle surface approach that uses locally geometrically adapted joint details to connect surface strips via pressure fit.

dimensional standard and moving towards a negotiated one. To define standard on geometric terms should be replaced with a part-to-part adaptation that fits the necessary aspects of the parts to make them interface correctly for whatever role they need to interface in. This might be a structural, compositional or aesthetic interface, or one of surface continuity. In a generative approach combined with fabrication machines, it may be possible to overcome the notion of standardization established in the industrial revolution and return to a more design-based resolution of detail. A big challenge in achieving this goal is the management of the resulting variety of parts and the robustness of the interface definitions as the range of functioning connections is far more challenging to test than a single constraint case. The jigsaw puzzle surface shown here was produced using a simple generative script that translates the local geometry of the design surface into connecting interlocking pieces on both sides of the strips, matching across the connection ridge. The performance of such a connection is dependent on the material and fabrication approach used as well as on the local geometry condition. Increased curvature across the ridge line may



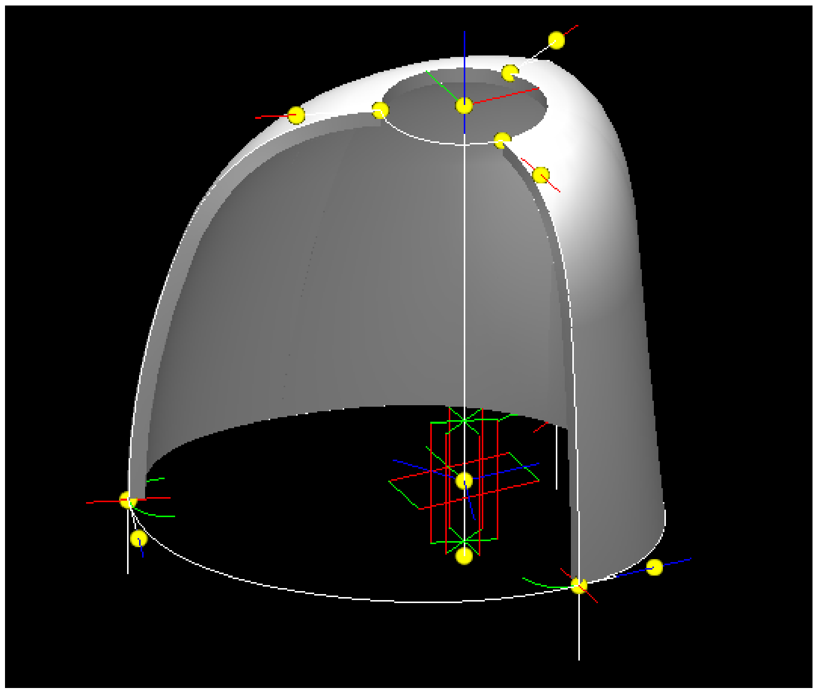
Kite surfing wing as an example of the geometry of a design being subjected to various forces that affect its actual shape. This literal performative example has many variants in more abstract terms. Form is subject to change both in abstract and literal ways from various constraints. An equilibrium is always temporary. Kite by the author in test flight.

lead to curvature changes from strip to strip that are too large to handle for the joint. In that case, the interface negotiation has to control more than just the local interface and influence the overall arrangements of strips on the surface.

4.2.1.12 Performative surfaces

A surface itself can function as the performative envelope, as shown in the example of a kite used for kite surfing. Here the surface is not used as the carrier of secondary geometry, except for the air frame, but is the performative surface itself. Its profile and overall shape are not only controlled by the geometric pattern that defines its default shape, but by the very forces it is generating as a wing during flight. The relationships are quite complex as all factors influence all other factors. The profile affects the lift and the drag. The drag and lift affect speed, which in turn affect the profile again and result in additional forces going into the overall arch. The kite arch finds its form in flight, and that shape is only remotely related to the design shape in the geometric program. This interdependency is very apparent in the kite example as it is determines whether the kite will fly or not, and is resulting in

The principle of design surfaces as hierarchical control mechanism.



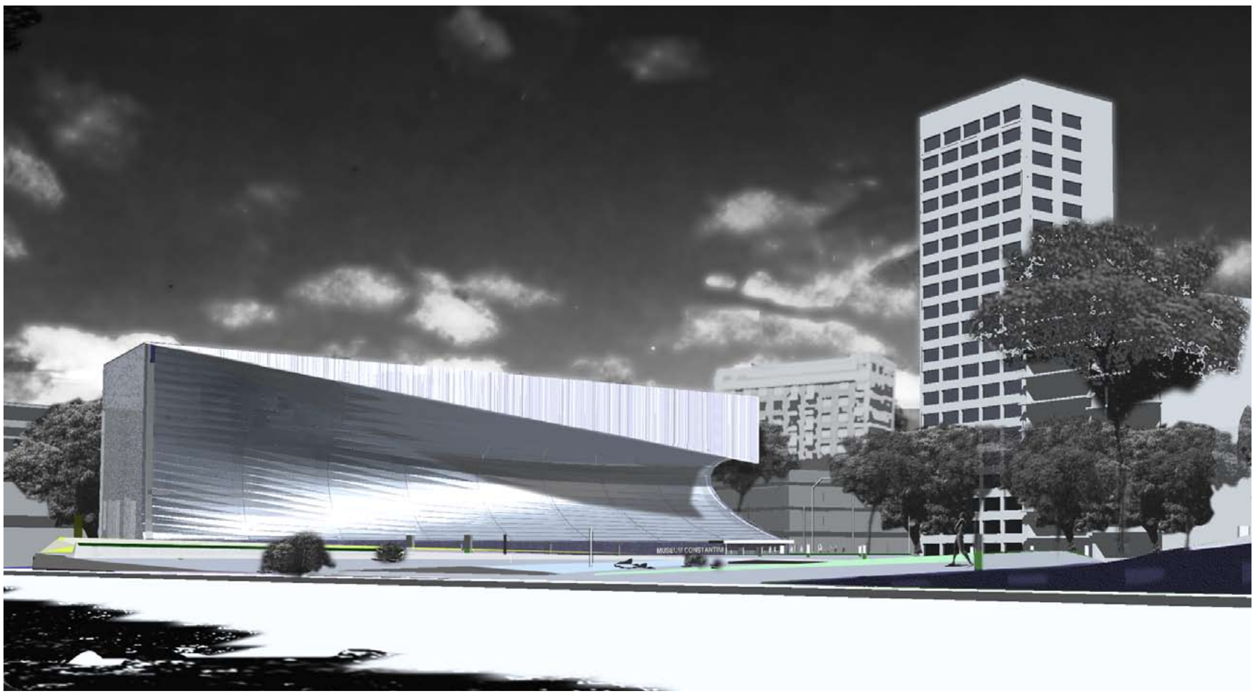
a visible change of shape. It makes clear how critical it is to test an assumption in practice to evaluate its validity. Teaching one's findings add another level of evaluation. At architectural scales, this deformation and the cross dependencies are less obvious and not all linked to the shape domain alone. But a key element one can take away from the kite example is the idea of a design surface.

4.2.1.13 Parametric descriptions of a design surface

Parametric descriptions of surfaces introduce the notion of a design surface that carries secondary design features. A hierarchical parametric construct controls different levels of parametric components that control the overall composition of the shape, the parts and ultimately the components itself.

4.2.1.14 Layered parametric constructs on design surface

The integration of multiple parametric dependencies into one parametric surface component is one approach to deal with the complexity of a design. In this example, the component allows the response to sunlight and gravity as well as to the local geometric context.



Surface as design driver in architecture. Competition project for Museo Costantini in Buenos Aires, competition entry 1997.

4.2.1.15 Design surface as an architectural design approach

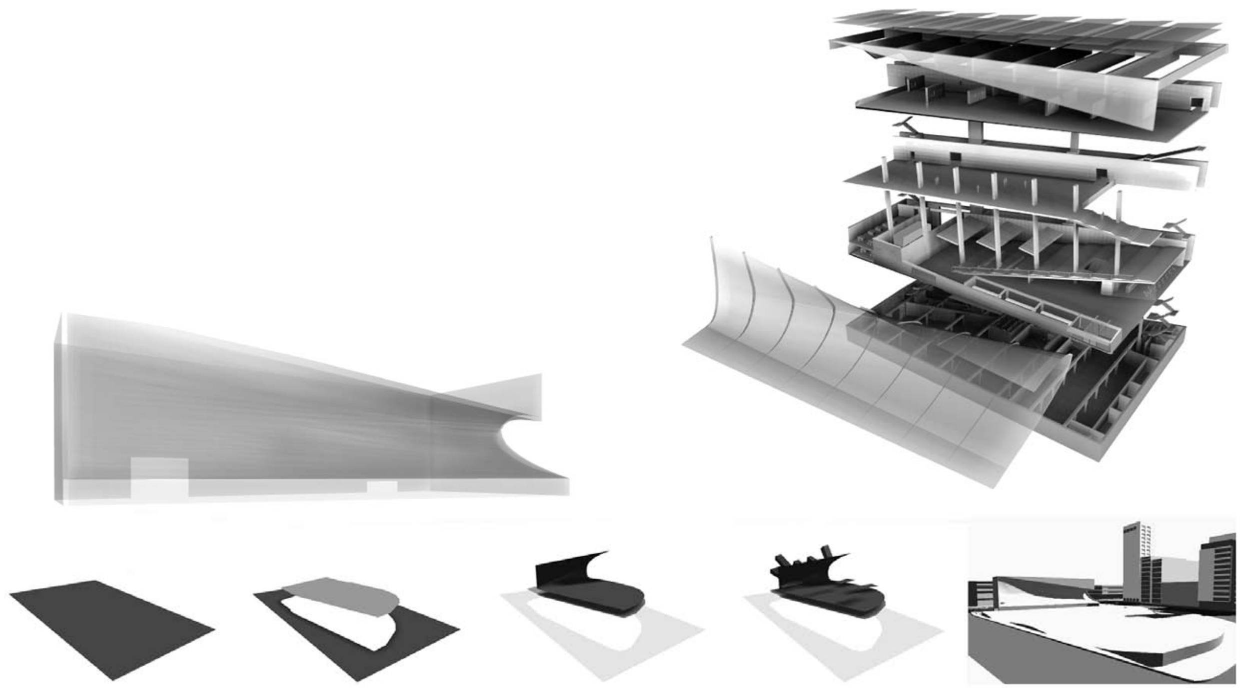
The notion of a design surface has become a prevalent design driver in architectural design. (Venice BIENNALE 2004, Surfaces) The competition entry for Museo Costantini in Buenos Aires was premised on a building-sized wave as a design surface. The curling up of the plaza to a wave allows the capturing of the urban plaza with the modest scaled museum on one end and the open plaza at the other. The surface acts as a sign and as the program interface between the public space of the plaza and the museum spaces.

Representing such large-scale design surfaces and the resultant volumes has always been a challenge in architecture. Volume is rarely solid, but rather implemented through enclosure. Slicing these volumes into cross sections is a common approach in

The project was exhibited at the 2004 Venice Biennale in the category "Surfaces".
Project credit: Axel Kilian, Juergen Mayer Hermann, Bettina Vismann

la Biennale di Venezia

translating free form volume into buildable form. The example of a popular dinosaur model clearly shows the approach, if only for a skeleton. In contrast, the volumetric model employed by another dinosaur toy, although more technically advanced in its production, does not result in more expression than the abstracted, sliced one. An advancement of the technique alone does not mean that there has been any advancement in the overall design. This becomes



The veil like wave surface bundles a program of ascending exhibition spaces. The urban concept develops the building out of rolling up the public plaza and activating the public space through the monumental wave. The exhibitions stack behind the face of the wall as if responding to a ground swell.
Project credit: Axel Kilian, Juergen Mayer Hermann, Bettina Vismann

clear when looking at the sculpture by Naom Gabo, which with a minimal use of surfaces pieces captures the essence of the volumetric expression of the portrait. Construction is not only technique but expression, not only structure but representation of an idea.

4.2.2 Surface strategies

There are different approaches to integrating material properties into geometric surface representations. They are:

- Realism. Realistic behavior is the goal of engineering applications. Realism refers here to the structural behavior of a material, like its bending resistance, tensile and compressive strength, or the dampening characteristics. Although those parameters are crucial for engineering considerations, they are not necessarily fundamental to a material-based design approach.

- Abstraction is a design-specific, approach to material which tends to be very selective in terms of material properties. Quantitative accuracy may be of less concern than qualitative components of the material studied. This might be its texture, it's bending resistance, the way it sounds, feels or its grain pattern appears. In short, the representation of the material might be any number of abstracted properties of a very large set of materials, which makes it hard to provide a holistic simulation library.

Early experiments, started by the author in 1999, were based largely on the abstraction model. The experiments involved programming spline surfaces based on eight control handles that provide incoming and outgoing slopes for a spline surface patch. The surface patch was used as the design feedback object. The surface provides straight forward visual feedback on the spatial state of the object. The motivation was to implement a physical sketching tool for physical three-dimensional surfaces from the combination of a programmed interface for surface generation in Autolisp in AutoCAD, in connection with the laser cutter as the physical output device.

The goal was to work with this combination alone and capitalize both on the generative potential of programming and the flexibility of the fabrication equipment in precisely cutting any flattened geometry. From my experience in fashion design and sewing from a decade earlier I was very familiar with the concept of pattern making and the assembly of spatial constructs from planar surfaces. In this spirit I approached the problem and followed an isocurve-based subdivision of the spline surface into single curvature strips. The Autolisp program unfolded the resultant surfaces through a triangulation procedure to the main plane and equipped the strips with a jigsaw puzzle adaptive detail.

The idea of the jigsaw like puzzle was twofold:

One, it provided an assembly logic which would register the edges of the corresponding surfaces strips precisely against each other without any measurements or particular care required.

Second, it implemented a connection which requires in the ideal case no additional fasteners but relies on the capabilities of the programming-fabrication link in producing the connection geometry as an integral part of the production, not unlike the traditional Japanese wood joinery.

Both goals were fulfilled in several of the prototypes, although the connection part was susceptible to failure depending on the cutting tolerances and the material and geometry of the shell.
This cross dependency initially seemed like an undesirable side effect of a not fully described and controlled, i.e. abstracted system. The material properties had not been made part of the generative process and therefore deformations due to the resistance of the material could potentially deform the joints in a way that they would fail.
Several attempts were made in trying to optimize the geometry and to make the behavior of the assembly more predictable. Variations in material, geometry and fabrication techniques allowed a more rationalized view of the cross dependencies. The factors are cutting depth, cutting surface related to the fabrication process, material thickness, material grain, and geometry of the overall target shape, scale of the overlap of the joints, local curvature conditions at the joint.

4.2.2.1 Feedback and cross dependencies

This condition points to a feedback effect between designed geometry and actual geometry of a physical assembly. Most building systems are carefully designed to minimize the cross influences between different components in order to avoid unpredictable results. This conservative strategy makes sense in a practice divided by discipline, which is overwhelmingly the reality of today's construction. Traditional craft, with its less complex projects at a more manageable scale, takes much more advantage of the secondary effects between material and design for instance. For instance, weaving a basket can be a highly precise task. But the positioning of the strands with respect to the targeted form will be variable as a response to the local and global material properties in connection with the weave and the fluctuations of the natural strands used. So here detailing is a process in constant adaptation to a desired goal. It cannot be planned out a priori as geometry,

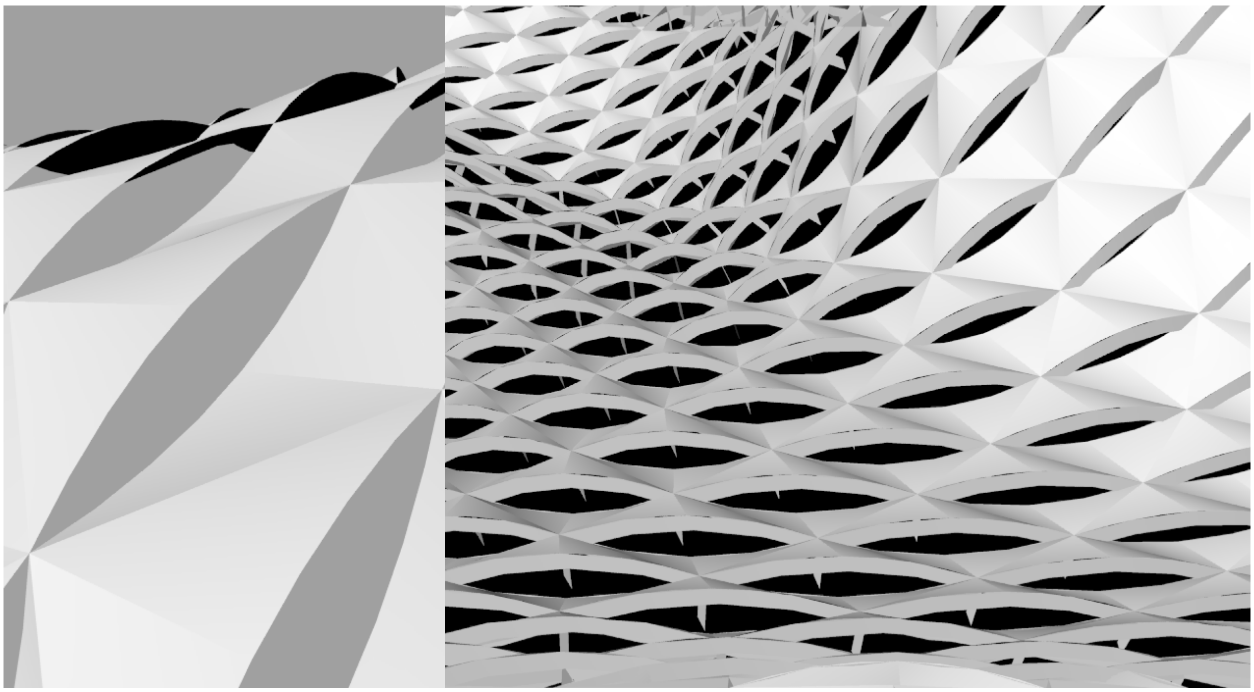
but rather is the result of as sequence or pattern, which essentially equals the topology of the weave.
How is it possible to get to such a level of fluidity in architectural design? The tactile, manual labor of basket weaving does not easily scale and therefore cannot be literally applied to a large project. The idea, therefore, would be to find a description which is responsive to the circumstances it is applied in, rather than to create a scaled version of a weave. These would be components that are created on the fly based on their geometric context, or which have enough degrees of freedom in their assembly to accommodate for any deviation from the target shape.
The thesis develops a number of approaches that replace the tactile manual response in the adjustment of a flexible system with computational, more abstract systems that allow adaptive systems to function by embedding properties in the geometry of fabrication. The first one is the previously mentioned jigsaw puzzle surface approach, where the translation of the geometry happens through the application of a joint detail.

4.2.2.2 Control of the surface

The example shows a corner based tangency controllers approach for the surface patch. This is a light weight approach to surface control that offers robust interaction and fair curved surfaces due to its low degree surfaces. Due to its four control points, the curves have order four and degree three at a maximum. The implementation is done using degree two, with a quadratic polynomial equation.
Achievements were the linking of generative programming with fabrication with a material sensibility and assembly procedures implemented.
Problems were the propagations of deformations through the assembly potentially causing joint failure when the parts were being assembled.

4.2.2.3 Control of a complex geometry with embedded properties

Parametric surfaces can be controlled and generated in a variety of ways. The control mechanism and the structure influence the possible results. Similar to any complex device the controls are ideally high level and have a certain level of abstraction to allow for control of a complex entity through few parameters. Below the abstract top level of control can be a set of controls with finer granularity, which regards sub parameters of the surface. Similar control systems can be found in kite surfing control rigs where there are primary controls that ensure the shape of the tensile structure is stable and the load transfer takes place and on top of that.



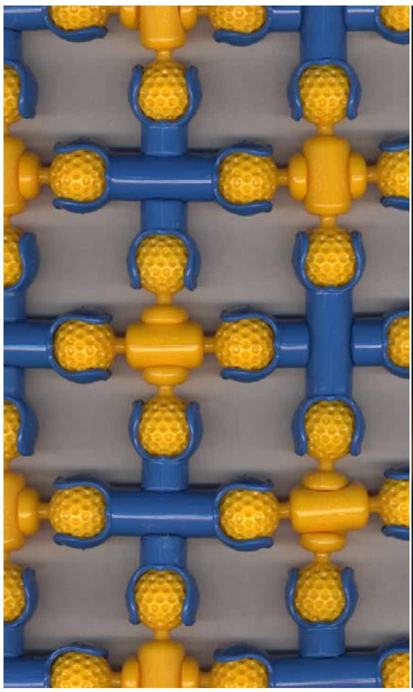
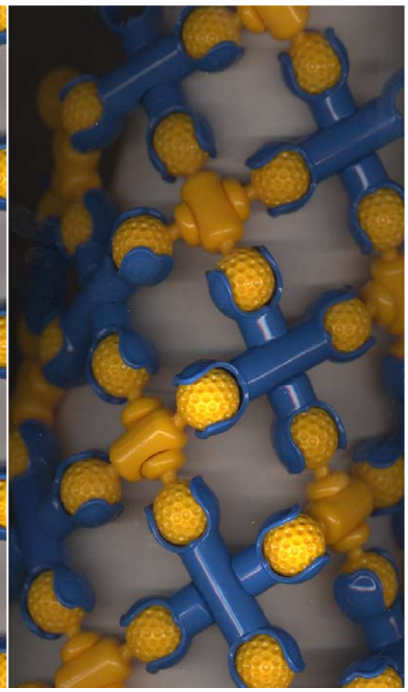
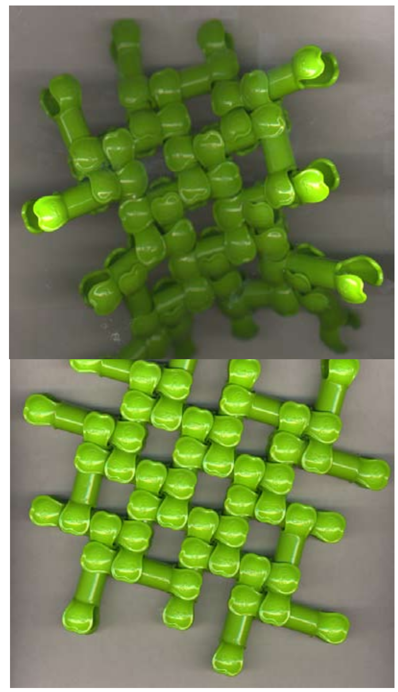
Surface based louvers responding with variable aperture to the global lighting condition and the local surface context.

Reigns for a horse are a similar case. The reigns don't control the horse directly, they just provide the input for the horse to interpret the situation and act accordingly. Of course, these are far more complex organisms and mechanisms that the average parametric NURBS surface.

Other models that were developed and explored are

- Hierarchical control rigs for surfaces
- Surface growth through Lindenmayer systems (through contributions of the author, and a project by the ED Group headed by Peter Testa; as well as MOSS by Testa, O'Reilly, Kangas, Kilian)
- three-dimensional printing of volumetric assemblies

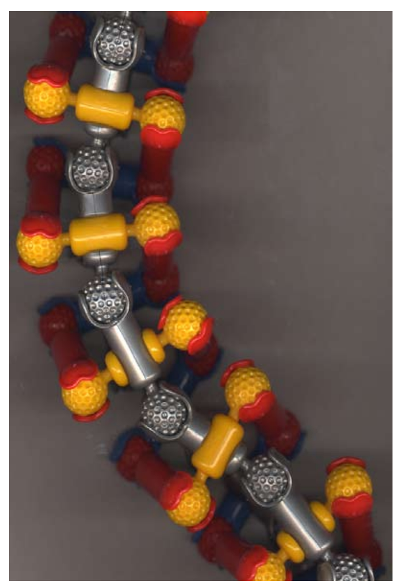
4.2.3 Surface controls – the design surface

Bottom up component based control of surfaces. In this case the zoob© system was used to construct part based meshes that respond within the constraints of their movements to control input.

principle

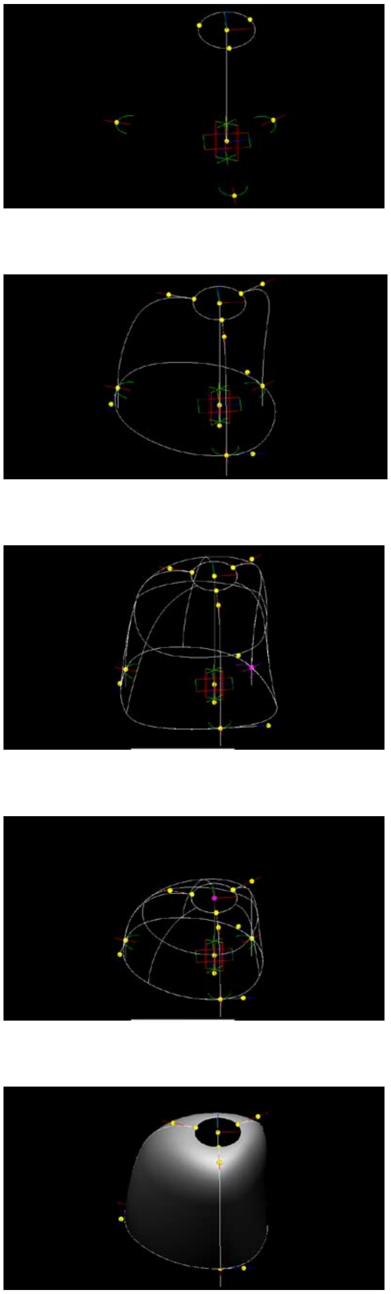
The following examples discuss how a wire frame skeleton similar to a control polygon can serve as a robust and visually traceable control mechanism. The control rig externalizes the dependencies in their geometric context.

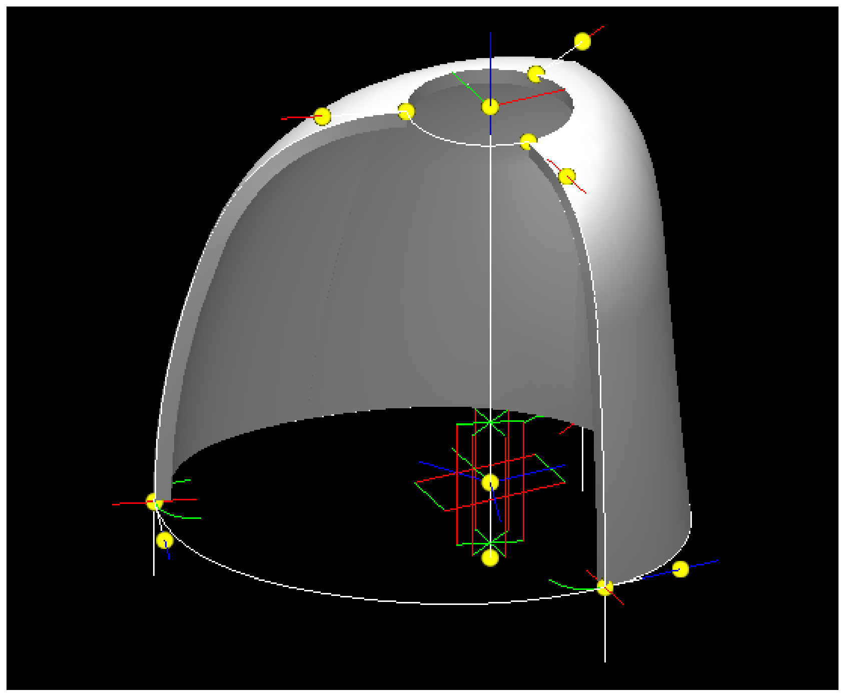


4.2.3.1 Experiment 1: Component-based surfaces, bottom up control

The ZOOB© toy was used to construct surface grids with bottom up control. The surfaces that can be are a function of the degrees of freedom of the components and their assembly patterns.

A component-based surface emerges from the assembly of elements. The exploration range is very limited, but results from local part interaction and affords a certain global redundancy of control from the multitude of parts working together to form the whole. The degrees of freedom of the assembly logic determine



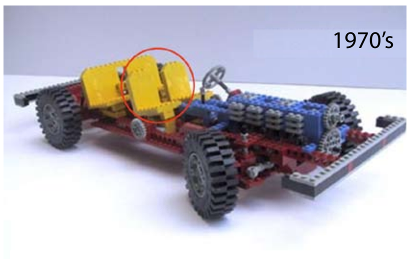


the possible shapes in a bottom up rather than a top down fashion. Digitally, this sort of assembly requires a constraint solver for the non-hierarchical modeling of the unit interaction, collision detection, and for imitating the movements. The example was used in teaching to explain constraints of system based building components and as an example of assembly based construction.

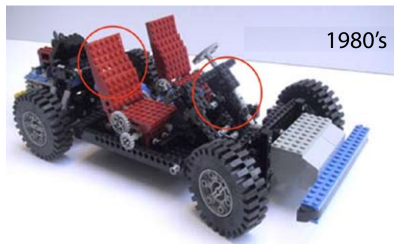
Top down surface control through a control polygon.

4.2.3.2 Experiment 2: Hierarchical, parametric, top down control

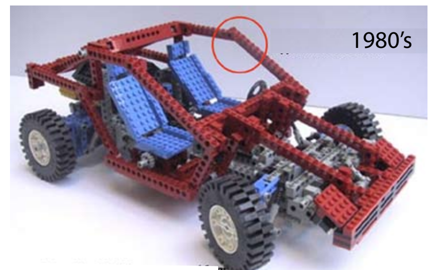
This example shows a control rig for a NURBS patch surface with tangency continuity across the control wires. The topology of the first order control polygon defines the features of the surface as well as the top level proportional controls governing height, positioning and relations of the parts to each other. On the one hand, this approach is very limited due to its downstream dependencies of all subsequent parts. At the same time it provides a very robust, almost sketch-like top down surface definition.



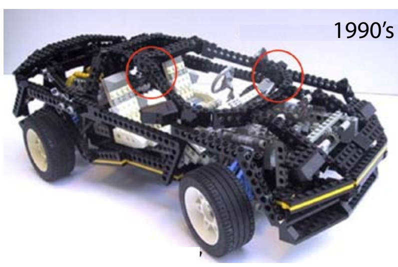
Technical abstraction of the idea of car conforming to the system"



Introduction of pixelization of non orthogonal elements



First attempt to representing form and not just function



Polygonal use of building elements to approximate curves

Fading of the block systems in favor of a shape derived post rationalized one

Shift in granularity over shift in system for approximation

The evolution of the LEGO super car series from 1970-2000. Over the years the models increasingly depart from the simple block system towards a post rationalized building system. Images LEGO.com and http://www.nd.edu/~lego/grp2/www/techlst2.htm

4.3 System

Systems are emergent organizational entities that go beyond the ad hoc decision making of improvisation. A system design has a holistic design goal and self-referential dependencies among its parts. In a system there can be agreements between parts without there necessarily being a direction connection. Parts are compatible by agreement to conform to a system. This is true not only for physical implementations, but reaches into software systems as well. Systems also have a certain self-perpetuating dynamic. Once they exist they tend to grow and incorporate more things not part of the system originally. Standardization is a good example of a system of regulated norms that ensure compatibility. Building systems in architecture have a long tradition. System standards for bricks evolved first before being formalized.

Of more relevance for the subject of the thesis are the cross dependencies between building and design systems and design goals. A structure built in brick will have a different structural language than one built in steel. It is crucial for design innovation

What is innovative?

The intelligent use of the system within the constraints?

The creation of new parts that extend and eventually compromise the system

O

Or redefining the scope of the problem?

Balance between the system constraints and the design goal. System inherent form language versus design language of the car.

to prevent systems from becoming conventions which prevent the continuous adaptation and reinvention of the agreements and norms based on the evolving context. Design exploration benefits from generative systems such as software programs or programming languages. But these systems have their own system-inherent limitations that need to be understood in order to avoid them.

4.3.1 Design goals and system dependencies

To illustrate this interdependent relationship between a system and its design goal, car building kits produced by the toy system Lego are documented on the basis of advertisement images launched between 1970 until 2000. The initial block-based Lego system allows only for mostly cubic shapes to be assembled from its standard units. The car model is representative of the idea of car as a minimal mechanistic block rendition of real cars of the day. Over the decades that follow, the block system gets increasingly more diverse. The design goal focuses towards mimicking shape rather than mechanical abstraction. This reveals the inadequacy of the block system for free form cars even more. Eventually more and more special form elements are introduced, to a point where the system seems to be a post-rationalized product of trying to implement the form of a car. One could argue that the expansion of the system to better conform to a goal is an innovation. At the same time the elegance of the inter-part compatibility is increasingly lost. The last image shows a shift in scale of the system-design goal relationship, similar to the relation of a brick wall and a concrete wall made of "liquid stone". It is crucial to realize the pressures that are exerted by a system of exploration onto the subject of exploration. The two are not independent and need to be carefully modeled.

4.3.2 Building systems and what can be built

A similar relationship exists between most building systems, such as bricks, steel beams or glass facades, and design. Languages of design develop around systems, such as in the brick Gothic period in Germany, with its highly sophisticated brick variations and façade patterns. The available system influences architects in their designs, and novel designs in turn push the system envelope.

But the system occurrences have by now shifted into the digital realm as well. Every software is a system in its own right and enforces constraints for the user and provides tools that make certain design moves easier than others. Foster and Partners follow an approach in their specialist modeling group where some project-specific tools are provided to the designers that allow only the design of rationalized design geometry, such as fair curves and surface patches. This limitation is not necessarily perceived as an obstruction as it is implemented through parametric constructs that can only be exercised in a pre-rational manner. This approach ensures that even in larger teams rules about surface qualities and parameters don't have to be enforced or checked but can be assumed to be met based on proper tool use.

Frank Gehry and Partners follow the opposite approach. The design exploration is not conducted with tools rationalized a priori, but the outcomes of the design development are rationalized later in a so called post-rationalization step. Building systems are either developed or existing ones adapted together with slight adjustments in the geometry to achieve an economic solution and make the design buildable. The truth is, of course, that these two approaches are far from pure. Accordingly, Gehry Technologies has devised tools that support the translation of the design processes most common in the office, such as cardboard-based surfaces.

4.4 Search

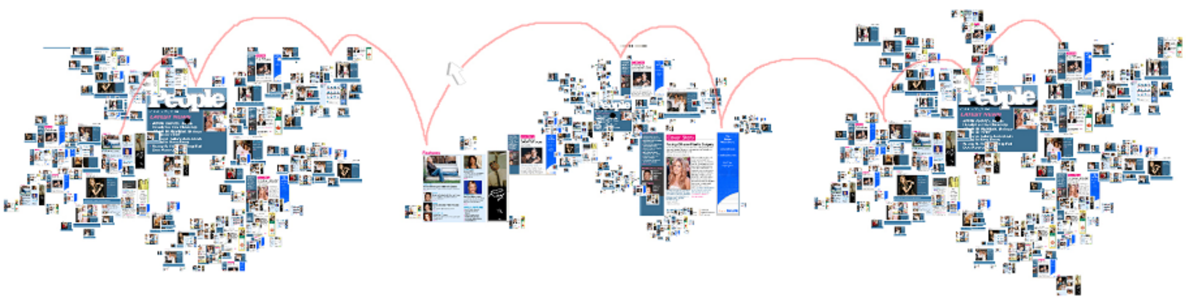
Frei Otto supposedly once said: "I do not design, I search". This is a powerful, statement, and at the same time a very honest one. If one truly is looking for novel design approaches, it is unlikely that they can be designed in a predictable manner. Design emerges through the increase in understanding of a design problem and its constraints. To search means to ask the right questions and formulate the framework accordingly.

Strategies for search can be embodied in design explorers. Most importantly, the ideal search is generative. Rather than revealing what it is already known it will create novel findings from known starting conditions. This expansion of the reference frame rather in place of the exhaustive sifting of its content is what separates design as search from combinatory evaluations. To accomplish such a search is a major challenge. The fact that constraints are present does not provide definitive answers in terms of design solutions. Any design proposal that satisfies the given constraints may still not fit the design intentions on other accounts not included in the initial description of the exploration. This is very common in design and in fact crucial to evolve the design target with the investments made in the exploration process.

The search section does not go into depth of more sophisticated search processes in design, but rather illustrates a few experiments by the author as representations of possible approaches. A large area of data-based search is omitted, and the examples presented here are not representative of the universe of search procedures.

4.4.1 Visual search –keeping track of design history

An exploration into the tracking of design history is the multi-branched browser history application. It allows browsing expanding nodes of a tree-like system based on time and location of interest. The amount of time spent on a node is the scaling factor in relation with the other nodes. This allows one to quickly find the nodes that were previously viewed more extensively. In the example



shown, the nodes are web pages of People Magazine. Static trees have been used extensively in the visualization of data structures, even in design software context. The introduction of a multi-branched design history that is dynamically weighted based on the importance or perceived importance of each step is a potential improvement. It still leaves some open questions however. The example shown here is only a sketch of a possible approach.

Exploration creates a visual history. Attention based scaling of the visual content creates a record of usage patterns.
http://destech.mit.edu/akilian/newscreens/people/browse2.browse2.html

4.4.2 Parametric control objects for design exploration

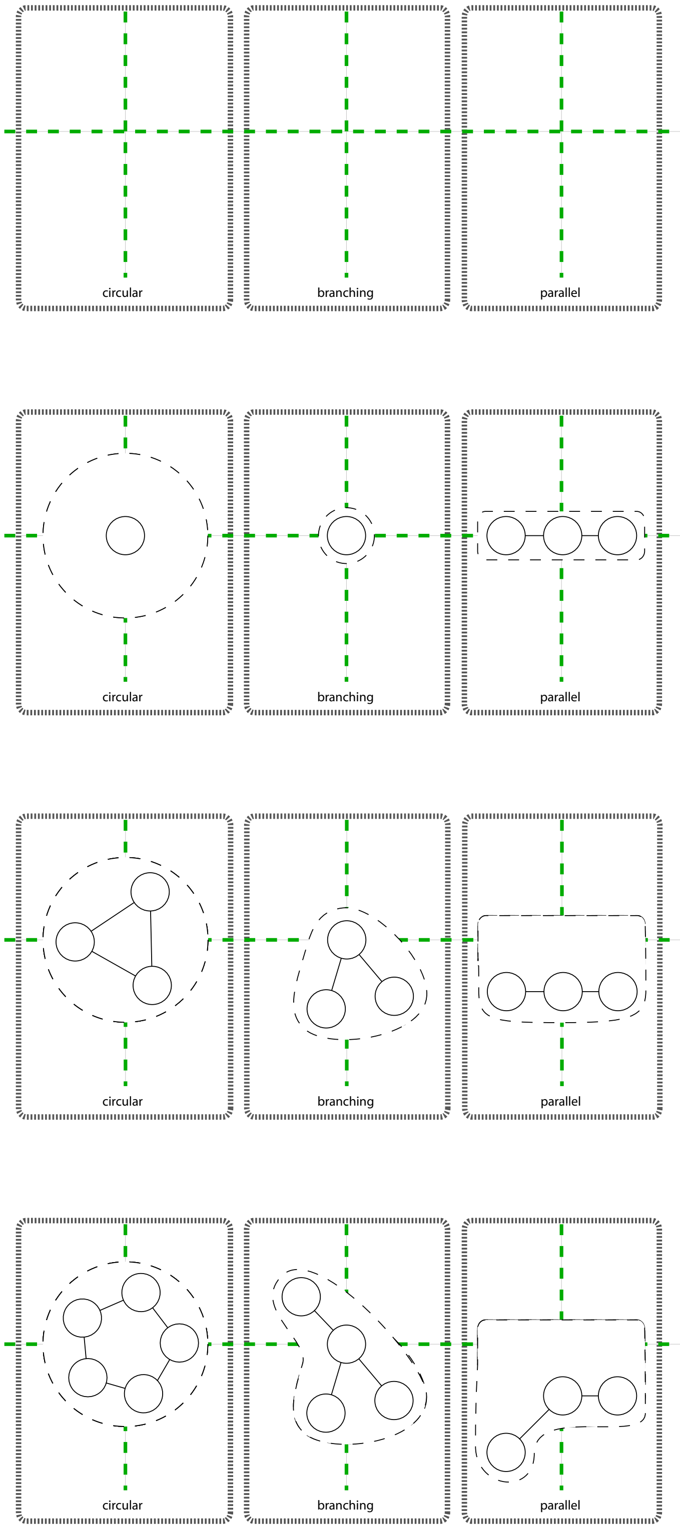
Designing with parametrics poses the challenge of evaluating the design range of possible outcomes presented by the parametric construct set up for the exploration. Higher numbers of parameters make the interaction with the design construct less intuitive.

The example shown here demonstrates a geometric control object to provide parameter setting interpolation and the possibility to record and memorize states of the parametric settings for the object.

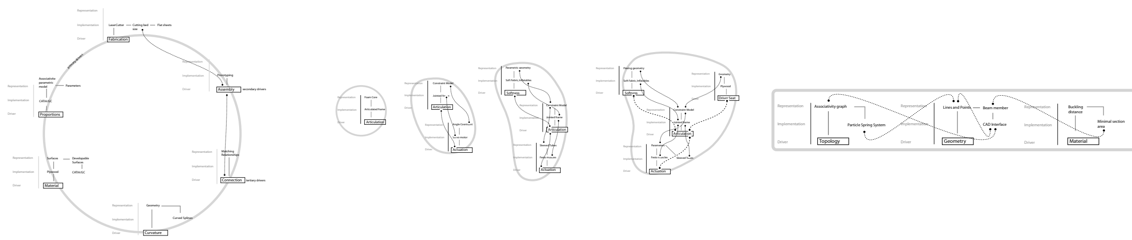
The example shows a family of objects whose parameters are mapped to a grid of points that sample a surface based on regular UV spacing. Moving the UV grid adjusts the parameters and regenerates the object family. By increasing or decreasing the sampling rate around the points of interest, one can explore parametric variations in more detail where needed.

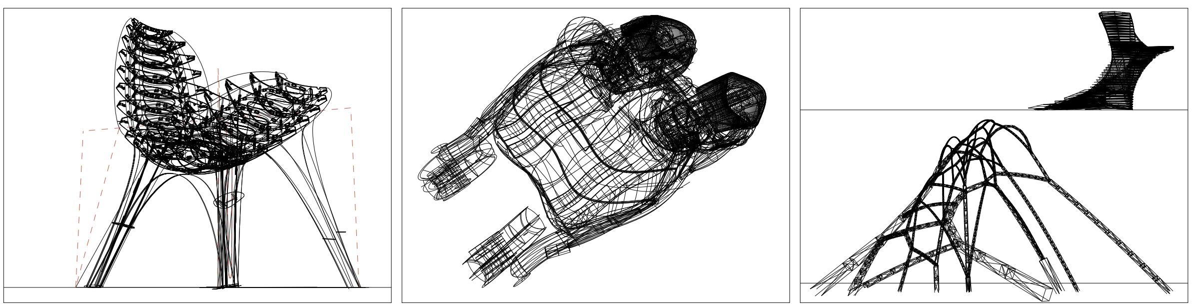
4.4.3 Genetic algorithm approach

Another approach is operating the parameters of parametric models through search strategies, for instance simplex search or



Evolving explorations in the three experiments





Genetic Algorithms (GA). Genetic Algorithms are essentially search algorithms built around a vaguely biological metaphor. But their name is misleading in terms of what the family of GA's is capable of doing. Nonetheless, they provide a robust way of connecting fitness functions to search procedures. Their modular structure and the ability to introduce different translations between genotypes and phenotypes make them relatively easy to adapt to different design problems.

5 The Experiments: Circular, Branching and Parallel Explorations

Three main experiments were conducted in the framework of this dissertation: The Chair, The Car, and the Hanging Model. The experiments are in chronological order of when the problem domain was first addressed by the author. The surface and material based experiments started in 1999, the concept car exploration in 2003 and the hanging chain model in 2004. The terms circular, branching and parallel are used to categorize the experiments. They refer to the properties of the solution space in relation to the design exploration of each problem domain. They do not literally refer to the network topology of the links of the design representations.

A circular exploration, then, refers to the interdependencies of design representations and constraints in defining the solution space. Circular describes the possibility of feedback loops between these representations and constraints, which in turn can create circular dependencies. The design goal of the chair experiment, which falls in the circular category, was to bring material, geometric and fabrication constraints into balance to produce a fabricatable prototype.

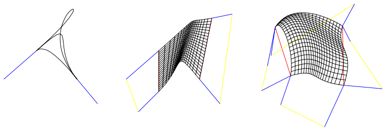
In contrast, the branching exploration focuses on establishing and defining a design problem through connecting different design constraints in a tree like fashion. The main experiment conducted as a branching exploration was the "Athlete" concept car design.

The hanging model is an example of the third type of exploration. The third type, the parallel exploration, finally, refers to a design exploration where the design explorer contains a fully understood set of constraints that can be exercised for design variations. The solution space can expand, but only within the boundaries defined by the constraint network. Design variations emerge from the parallel interaction between the different constraints. The main experiment conducted as a parallel exploration was the hanging model.

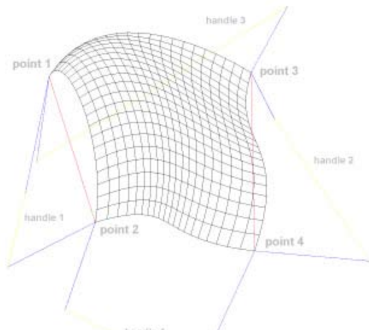
The experiments are presented in the order in which the respective problem domains were addressed by the author. The surface and material based experiments that led up to the chair design started in 1999, the concept car exploration began in 2003, and the hanging model was started in 2004.

5.1 Circular Explorations: Refining the Constraints

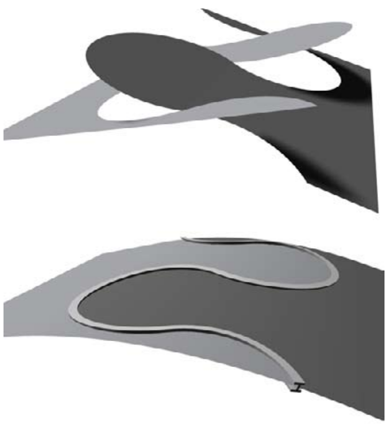
The circular experiments are centered on the integration of multiple constraints with the goal of refining their relationship through design exploration. The experiments are addressing issues of surface and material interaction as well the question of fabrication for free form geometries.



B-Spline patches as the basis of the surface modeler



Eight control tangents define the patch. Implementation in AutoLISP



The joint principle: an adaptable geometric unit creates an interlocking connection that is pretensioned by bending across the ridge.

An assembly of two adjacent strips through the adaptive joint. The pressure fit of the joint holds the pieces in place.

5.1.1 Design Surface Principles

The preliminary sets of experiments leading up to the chair design were all based on the design surface principle. The principle is based on design representations that draw from one main surface as the guide for subsequent components or details. The experiments overlap in part with the conceptual explanations of surface in chapter three. The series of case studies for the first set of experiments investigate the circular dependency networks present when flat sheet based fabrication is combined with curved geometry and friction and assembly fit connection details are used for assembly.

The appearance of computer numerical controlled (CNC) prototyping devices in the architecture studios in 1998 triggered the interest in the potential of such machines in a design context. Simultaneously the author had engaged in a series of classes aiming at learning how to program as the limitation of manual modeling and rendering became apparent. The development of design specific digital environments seemed crucial for the aesthetic development of design. The three dimensional puzzle surface was developed as an attempt to develop a digital physical evaluation tool using programming to develop an interface to manipulate spline surface patches and a laser cutter to rapidly fabricate those surfaces. The goal of the assembly detail was to allow the parts of the flattened surface to be assembled without additional fasteners but rather embed the connection logic into the fabrication of the parts. Another aspect of the links was the to make them unique such that only the parts that belong together fit together like in a jigsaw puzzle and therefore labeling was unnecessary. For the scale of one hundred connections this is feasible, beyond that the labeling makes at least the rough sorting a lot more efficient.

The joint / surface / programming / CNC method raised questions about the notion of an assembly in the context of fabrication and the dependency of design geometry and physical output in the context of freeform surfaces. Most importantly it demonstrated the shortcoming of the established geometry centered digital design model. The digitally represented geometry is not sufficient in capturing the processes involved in fabrication and material forces during assembly, especially not if novel processes such as generative design are employed that open up new possibilities to take advantage of materiality and context specific geometry. (Kilian 2003)

5.1.1.1 Experiment 1: Combining Fabrication with Generative Design – The Puzzle Joint

The puzzle joint experiment was conducted to test the connection of digital modeling with generative programming and rapid

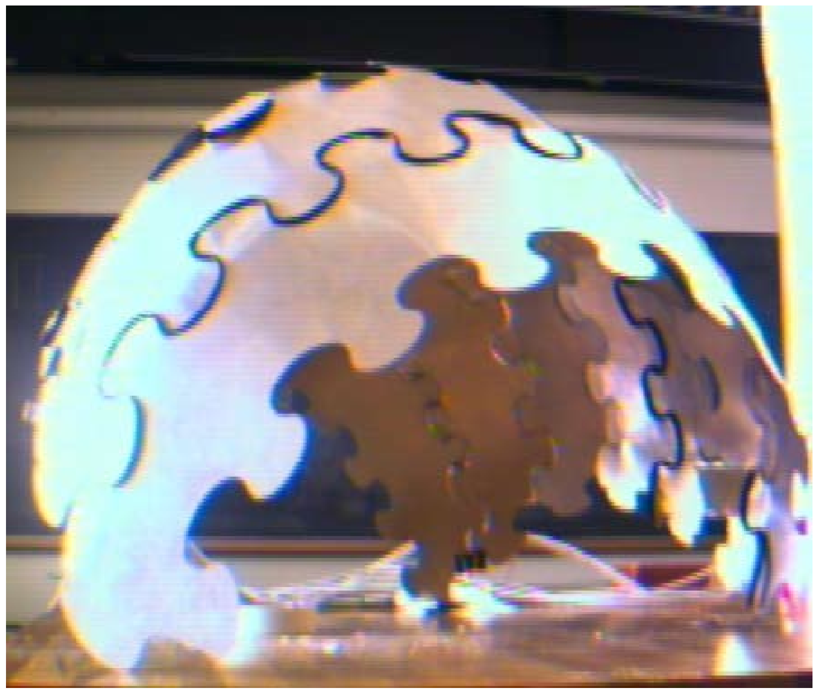
prototyping, to produce physical sketch surface models. The physical surface models were assembled from developable strips connected through a puzzle-like detail. The use of programming as a design approach allows the generation of connection details that correspond to the rules of flat sheet rapid prototyping techniques of laser cutting and water jet cutting. With numerically controlled cutting there is no need to keep the joint detail related to manually achievable forms or to apply a standardized dimension. The goal of the experiment, then, was to demonstrate the possibilities of programming to generate cutting geometries that adapt to the local surface properties. The fundamental questions were how to formulate and capture design intention through programming, and to investigate the influence of the use of generative modeling

Puzzle assemblies in different materials. Material and fabrication technique affect assembly and friction fit. Geometry has to adapt to material and fabrication constraints.

in combination with rapid prototyping on the design language of physical objects

Rapid prototyping and CNC machining tools are increasingly making their way, not only into production, but also into design schools. The machines are posing new challenges, not so much in their ability to execute drawings done on a CAD system by a process very similar to drawing, but by their ability to cut any geometry

An aluminum assembly with pressure fit rubber strip. Early waterjet aluminum cut model 1999..



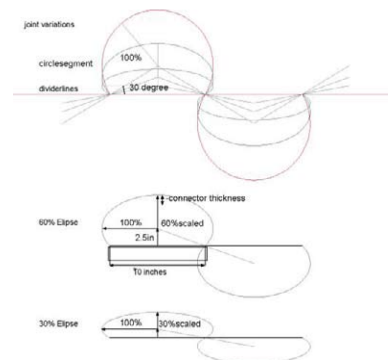
within the limitations of the machine. The question is how to expand the use of such machines and explore their potential. Two possible approaches are

- Extract the capabilities of the machine and embed them into a generative program that explores the possible forms and cuts within the limitations
- Design an object and adapt it to the machine's capabilities

There are always shapes for which a particular fabrication process works well and other shapes that need to be redesigned in order to be fabricated efficiently. As in conventional craft and manual production processes, there are "easy" and "hard" procedures in CNC. In this sense, CNC machines do not differ from the tools and processes involved in a conventional craft. What does differ is the potential for the creative reinvention of details originating in conventional craft in a CNC process. This can be accomplished through generative techniques where a customized solution for each detail is produced. However, few designers have explored these possibilities.

Fabrication and language of details

The language of a detail results from the combination of material, required performance, design intentions and its manufacturing process. If any of these parameters change, there is potential for a new detail to emerge. So far, few specific details have emerged from the use of computer controlled machining. In this experiment, the author proposes an adaptable detail over a curved surface that would have been very hard to produce in any way other than through the combination of computer-controlled fabrication and generative modeling.



The joint principle showing the variable parameters controlling the curvature and alignment of the joint parts.

Tolerances, between pressure fit and constructability

The process of fabrication allows for the specification of very precise cutting dimensions. In this experiment, the precision is crucial in order for the joint to be a pressure fit joint. But too high a precision in the fit will prevent the assembly of the joint: due to its spatial curvature, the continuous detail cannot be assembled all at once but only sequentially. The sequential assembly requires larger fitting tolerance to allow the pieces to move into place. The challenge is to find the right balance between a tight fit that would cause problems in the assembly, and a looser fit and that might cause the pressure fit joint to fail. Ideally this would have to be modeled into the geometry generation in the first place rather than to compensate for the variations through the cutting tolerances during fabrication.

Ability of details to adapt to their geometric context

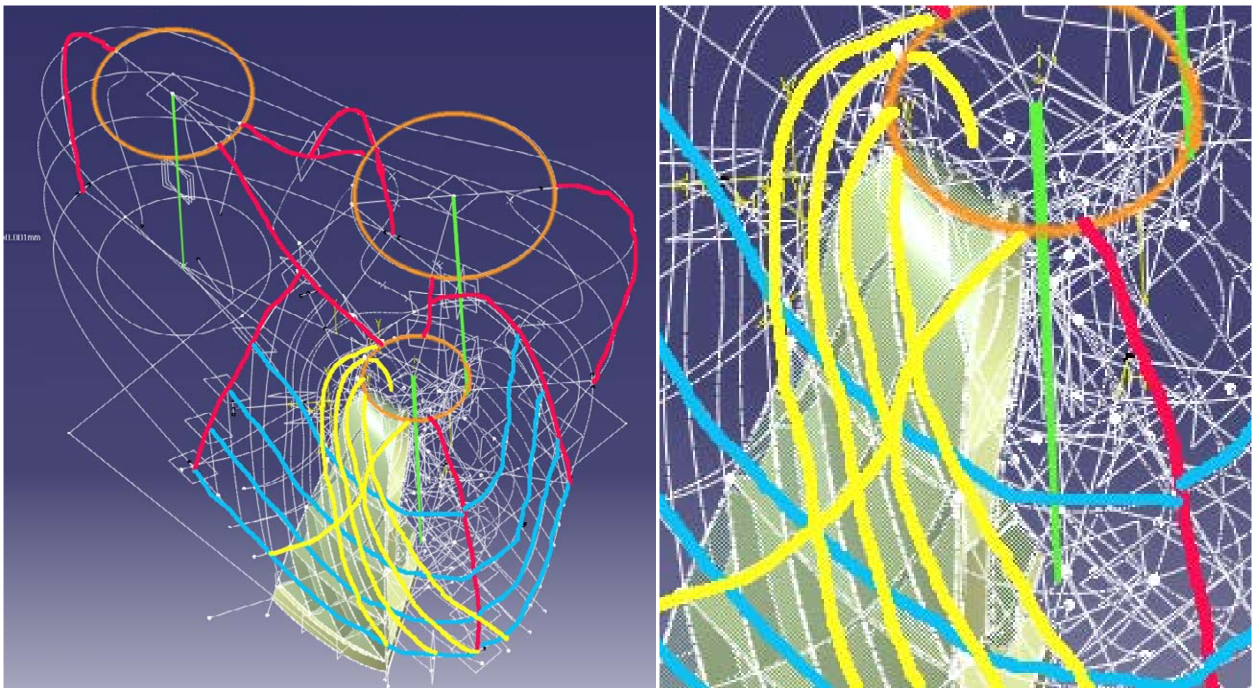
A connection detail for a varying geometric context needs enough flexibility to work in the complete range of scenarios. Many experiments have been made, for instance in textile and membrane design, to come up with adaptable details to correspond to the geometric context. Assembly details in car design often depend on geometrically adaptable connections. For instance, a door seal follows the curved rail of the car body and the frame between the double-curved windshield and the car body describes a spatial curve. Such geometrically adaptable details are nothing new. What is novel though about the detail presented here is that its geometric dimensions vary based on the context and are manufactured for that exact location. A further extension of this approach is to allow different variations of the detail during generation based on conditional checks of the local context. For instance, if a curve becomes too extreme for one type of detail, a different type might be used.

Changes in standardization

With the generation of geometry through scripts that allow for parameter-based variations, the notion of standardization shifts from dimension based descriptions to a topologic system of relations. It is no longer necessary to fix a certain dimension at the interface between components, but the relationship of the two parts defines the shared dimension. The standard shifts to the description of the relationship. It is important to know how elements relate to each other, but it is not necessary to know their absolute dimensions. Parametric modeling has set the stage for the expression of elements as a set of relations that have variable dimensions. A non-dimensional based standard relies heavily on fabrication techniques that allow for the production of varying geometries based on a system, as in the example of the joint shown here. Generative approaches through programming allow the generation of complete systems out of a set of rules. Robert Aish, Director of Research at Bentley Systems, has developed a very promising approach to programming integrated into parametric modeling systems in the Generative Components extension to Microstation.

Self-registering geometries

When using manufacturing techniques that require post-fabrication assembly, it is very helpful to have self-registering geometries that allow for exact alignment of pieces in space. The puzzle joint is one approach to this problem. The continuous curvature of the joint provides a continuous fit between parts. In addition, it allows, in most cases, for only one, unique assembly of the pieces. In order for the pieces to snap together, they have to take on the desired three-dimensional shape. This ensures proper alignment of the pieces and the approximate three–dimensional shape of the overall form. However, inaccuracies result from the transformation of the digital geometry into the physical representation, as the current digital models do not take into account material properties, as for instance resistance to bending and material stiffness. When, in the case of the joint, one node forces a partial buckling of the



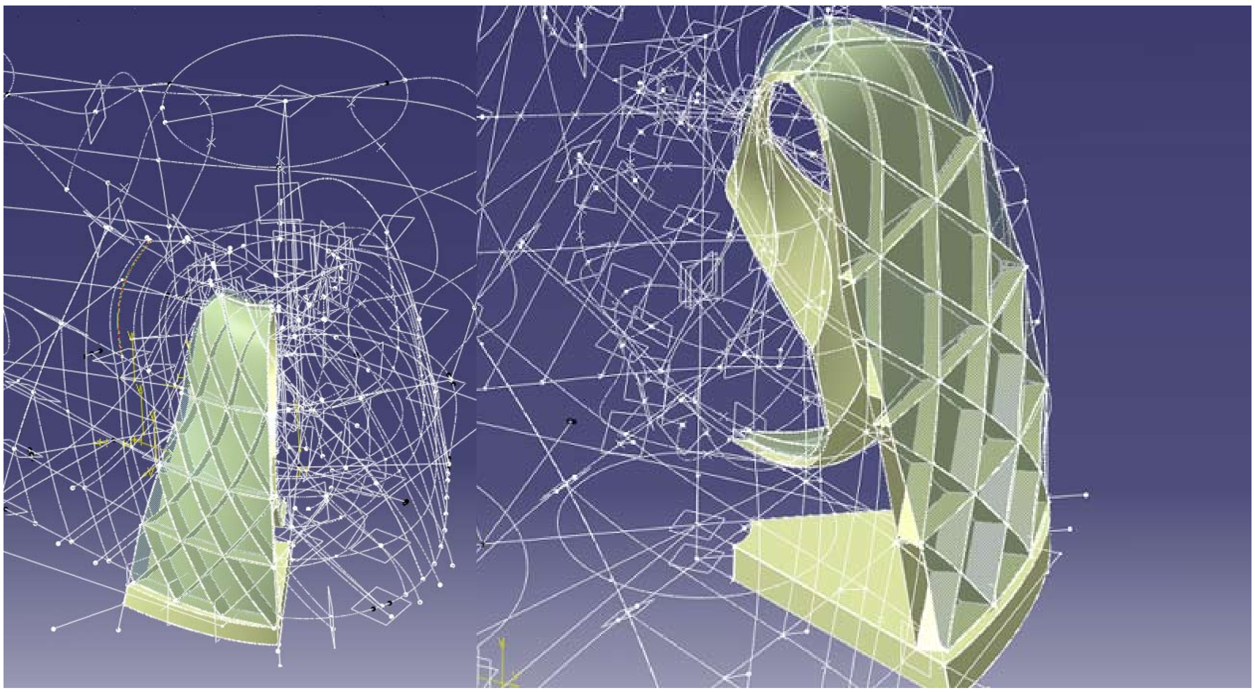
Reading pavilion project in digital mockup class 2002.
Team: Axel Kilian, Carlos Barrios and Tim Morsehead. Parametric model in CATIA showing the development of a adaptive detailing principle.
Image: Axel Kilian

surface, it affects the neighboring nodes as well. In the worst case, the propagation of these deformations renders the assembly of the pieces impossible.

5.1.1.2 Experiment 2: A nested, hierarchical, top down surface control skeleton – the Reading Pavilion

The project was done by the author in collaboration with Tim Morsehead and Carlos Barrios in the context of a fabrication workshop co-taught by Mark Burry, Dennis Shelden and Larry Sass in 2002. The pavilion is an example of a nested control structure. High level controls propagate through the surface all the way to the implementation details.
This pavilion offers a robust solution for the fine-tuning of

proportions and general layout adjustment even when the design is already fully detailed. The goal was to model and define all details such that for changes in the systems all parts would still conform to the material limitations and the assembly logic. The author developed a CATIA model that incorporated a hierarchical control skeleton. It allows layout changes of the floor plan as well as the proportions of the envelope and the skylights. The model allows

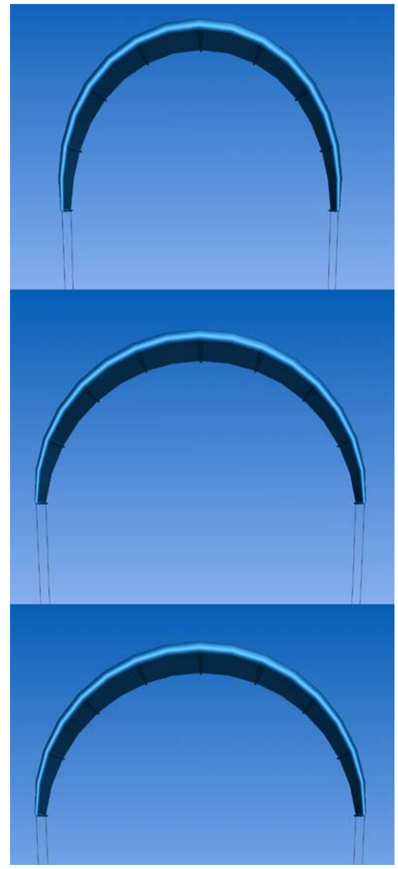


The hierarchical control grid for the pavilion. It allows the high level edit of the overall layout of the pavilion through the positioning of the floor plan. Image: Axel Kilian

changes to propagate through the design surface all the way to the skin sandwich details, which update accordingly.

The surface is developed as sandwich construction of varying thickness. The thickness is based on the location in the skin. The higher up, the thinner the sandwich becomes to allow for flexibility of the seating pods that hang from the skylight openings. This strategy proved successful in both paper and cardboard prototypes as well as in a full scale prototype piece using metal and wood.

With the evaluation of parametric approaches through hierarchical control skeletons the initial investment into the parametric model is high and its topologic flexibility is low. But it offers a high degree of proportional exploration even at full implementation detail. Another shortcoming was the potential loss of the developability property for the surface panels making up the double curved façade. This shortcoming triggered additional research into the generation of developable surface strips from NURBS surfaces. In the reading pavilion project the author established the principle of

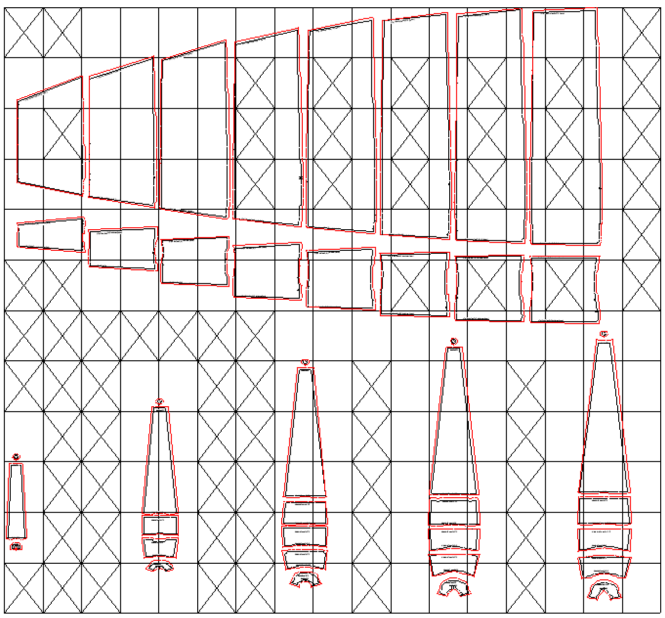


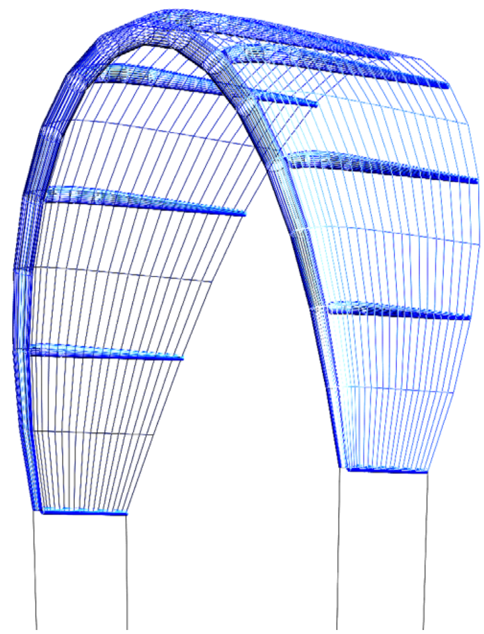
aligning surface strips according to minimal and maximal curvature present in the surface. Where minimal and maximal curvature is very similar in magnitude, the strips are placed diagonally to the two principle directions of curvature. Where one curvature is much bigger than the other, the strip aligns itself with its direction. This results in the spiraling effect of the surface strips in the reading pavilion. This approach was explored computationally in the "scripting strips" project using scripting in Rhino.

Design surface responds to the environment. The design was done through a parametric kite design software by David Aberdeen. It allows the quick fabrication of design iterations based on a set of control parameters for a variety of families of kites.
Test flight of a self built 120 square foot kite by the author and some design variations in the software.. The translation of the pure geometry in a cutting pattern for buildings the main challenge.
Surfplan by David Aberdeen
http://www.surfplan.com.au/sp/

5.1.1.3 Experiment 3: Performative Surfaces –Pneumatically Stiffened Kite Membranes

A relevant variant of the design surface principle with less complexity but more interdependencies is the example of kites used for kite surfing. The experiment shown here exemplifies the connection between designed geometry and actual geometry under load. The tensile membrane, which is stiffened by a pneumatic frame, has no





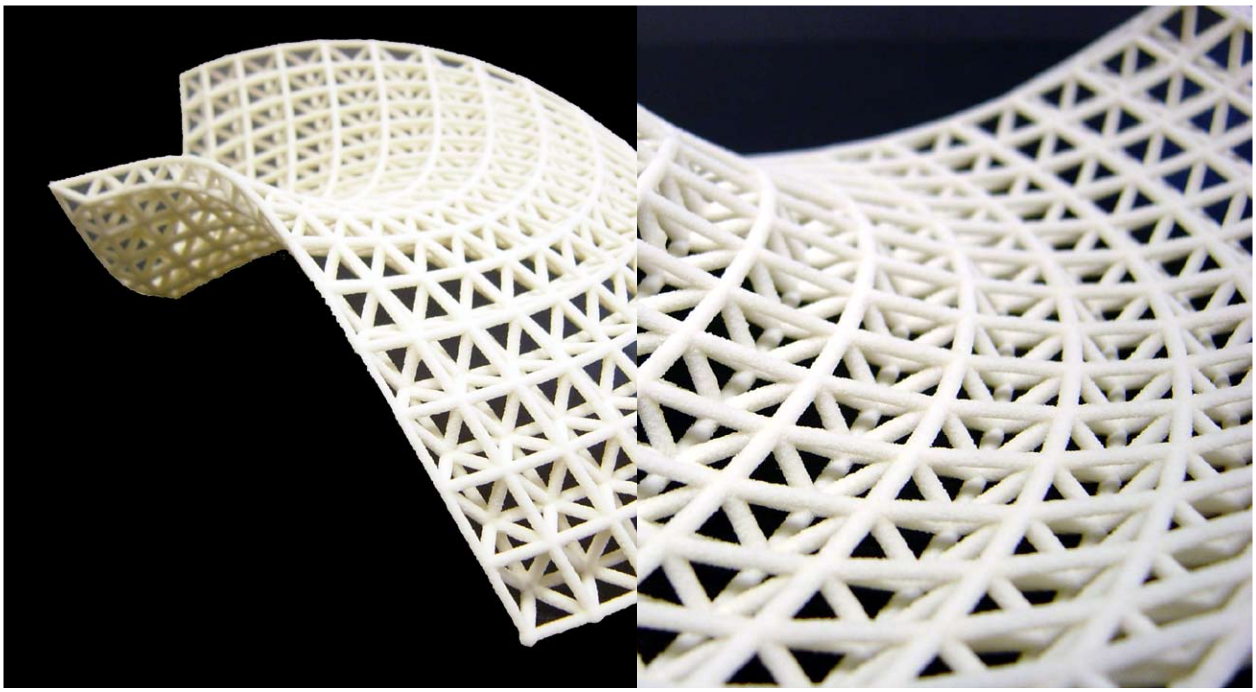
The automated layout patterns of David Aberdeen's surfplan software is an example of the automation of a process of design built iteration previously done manually by the author. David Aberdeen translated the knowledge of the community of kite builders into a design explorer with building output.
Images: Screenshots surfplan software of a design by the author.
Surfplan by David Aberdeen
http://www.surfplan.com.au/sp/

fixed state but is in constant search for equilibrium. Form giving factors are:

- The design geometry, through the wing profile embedded in the fabric cutting pattern.
- The load from the control lines that connect to the rider.
- The wing producing upward lift due the true and apparent wind generate by flying the kite.
- Speed and control input of the rider.

The geometry cannot be modeled without taking into account the deformation under load, where load is a combination of the above factors. In the parametric kite software "Surfplan" by David Aberdeen, which the author used to design and build this sample kite, the geometry emerges from a finite element model that simulates the kite under load. The software is also interesting for its integration of fabrication output as an integral part of the design. The cutting patterns can be printed and used as templates for the fabric pieces to build the kite. Therefore it is a perfect example of an integrated design-simulation-fabrication loop.

Kites are not architecture obviously, but the kite example highlights the dynamics of competing forces present in any design. Design goal have to be aligned and work together. To quantify



Three dimensional Z-corp print with adaptable rod diameters based on the geometric context of the placed component. The geometry is generated through a Rhino script that operates on a NURBS design surface chosen as an input at the beginning of the generation cycle.

such performance characteristics is easier in a kite than in the complex context of building. But for the quantifiable dimensions of architecture at the very least an integration of performative aspects in the generative process would be desirable.

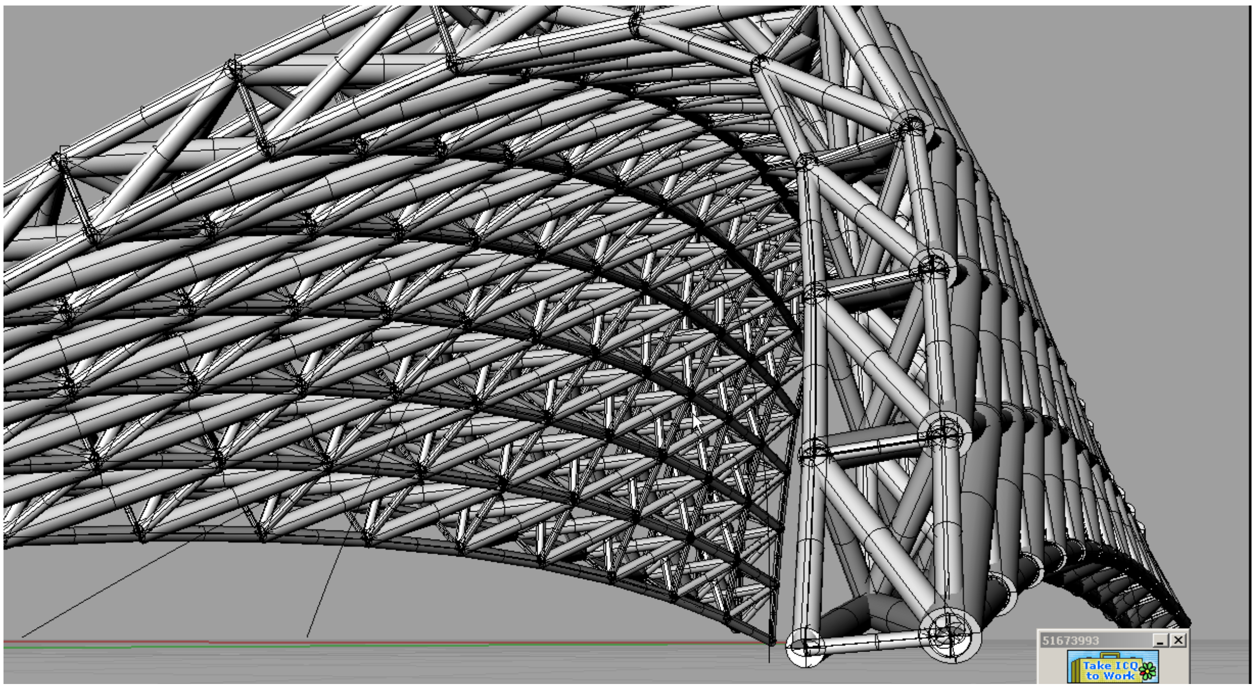
5.1.2 Design Surfaces and Parametric Components

In most real world examples, multiple constraints act in combination on a component of a roof assembly, to take one example. This requires a more sophisticated structure to solve the relationship between surface geometry and environment. The following experiments show different ways of integrating external parameters into a surface based geometry.

5.1.2.1 Experiment 4: Component Population – Geometric

Adaptive Space frames

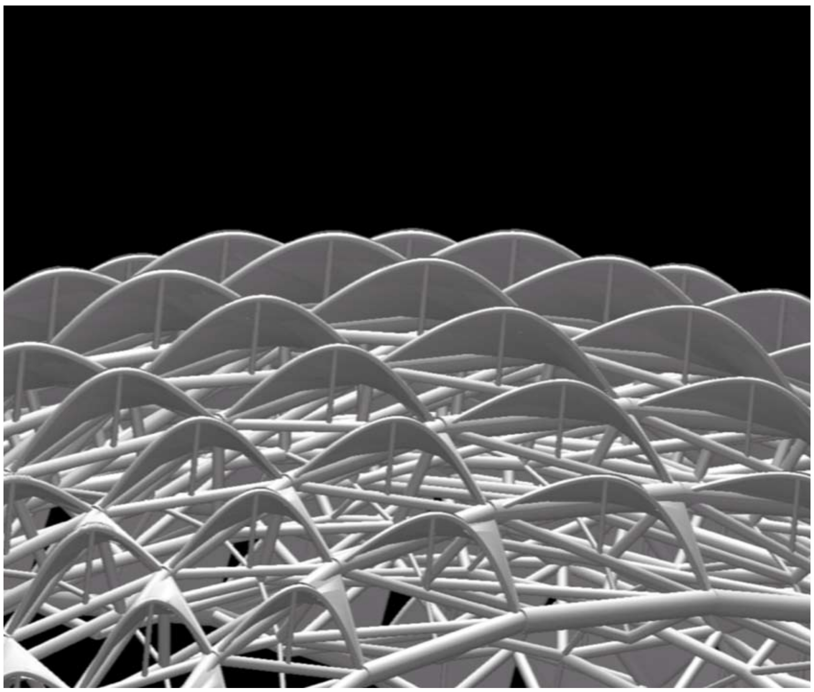
The general case of surface population is based on the parametric representation of surfaces through a two-dimensional UV parameter space. Mapping two-dimensional coordinates onto a three-dimensional surface offers a robust and easy to implement approach to work with design surfaces. This principle has been explored excessively with the use of parametric design systems and



An example of a digitally generated geometry following the underlying NURBS surface and adapting in structural depth and member diameters.

will not be illustrated in depth. But the approach is still powerful enough to illustrate some variants of the basic principles with some additional constraints added.

In the example shown here the basic population unit is a space frame pyramid that can adapt to the distortions of the UV space dictated distortions. All volumetric units are responsive to their respective length and position within the pyramid unit and adapt proportionally. In combination with three-dimensional printing techniques here, a ZCorp© process, the curving space frame can be output in physical format. Its value is very limited beyond a representational artifact as the assembly methods for large-scale space frames are still very different from the additive printing methods available at the model scale. Nevertheless, it holds a certain interest in its implications for material. If material properties can be customized through spatial lattices, the properties of an otherwise monolithic material can be substantially altered to provide additional functionality such as acoustic or tactile qualities or the ability to absorb moisture or light. This could potentially form a research field of spatial material design in the gap between material engineering at the molecular level and material use at

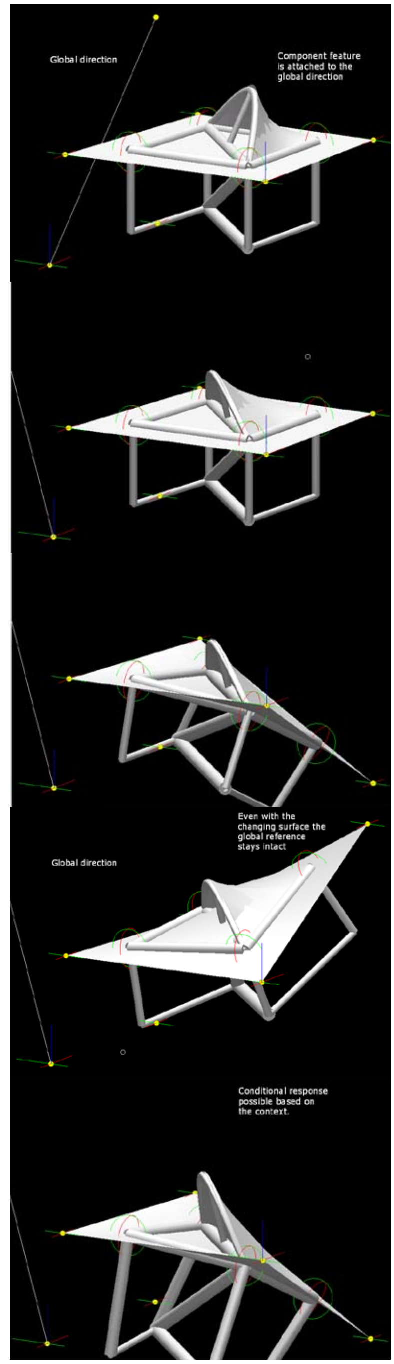


the architectural scale. The component population approach becomes even more challenging with the integration of additional constraints, besides the input from the base design surface.

5.1.2.2 Experiment 5: Integration of environmental constraints – adaptive components

An example of such an interaction between base design surface, external factors in form of the sunlight direction, structural considerations and assembly conditions is the adaptive louver experiment. There are several basic responses that can be integrated:

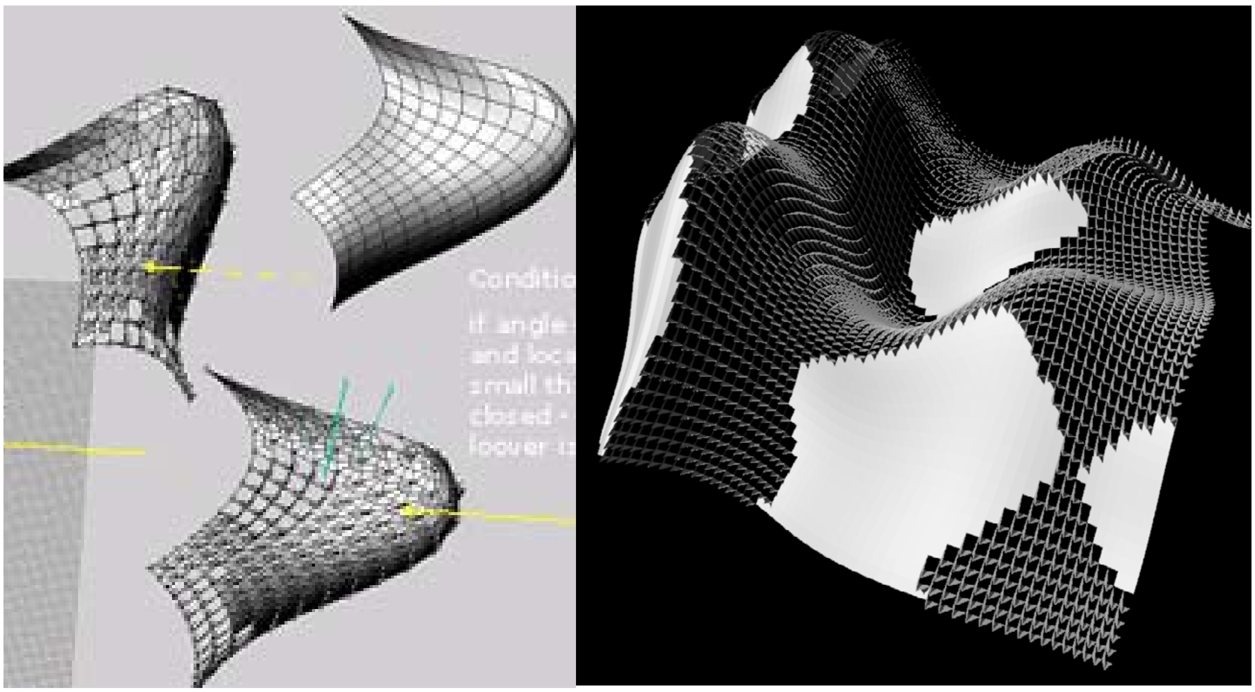
- Locally geometric: response to the design surface
- Globally geometric: adjustment of the unit to a global factor such as the sunlight vector or a predominant wind direction



An adaptive component that responds to three levels of context:
- the local surface geometry context
- the global light direction vector
- component based scaling based on forces present.

- Force propagation: response to propagation of forces from load to the supports;
- Interfacing dimensionally: functioning as intermediary between other sets of components such as a truss between a roof membrane and a structural column. The unit has certain input and output requirements to fulfill.

At the intersection between different constraints and the mediator



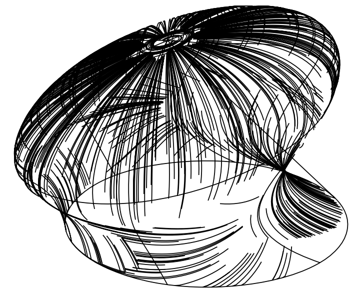
Conditional surface components that adjust their state based on certain thresholds such as a minimum angle between component and sun angle..

between different requirements the unit based approach becomes more interesting. But the possibilities for material response within this system are limited but interesting field responses emerge out of the interplay of the different constraints.

5.1.2.3 Experiment 6: Conditional Components

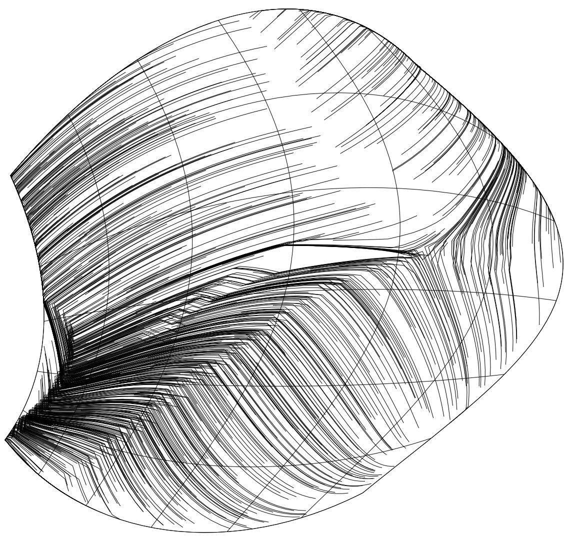
The introduction of conditional behavior in the geometric varying components allows for a new set of behaviors. For instance it is possible to create components that orient themselves to the sun but are closed if the angle of the sun passes a defined threshold in relation to the surface. Even with simple conditionals like this the vocabulary of possible responses and the emerging patterns are convincing in their response to the local surface geometry.

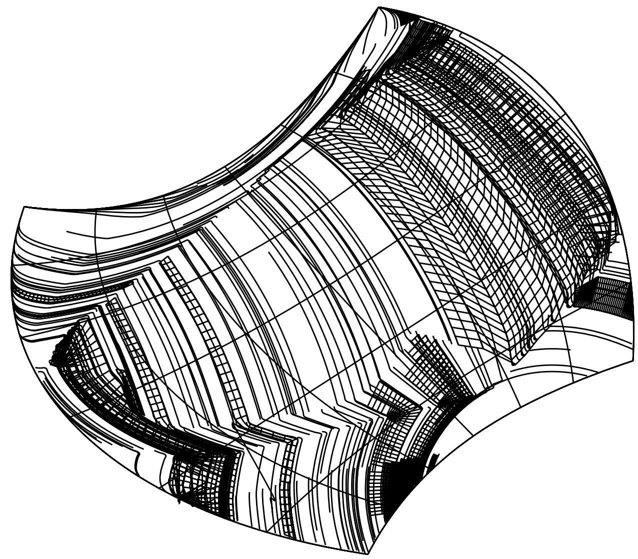
The next level of conditional behavior would be the detection of neighboring components and the propagation of component states throughout the system. At that point it would be advisable to depart from the grid like positioning of components towards more flexible placements that are not dependent on the UV grid alone.



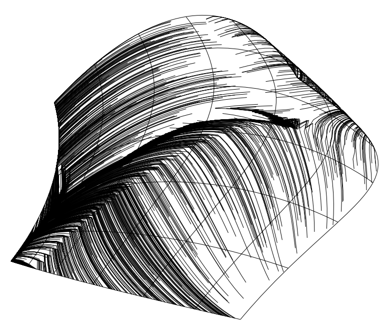
5.1.3 Surface guided Component Generation

From the set of previous case studies it becomes apparent that



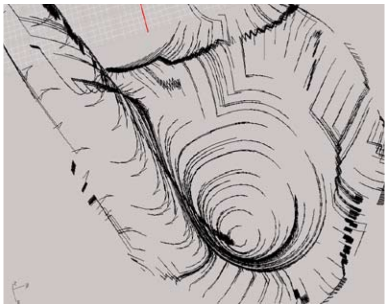


the top down subdivision of a design surface into homogenous components is not adequate for many design problems and other strategies need to be developed. The following two experiments follow the approach of growing components within the desired constraint envelope of the design surface. In the first experiment, the scripted strips, this is done by using top level design inputs in form of a design surface. In the second experiment, it is accomplished by using a grammar in form of a Lindenmeyer system. The second experiment is a citation of the MoSS project by the Emergent Design Group headed by Peter Testa, together with Una-May O'Reilly, Markus Kangas and Axel Kilian (Testa et al 2000).



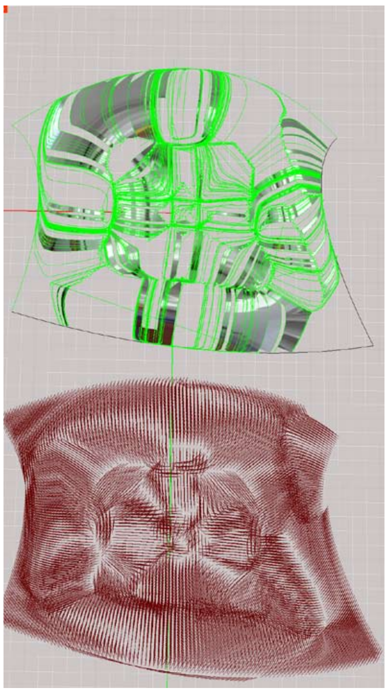
Traces following the min max curvature vectors on double curved surfaces. The traces are tracked and joined into strips if they are sufficiently close to form a developable surface strip.

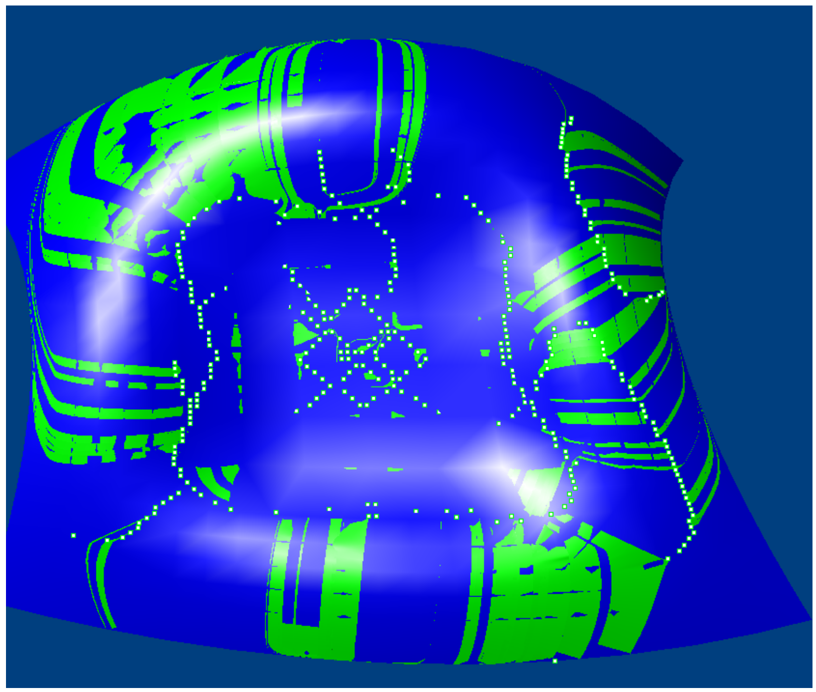
5.1.3.1 Experiment 7: Growing developable strips along a

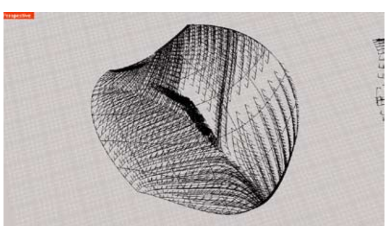


surface – Scripted Strips

With the Scripted Strips the author further developed the maximal-minimal curvature surface approach for developable strips started on the reading pavilion. In this case, the placement and orientation of the strips is generative through a script that extracts the maximal and minimal curvature of a NURBS surface and provides those samples as the input to a surface walker that creates paths



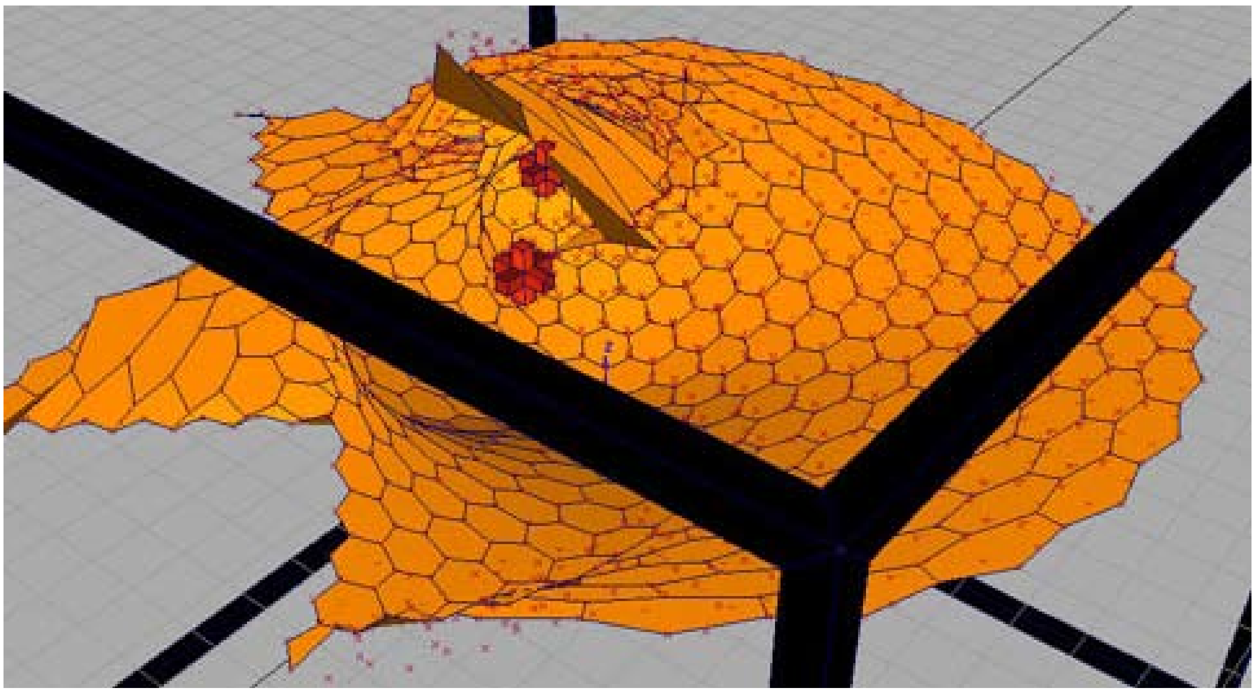




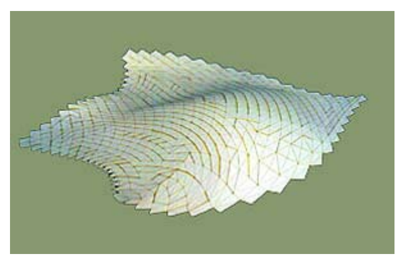
on the surface. A second step evaluates the created paths for their compatibility with the developability goal and creates strips from the set of matching curves whenever possible. The test for developability is not very rigorous but close enough to function with flexible materials such as cardboard. The more interesting aspect of the case study is the emerging tiling patterns that reflect the heightened properties of the underlying NURBS surface. The stepping progress stops when 90 degree turns are encountered

More complex surfaces require an adjusted technique for determining the starting points for the strips on the surface. The change in curvature is monitored and at thresholds of change a new strip is started.

and strips are only created if the gap between adjacent borders is below a given threshold. The emerging topology is very interesting, resembling the patchwork agricultural landscape of parts of Europe. This might suggest that similar factors influenced the laying out of farm land in hilly terrain due to access routes and slopes of the fields and their influence on what equipment or tools could be used.



The key result of this study is that complexity and variation can emerge out of the interplay of simple spatial rules and a given design surface.



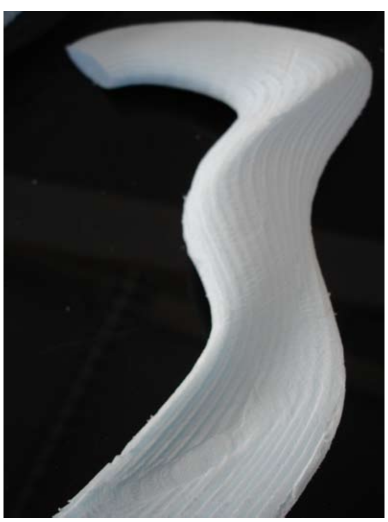
MoSS by the emergent design group. MoSS is a surface growth program based on Lindenmeyer systems in three dimensional space using attractors and repellors to shape the growth of surfaces during generation.. Image: Axel Kilian for edg. http://web.mit.edu/edgsrc/www/moss/index.html

5.1.3.2 Experiment 8: Growing the components into a surface – MoSS project –Peter Testa et al.

The MoSS project employs a Lindenmeyer based growth grammar. Each geometric unit develops out of the interplay of the grammar instruction and the geometric distortion when implemented in the three dimensional environment. The project is documented in depth by the edg group who developed it. The edg group was headed by Peter Testa at MIT in 1999. The author contributed the processing of the growth geometry into geometry that can be

Translation of NURBS surfaces into milled volumes following the isoparam curves. A very light weight approach was taken to model the surface with four equal order B-Splines only. This allows for robust editing of the splines and easy fabrication.

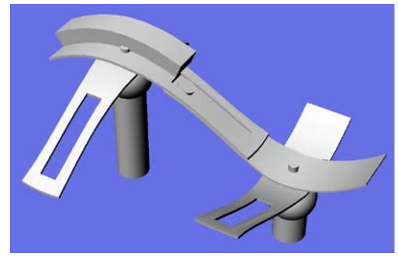
prototyped through laser cutting. MoSS is a project developed by Peter Testa, together with Una-May O'Reilly, Markus Kangas and Axel Kilian (Testa et al 2000) prior to this thesis. The contributions by the author carried on in future experiments documented here. The key result of the MoSS projects are: Growth as a generative principle with the presence of forces is a powerful concept and the MoSS project was only a preliminary study of what is possible. If



growth principles are applied in multiple stage fashion additional performative constraints could be applied which force based deformation or load dependent thickening of materials. Overall it would provide a higher integration of form contributing factors.

5.1.4 Fabricating Surfaces – Material Computes

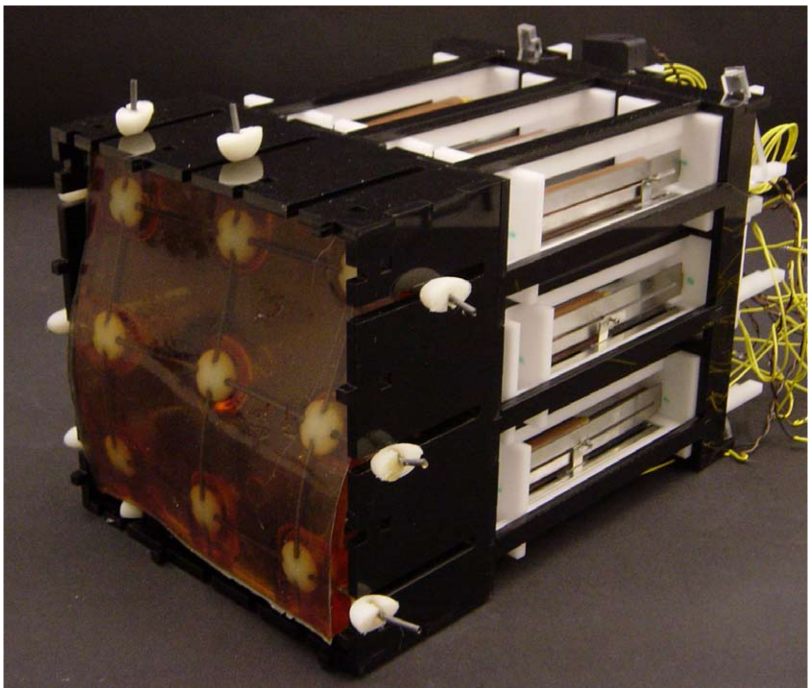
Several of the previous studies came upon the effect of material resistance. In fact the description of computational geometry is in a large part based on formalisms derived from material based curvature generation. The following case studies are based on the material ability to interpolate curvature based on material resistance. The projects are fabrication surfaces that allow for the control of curved surfaces for the casting of tiles. The control is reduced to a grid of pins that are attached to a rubber flexible surface.

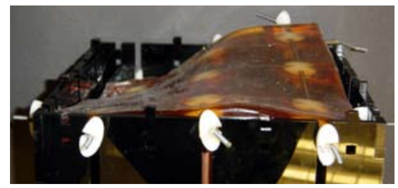


Project with Kyle Steinfeld for the "How to make almost anything" course by Neil Gershenfeld.

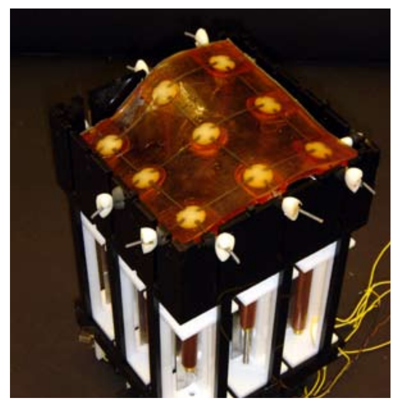
5.1.4.1 Experiment 9: Surface Fabrication through three axis milling

This case study is about the creation of surfaces through a subtractive milling process. The milling paths are directly linked to

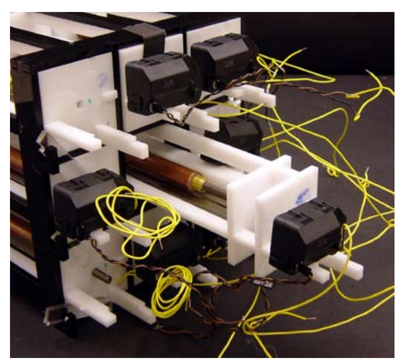




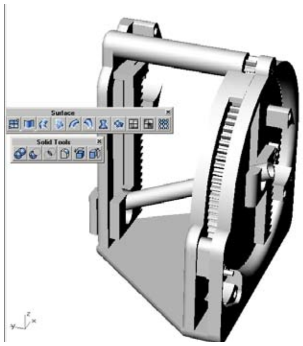
Computer numerically controlled pistons drive the surface into the desired position.



the isoparam curves of the NURBS surface patches. The modeling approach for the surface is that of creating a closed loft from 4 edge splines in space. This creates the aesthetically appealing spatially undulating surface milling pattern and is a very robust and fast modeling and milling approach. For volumetric double sided cuts the part needs to flipped in its frame. The next level of complexity is reached when the cutting depth or the geometry of the surface requires a part assembly of individually cut pieces. The generation of the parts from the constraints of the fabrication envelope constraint and the assembly logic poses an additional challenge especially if surface continuity across the parts is crucial. The thesis contains examples of such assemblies in the one wheeler car design study where carbon fiber molds were fabricated in such a



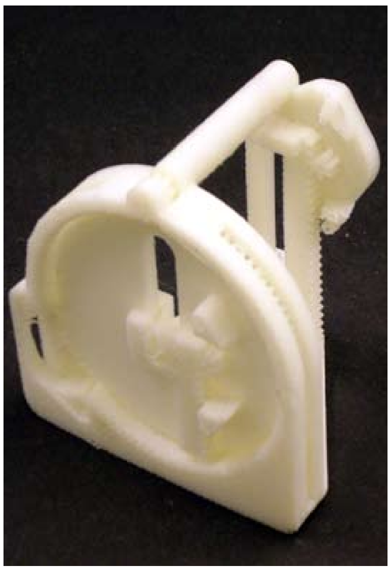
Computing surfaces outside the digital context. Material computes its shape based on material resistance. A mechanical control polygon can recreate digital curvature physically. Image: Axel Kilian, Kyle Steinfeld.



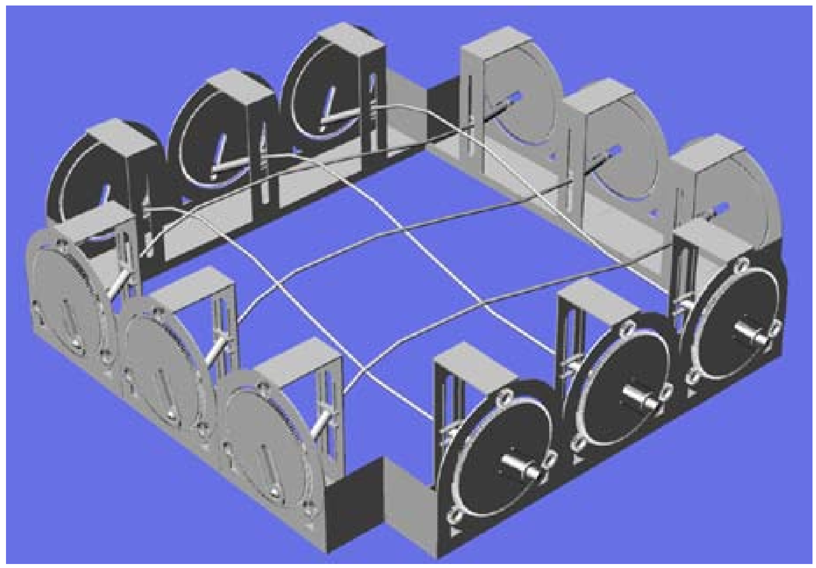
fashion. This process should ideally be automated with the cutting geometry and the fabrication constraints as the inputs.

5.1.4.2 Experiment 10: Fabrication devices using material resistance to approximate curvature

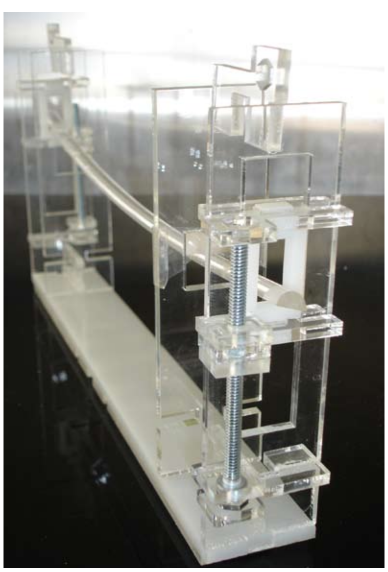
The thickness of the rubber surface interpolates the control pin grid to a double curved surface that allows for the casting of

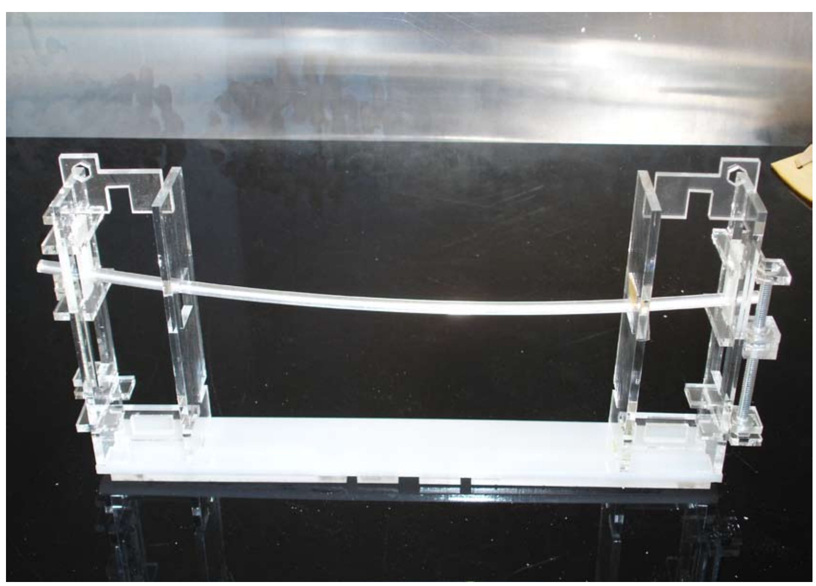


Physical spline model controlled through a three-degrees of freedom polar tangency device.

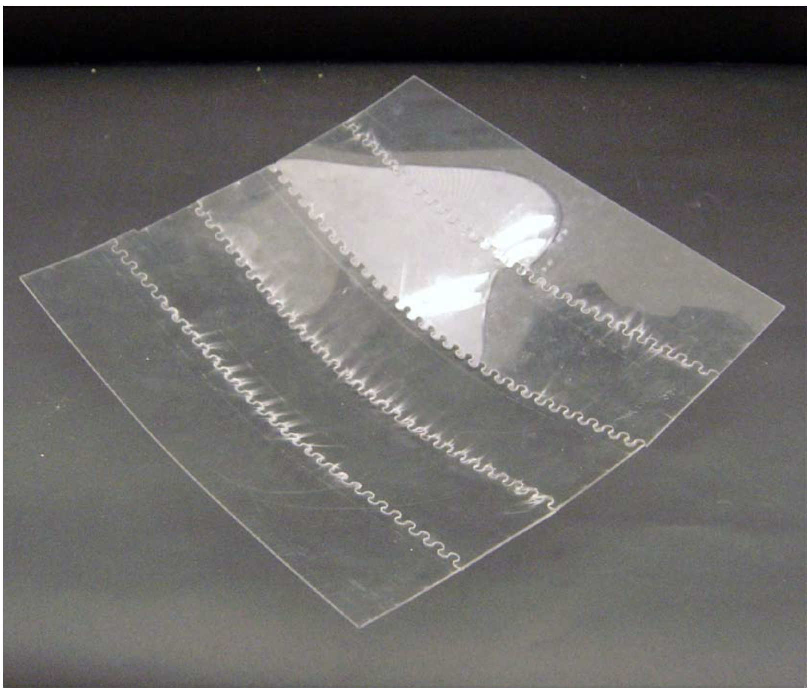


The tangency controllers are positioned at each end of the physical spline grid and control the curvature from the rod's ends. Both physical curvature devices were developed for adjustable ceramic molds. Material resistance plays a key role in the determination of form





ceramic tiles. The control of the pins is through motorized threaded rods with sliding potentiometers attached to them for position sensing. The main challenge was how to keep the surface taught in different curvature positions, as the material has to substantially lengthen in a curved position. The implications for surface design and fabrication are that a possible approach to the fabrication of complex geometry maybe to leave the material curvature control



up to the physical entity itself and calculate it through the material resistance itself. This project was developed as part of the class how to make almost anything with taught by Neil Gershenfeld. The final project was developed in collaboration with Kyle Steinfeld.

An earlier development by the author envisioned a physical spline based surface controller for casting tiles as well. The important part is the physical tangency controller for introducing curvature into the spline grid. It was designed to control the slope of a tube through its start and end point. The tube would be connected to a spline that would be dimensioned based on the physical resistance of the material. The project did not advance beyond the prototyping of the tangency controllers.

5.1.4.3 Experiment 11: Puzzle surface – material constraints

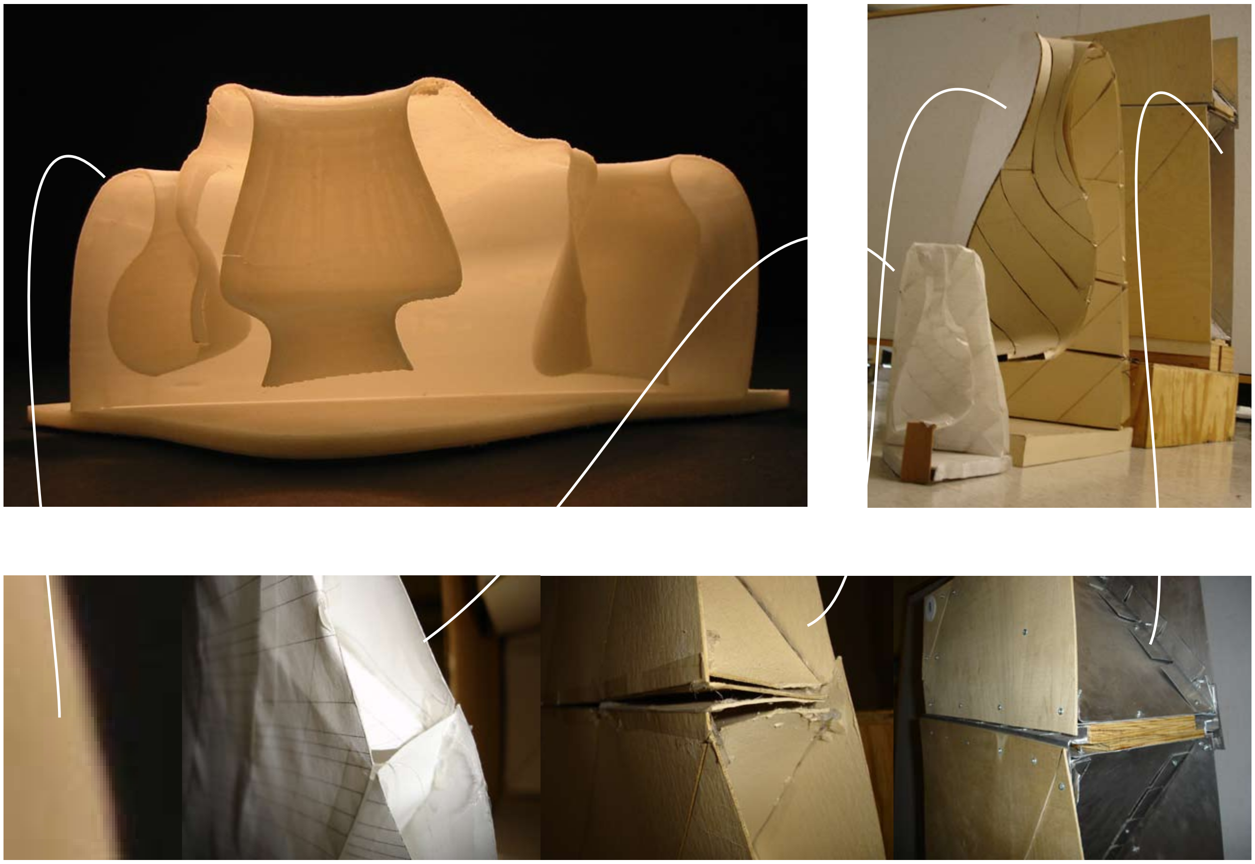
Material resistance also plays an important part in forced assemblies of parts. In this variation of the puzzle surface polycarbonate was waterjet cut following the design geometry. After assembling the piece the friction fit joint introduces local double curvature by distorting the material. This tensioning of the material distorts the overall geometry away from the design geometry, but at the same time it creates a stronger shell like assembly of the parts. These effects can be used to the advantage of the design if they can be sufficiently well predicted. Also the design geometry has to take into account the distortions introduced by the assembly process in order to achieve a predictable overall shape. Because of these challenges such effects are avoided where possible in building practice. Design exploration should integrate this evaluation to use the material potential. In bridge design and large engineering structures this is already the case. Equally traditional craft based processes are using similar principles for efficient use of material. For instance basket weaving relies on the interlocking of rods that are stressed in bending and provide the elastic rigidity of the final assembly. More such synergy effects would be desirable in construction.

– geometry feedback

The three-dimensional jigsaw puzzle surface works similar in providing resistance to the bending forces introduced through the connections between the surface strips. The additional challenge in the jointed connection is the forced overlap of different curvatures in the adjacent strips. The puzzle joint from one side shares curvature of on strip but reaches over into the other strip,

Reading Pavilion with Tim Morsehead and Carlos Barrios.. The mockups for the pavilion ranged in scale from 1:30 to 1:1. All used the exact same design geometry to drive the scale based translation into constructible parts. The addition of material thickness and component volume and the fragmentation of monolithic parts into sub elements with the required connections was a good exercise in modeling robust design geometry that can support a range of scales and material dimensions through parametric changes.
Images: Axel Kilian, Tim Morsehead

which follows a different curvature. The friction fit connection forces the part partially into the curvature of its neighbor. Due to the flexibility of the material partial double curvature occurs around the connection and the joint gets additional forces from the incompatibility of the curvatures.



3-D print section1:20 Paper model section1:10 Cardboard model section1:5 Wood/Metal model section1:1

5.1.4.4 Experiment 12: Scale steps of prototyping surfaces

The reading pavilion study also was an interesting study in the development of scale related prototyping choices from a shared digital model. With each scale increase the core geometry generated by the CATIA© model was interpreted with additional information and thickened according to the material. The joining places posed an additional challenge due to the unconstraint nature of the connection. Intentionally the connection between the pre fabricated brick units was left curved, dubbed also the roller coaster connection to its spatial twisting of the two surface rails. The detail modeling was done manually using the wire frame geometry as the center line of the joint development.
The reading pavilion was a project conducted in collaboration with

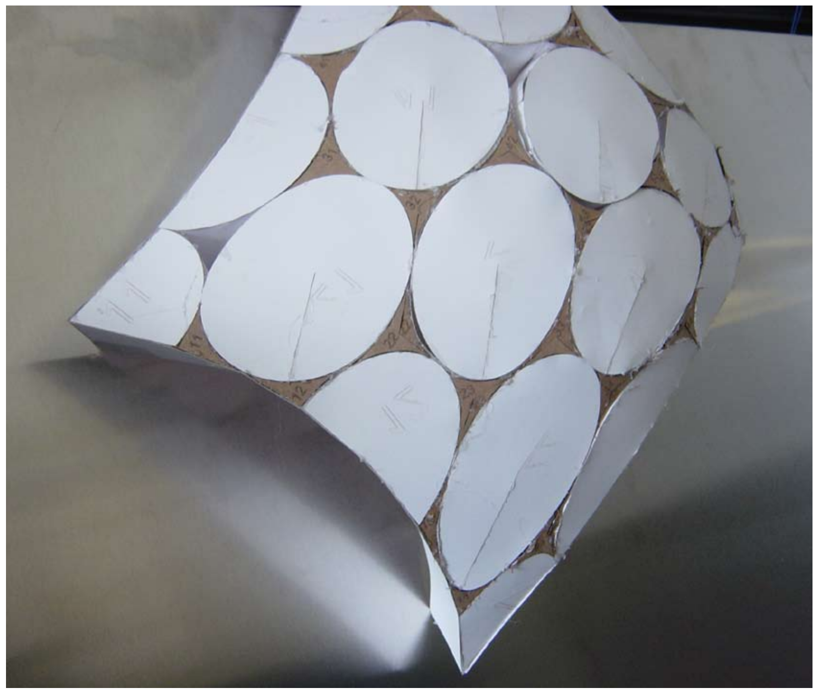
The translation of design geometry through four different scale to a fully detailed assembly mockup in metal and wood. There are several adjustments necessary.
First: offset of material thicknesses and the adjustment of neighboring part relationships.
Second: formulation of connection details as scale, mass and forces increase.
Assembly sequence dependent details as components become heavier and more rigid.
Images: Axel Kilian, Tim Morsehead.

A cone based construction approach to double curved surfaces. The developability constraint is embedded in the geometric primitive of the cone. The edges lie on the target surface as well as the high point of the cone. This provides a very good visual fit to the surface. Cone based rationalized construction dates back all the way to R. Liming, who designed fighter aircrafts during WW II. (Liming 1944)

Tim Morsehead and Carlos Barrios.

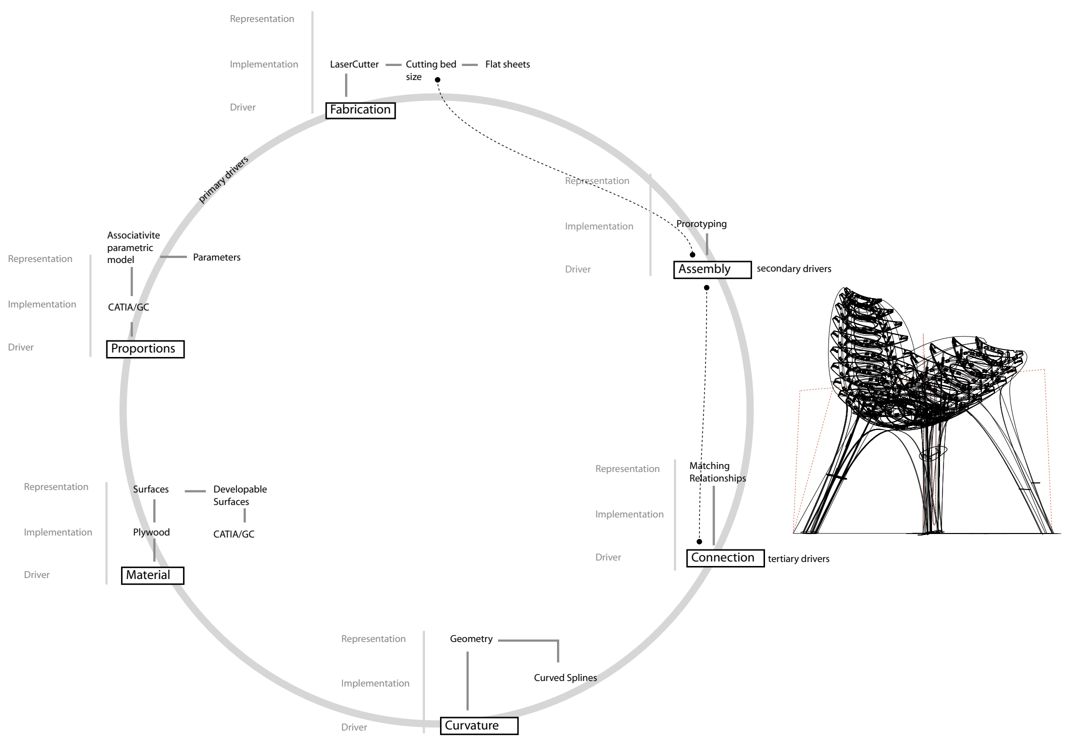
5.1.4.5 Experiment 13: Cone based fabrication translation of double curved surfaces

Geometric primitives implement the developability constraint for the translation of a double curved surface into developable surface parts.



One other possible approach in adding material constraints into the fabrication of freeform geometries is through the use of geometric primitives that enforce the material condition required to build the surface physically. One such small case study is the cone based surface example, where the use of the cone primitive allows the translation of a double curved surface into developable strips with a comparatively tight fit with the original design geometry. The

Overview of the physical based experiments. In summary he image shows a range of formulations of material strategies in response to material constraints, design surface and generative and explorative methods.



The chair experiment diagram.

reason for this visually seemingly very accurate translation is the fact that the cone’s bases follow the underlying design surface directly as do the tips of the cones. Thereby the recognizable features of the geometry support the properties of the surface whereas the reduction in geometric precision happens in the single curved parts spanning between base ridge and cone tip. One could argue that this is very similar to the usual developable strip approach used in many projects to date but in fact there the paths are less likely to be spatial or if they are they are still dividing the surface up in clear linear sections emphasizing the facetting of the surface and the loss of precision in that direction. There are many subsets to the cone approach and most of them really only become useful with a programming approach to processing the surface and the geometric primitives, some of which have been previously explored involving Voronoi diagrams and Delaunay diagrams.

5.1.5 Chair Experiment – Circular Exploration between Surface and Material Constraints

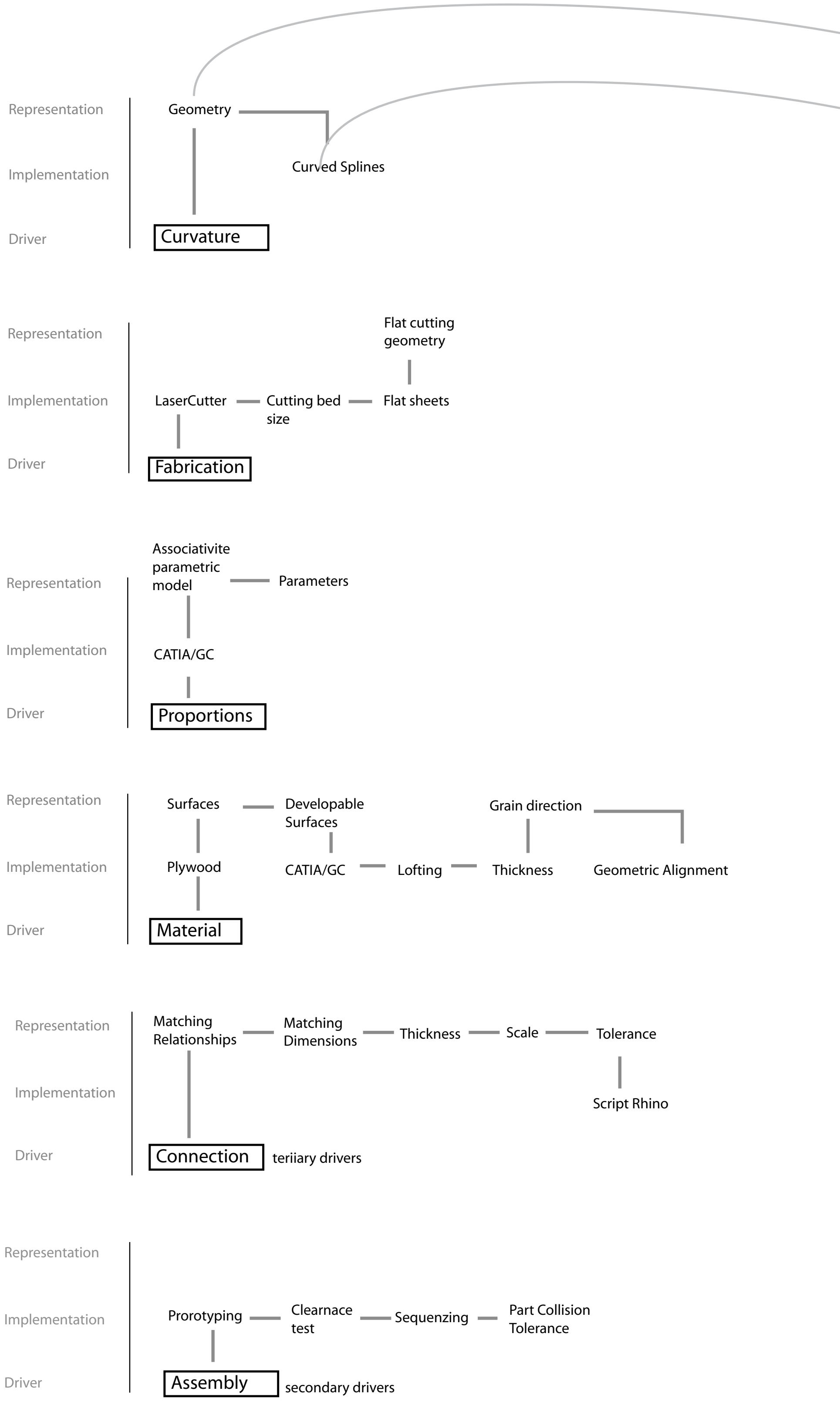
Example of a design case study that integrates several constraints in pursuit of an aesthetic design goal formulated through a geometric sketch.

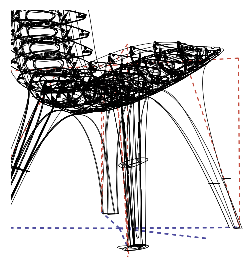
The chair case study was conducted as a test application of the original puzzle detail on a designed object scale such as a roof or a piece of furniture. In this case due to time and cost constraints a chair was chosen as the test case as both material amounts and fabrication tools stay at a manageable stay for an initial first run.

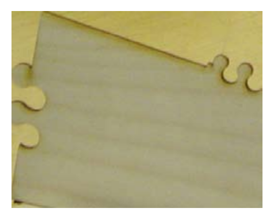
5.1.5.1 Process

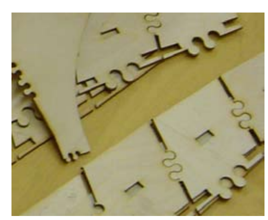
The chair case study was conducted over a period of three months starting with a design idea in form of a geometric sketch model. Over the course of development the aim was to produce a prototype using thin plywood and explore the aesthetic and structural potential of single curvature based assemblies.

The chronological sequence was a paper based 1:2 scale mockup of the geometry sketch in rhino. The literal translation proofed to be structurally insufficient to support weight. The findings of the paper mockup were valuable in identifying the weak spots in the assembly for a second round of geometric modeling. For the second digital representation a parametric model was chosen and implemented in a number of parametric environments, namely CATIA© and Generative Components©. The parametric model severed as the persistent accumulative model at the center of the iterative modeling cycles. The topologic structure of the control geometry allows for adjustment of the number of parts. A set of parameters allows for the adjustment from the iterative prototyping cycles. The main role of the parametric model is therefore to collect and unify the findings from the range of prototypes and aesthetic evaluation conducted in the process. The dependencies are circular; meaning any change in one parameter affects all the other parameter. These cross dependencies are a challenge and


Representation
Implementation
Driver
Geometry
Curved Splines
Curvature
Representation
Implementation
Driver
Flat cutting geometry
LaserCutter
Cutting bed size
Flat sheets
Fabrication
Representation
Implementation
Driver
Associativite parametric model
Parameters
CATIA/GC
Proportions
Representation
Implementation
Driver
Surfaces
Developable Surfaces
Grain direction
Plywood
CATIA/GC
Lofting
Thickness
Geometric Alignment
Material
Representation
Implementation
Driver
Matching Relationships
Matching Dimensions
Thickness
Scale
Tolerance
Script Rhino
Connection
teriiary drivers
Representation
Implementation
Driver
Prorotyping
Clearnace test
Sequenzing
Part Collision Tolerance
Assembly
secondary drivers







usually the goal is to eliminate such effects in a design process. In the chair example these cross dependencies were intentionally allowed and studied for their potential in driving the aesthetics of the chair design further. The core parameters were:

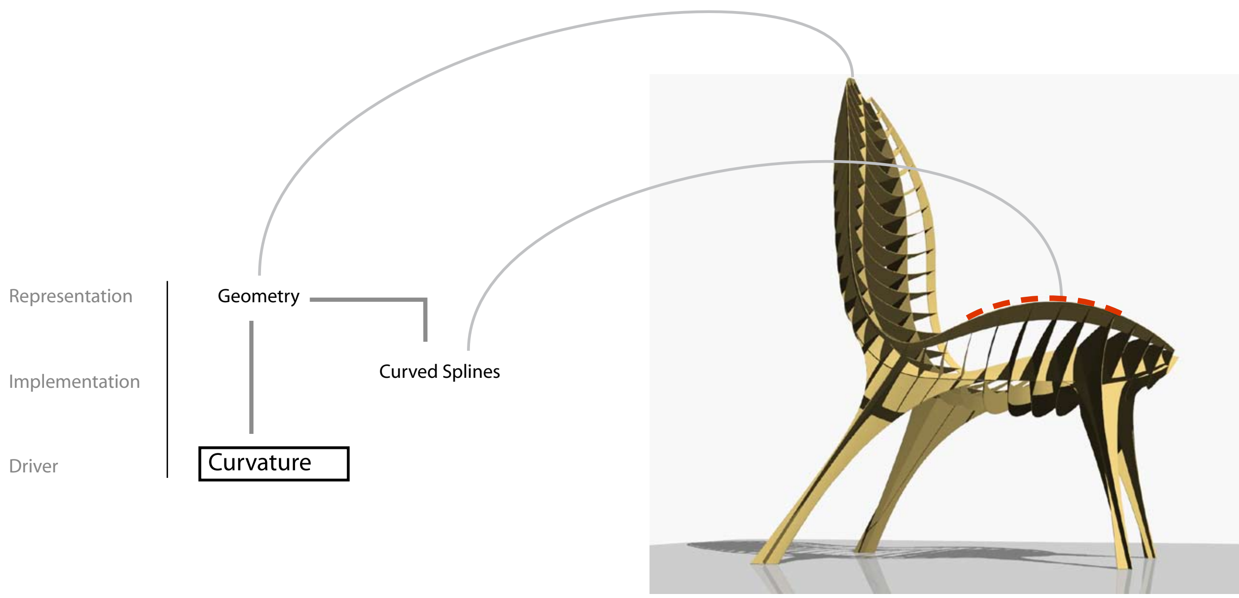
- thickness of material

- maximal allowable curvature
- tolerances for friction fit joints
- tolerances for assembly critical joints
- structural scaling of components
- aesthetic parameters for proportion
- aesthetic parameters for distribution of parts
- aesthetic parameters for number of parts

Some of the parameters, for instance the assembly tolerance parameter, were added during the process due to experiences with the full-scale partial prototypes. Others were there from the beginning as for instance the proportional values for the overall design of the chair structure.

Some of the parametric descriptions have topologic flexibility meaning they can adjust fluidly to the adjustment of the number

The initial geometric sketch modeled in Rhino from NURBS surfaces and splines. The geometry captures an idea and turns it into an image. It formulates curvature in the relationship of the parts as the aesthetic driver for the chair design.. The design motivation was from the beginning to build a chair assembly with all curved developable surface parts through a friction fit assembly.

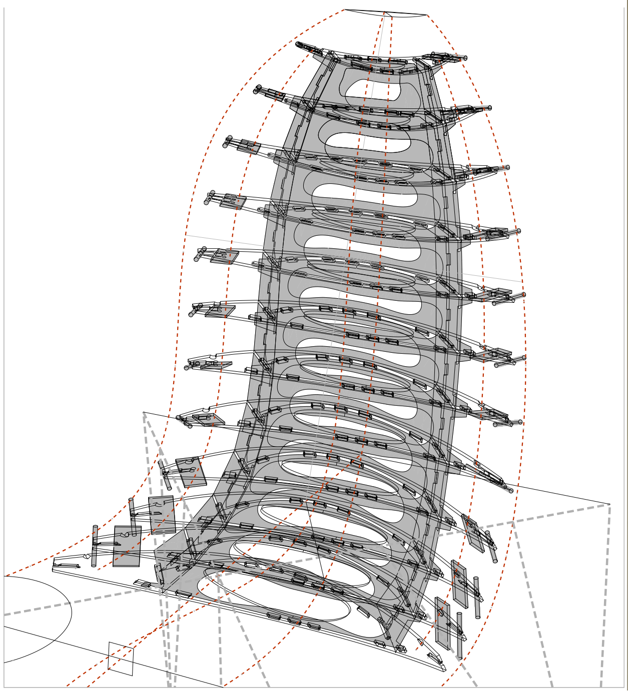


Study of the curvature constraints and its representation through geometry and implementation through spline curves and surfaces.

of parts through proximity triggered point selectors and topology independent representation of parts.

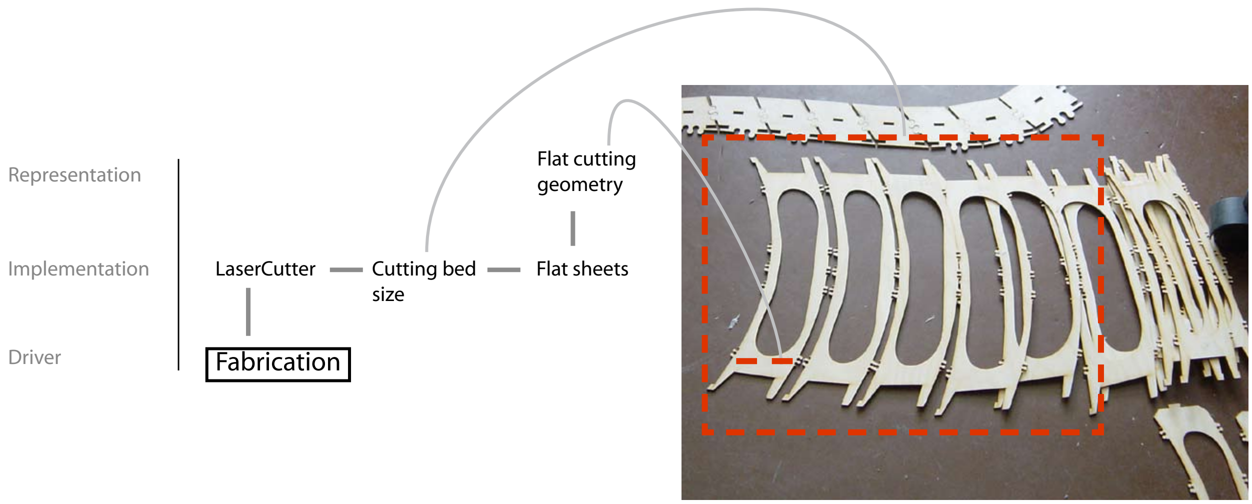
5.1.5.2 Geometric sketch

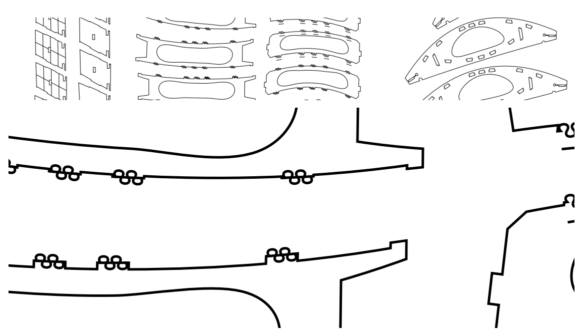
The geometric sketch captured and established the aesthetic intention of the design as a stylized constraint free, spatial composition carrying the idea over into geometric form. This design



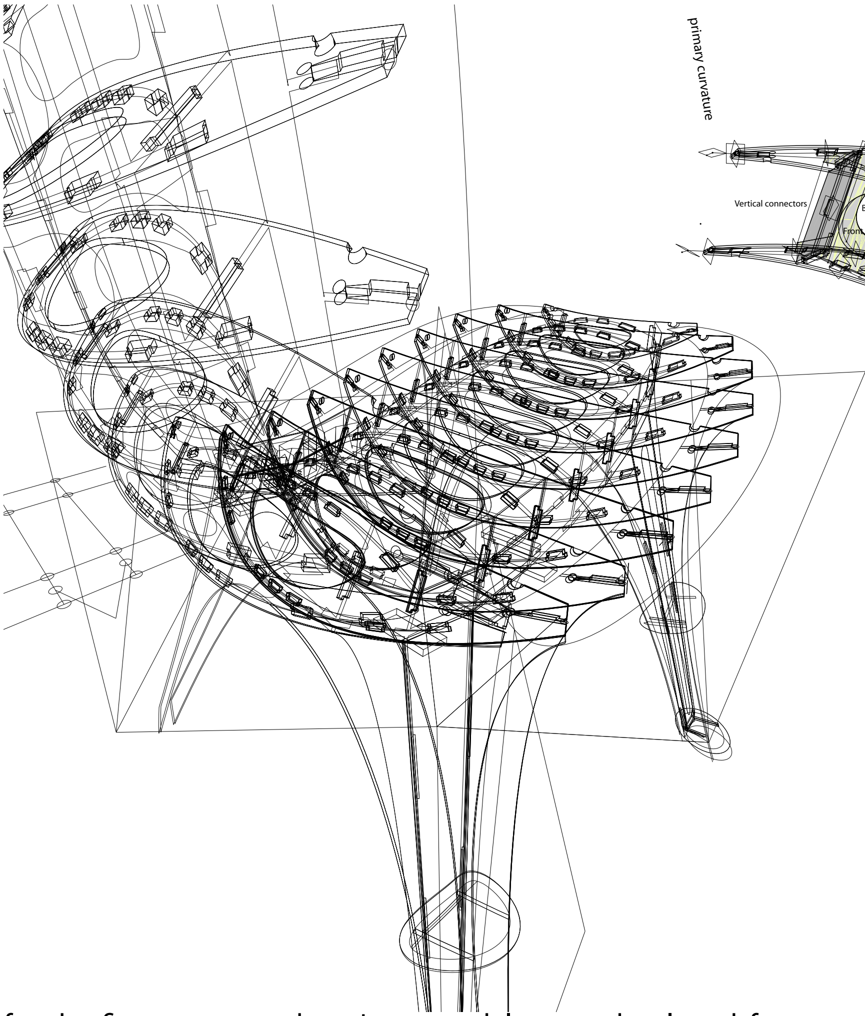
was produced in a matter of hours without additional sketches directly within the three dimensional modeling environment. It is not measured to be to scale or even functional but is purely a geometric response to a design idea that formed the basis and evaluation base for all further exploration as more and more constraints were integrated into the design up to the full scale prototype. The guiding principle was the interest of the author to design a chair from thin curved surface pieces with a light-weight, feature-rich appearance. The geometric sketch establishes a testing platform to measure later design variations against. It was not directly used for the following design representations except

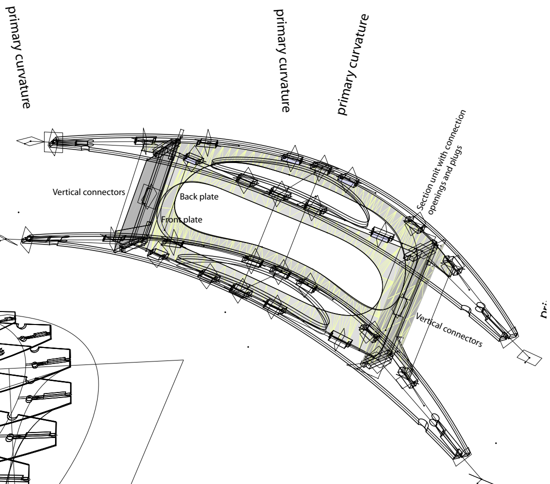
Establishing the fabrication constraint as a design driver for the translation of the geometric idea into a prototypable and fabricatable assembly. Wood and the laser cutter are chosen as implementation choices for the design driver of fabrication. The image shows the final chair assembly both in parametric description and in physical form.





The fabrication design driver is implemented through the laser cutter, which triggers several other constraints such as cutting bed size, flat sheet material stock and flattened cutting geometry.



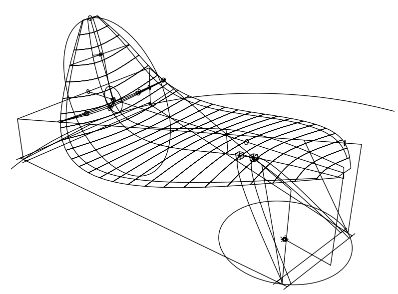


for the first paper mockup. Later models were developed from scratch, guided by the findings of the earlier prototypes. This was done to avoid post rationalizing the initial geometry and instead let the design idea and its geometric manifestations evolve and mature without the direct transfer of the initial geometry.

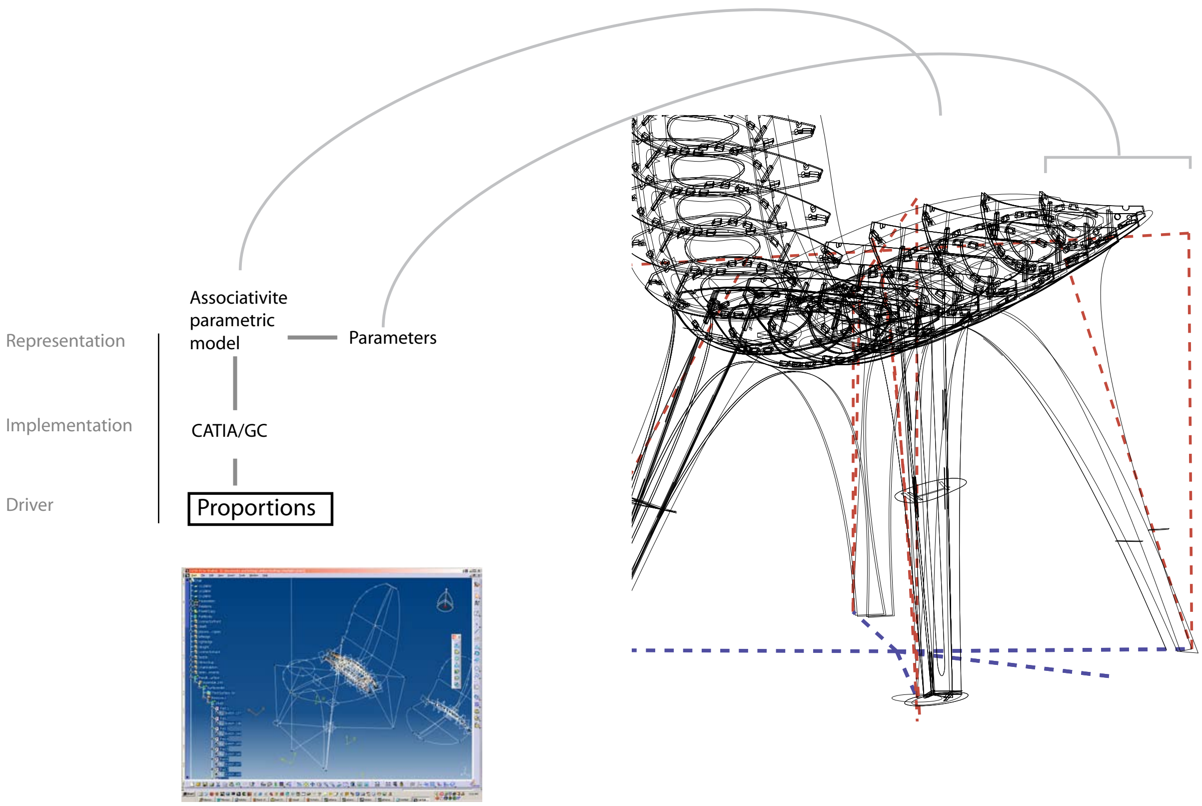
5.1.5.3 Paper Mockup

The mockup is a direct mapping of the geometry sketch into a paper model at a scale of 1:2 in order to verify the spatial validity of the design. The paper model helped to validate the developable surface condition for all the parts of the geometric model sketch. But it also showed substantial structural flaws with the open meshed curved surface assembly. Through several iterations adjustments

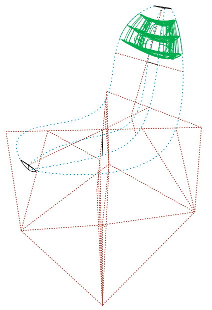
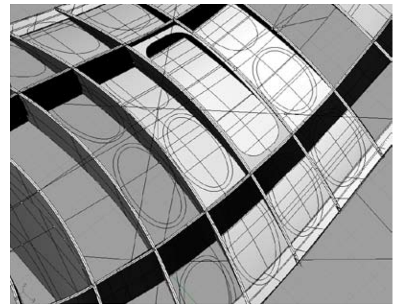
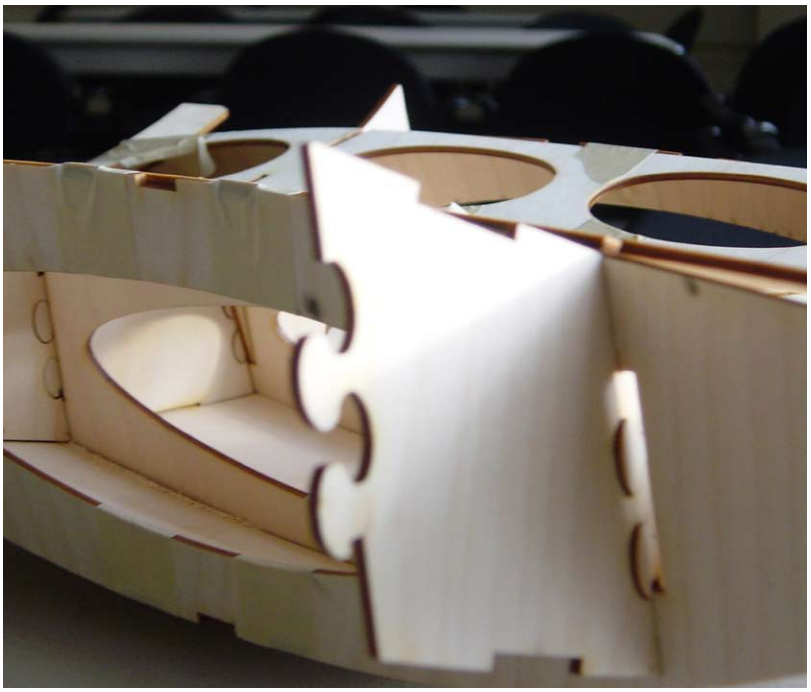
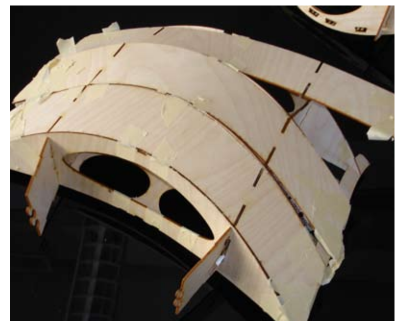
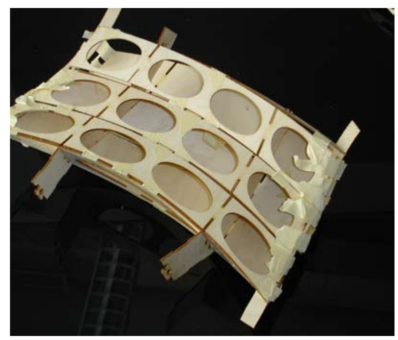
The proportions introduced in the geometric sketch act as the design driver. Through parametric software in CATIA and GC they are translated into an associative model for the control skeleton.
Proportions are then implemented through the establishment of equation relationships between the



different geometric entities. The proportional links between parts also ensure the correct geometric response for varying scales of the model.



The proportional constraint is implemented through CATIA and GC and represented through an associative, parametric geometry. This construct is controlled through parameters, which define the proportional relationships between the parts.

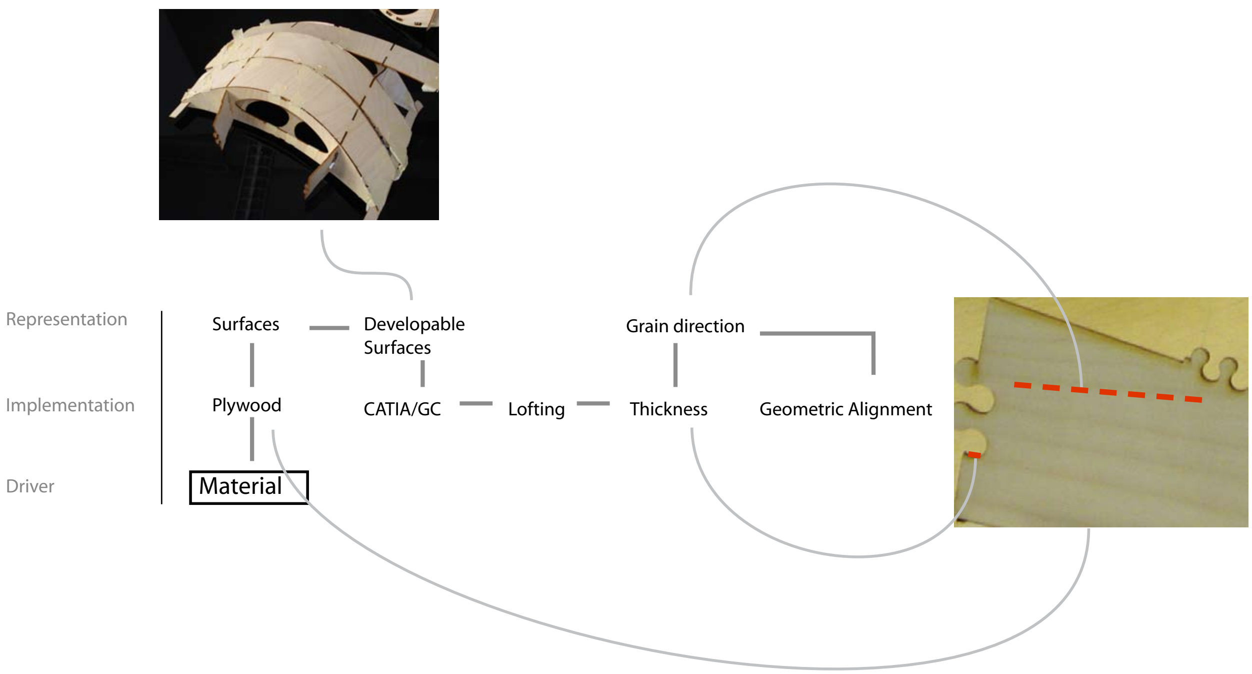






were made in the physical prototype to test possible changes. These findings were passed on into the next design presentation, the parametric model of the next iteration.

5.1.5.4 Parametric platform for gathering revisions and exploration

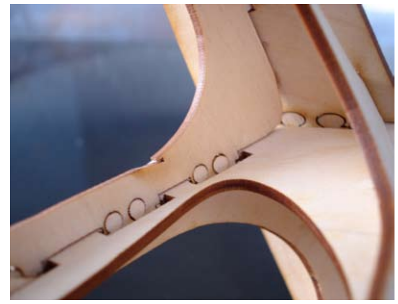
The parametric model numerically links the parameters together in a simultaneous representation of the current model. In its gathering role it is central but in terms of its role in the design process it is only one of several design drivers. There are other factors in the interaction between the different design representations that are not captured by the parametric model like the aesthetic of the design or the assembly sequences. Therefore the design is not

Establishing the constraints for wood happens through a series of material mockups of the design geometry. The findings are analyzed and mapped into the design geometry-material geometry translation. One such finding is the role of grain direction in the types of part geometries of the chair. Different orientations were tested and the best fit integrated into the process illustrated to the right. Selective prototyping played a key role in this small material based exploration.



The material constraint is turned into a driver through the choice of 1/8 inch plywood. The material implementation is represented through surface geometries, which in turn are required to be developable to fit the material choice and the fabrication constraint. The implementation of the developable surfaces again happens through CATIA as the implementation choice, with parameters for material thickness and geometric alignment constraints to account for grain direction.

completely externalized but still many of the circular dependencies become apparent even in this partial representation of the design dependencies.

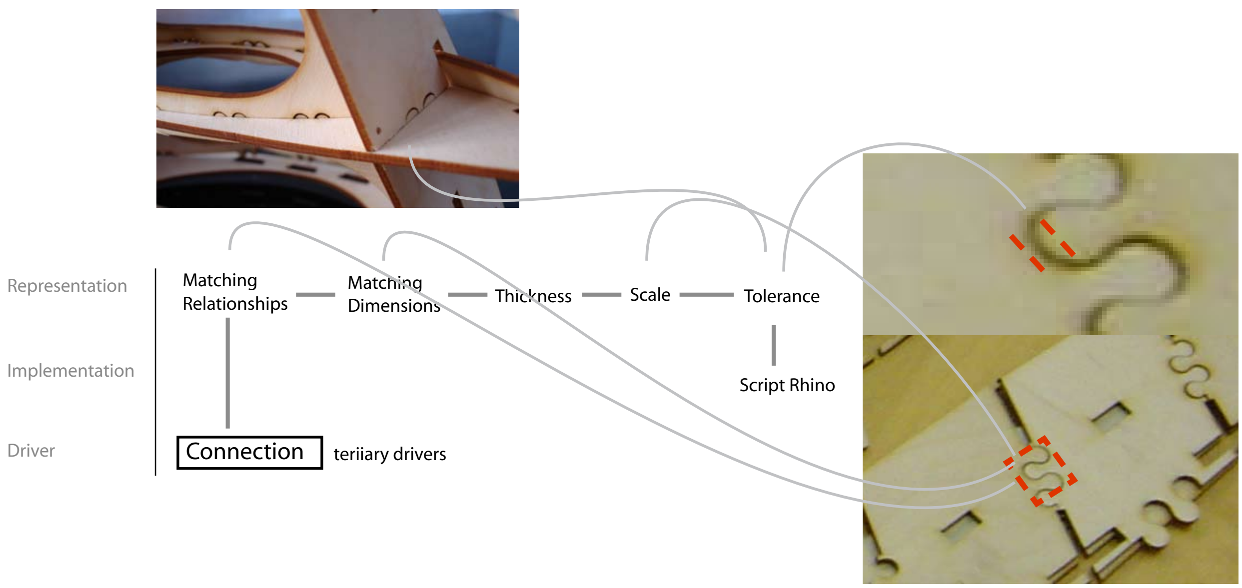


5.1.5.5 Development prototype

With the parametric model it is possible to create partial prototypes to integrate the detailing strategy in to the geometric information. Several iterations were necessary for this step to find the right parametric environment to support the integration of material thickness and parametric adaptive detailing. The challenge is to formulate the approach in a way that captures the essence of the design sketch while adding structural and assembly logic. The initial approach taken proofed insufficient both in aesthetic terms as well as in assembly terms. The transformation created too many details in an otherwise solid appearance while loosing the rib like aesthetic.

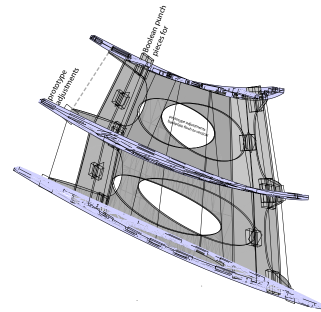
Key points regarding exploration: The establishment of a material

A series of selective prototypes were constructed from the design geometry to test connection constraints in the context of the full assembly. Fine adjustments had to be made through the connection parameters in order to achieve a reliable fit throughout the range of geometric variations of the chair. A mixture of adaptive locally derived parameters and min max material dimensions proved to be most reliable for the final iteration of the joint positioning and scaling.



The connection constraint is a secondary constraint triggered by the implementation choice in the fabrication constraint. Still due to the frequent use of laser cutting a language has been forming around assembly techniques of these parts. For the author this vocabulary is based on the puzzle joint and it plays the role of a design driver here. Its assembly and aesthetic features make it an important part of the overall design. The representation chain links all the dependencies and finally is implemented as a rhino script.

strategy is a key design decision in the process and carries equal importance to the early design geometries. It either carries the design intention or not, it is the first check point of all design constraints rejoining after separate explorations.



5.1.5.6 Detail prototype series

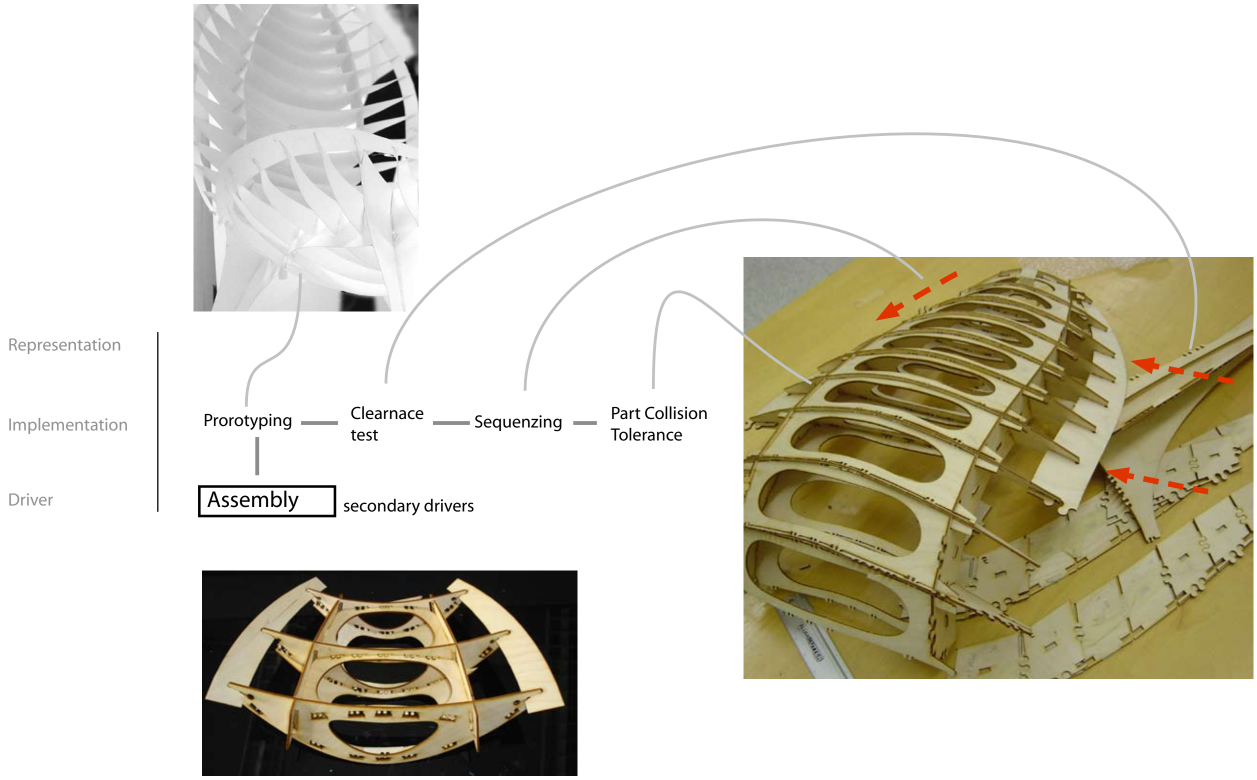
This is an instance of iterative prototyping of a selected portion of

the overall design in order to test the assembly, fabrication, and aesthetic function of the design. Adjustments are integrated into the parametric model. The parametric model acts as the vehicle for the integration

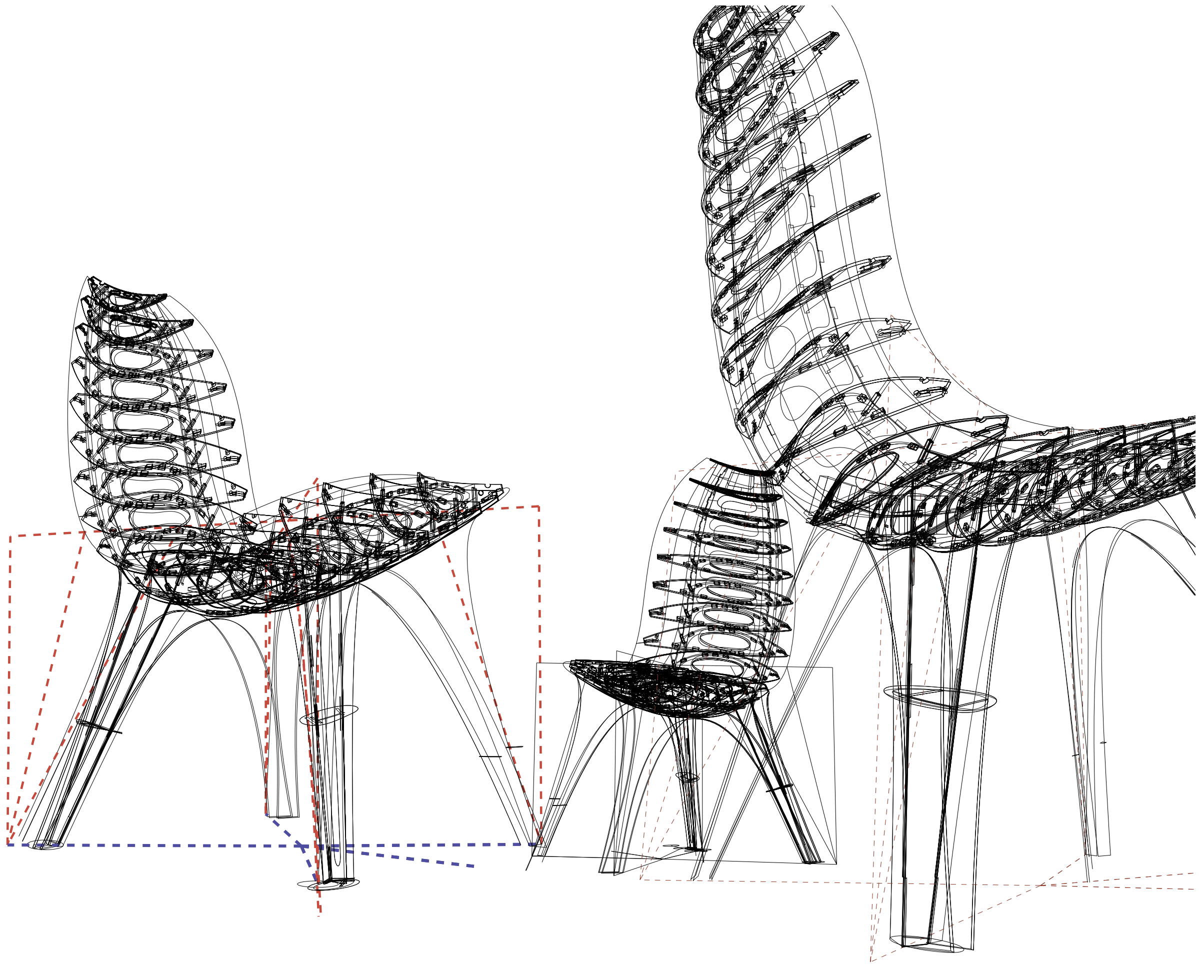
The final assembly challenged all the implementation choices made so far. The initial choice of grain direction proved to be wrong as parts kept breaking. This showed the failure of tight tolerances for assemblies that require force and rely on multiple connections to be made simultaneously. Redesigning the outer rail joins to withstand much greater bending forces through rounding off the inner slot allowed for the close fit tolerance needed for structural integrity while allowing for assembly.

5.1.5.7 Calibration of parametric model

The process of calibrating the parametric model involves the iteration between producing a prototype, analyzing its shortcoming and translating the results into the parametric representation either through changes in the topology or in changes of the dimensional parameters. All parameters affect each other; any change in the geometry has potential influence on the remainder of the model.



Assembly is a constraint at the intersection between the connection and fabrication constraints. Assemblies are one of the most complex constraints to externalize. It is a time based sequence of spatial movements supported by humans or machines. The chair design only needed minor adjustments throughout the first prototypes to reliably accomplish the assembly. The timing was a matter of practicing sequences and adjusting tolerances based on the detected clashes through the physical prototypes.



5.1.5.8 Fabrication preparation

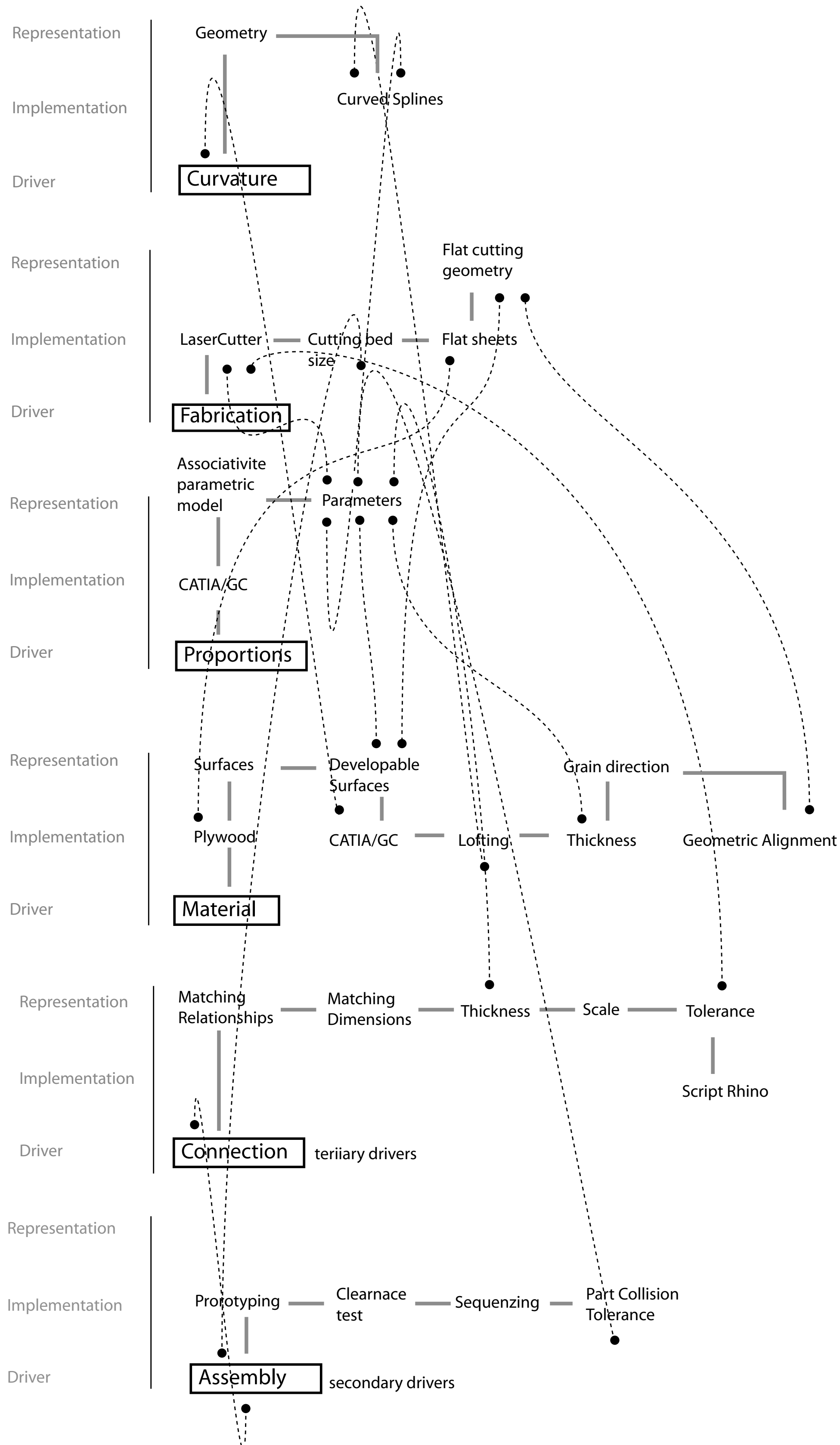
The final prototype is mainly tested for assembly tolerances and grain alignment and preparation of the layout of parts within the material constraints of the sheets available.

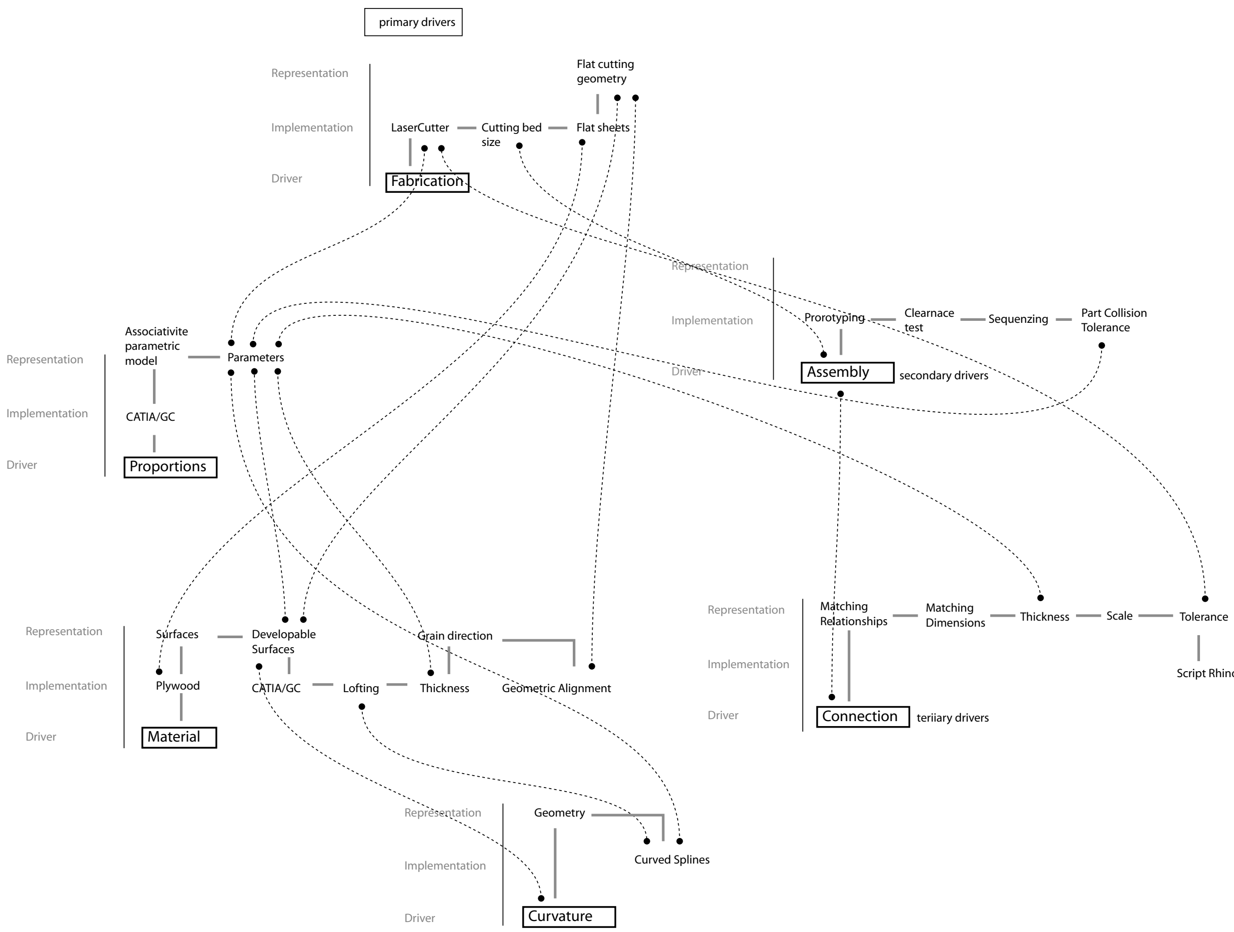
5.1.5.9 Generation of connector detail

Structural integration of the parts relies on two principles combined. One is the friction fit jointing of tangent surface assemblies and second the spatial assembly of parts in an interlocking way that blocks the friction joints from coming undone. This principle propagates through the entire assembly effectively locking all parts in place

One possible chair design.

The complete constraint dependency diagram that identifies the cross dependencies and how the constraints act as design drivers in the overall design.

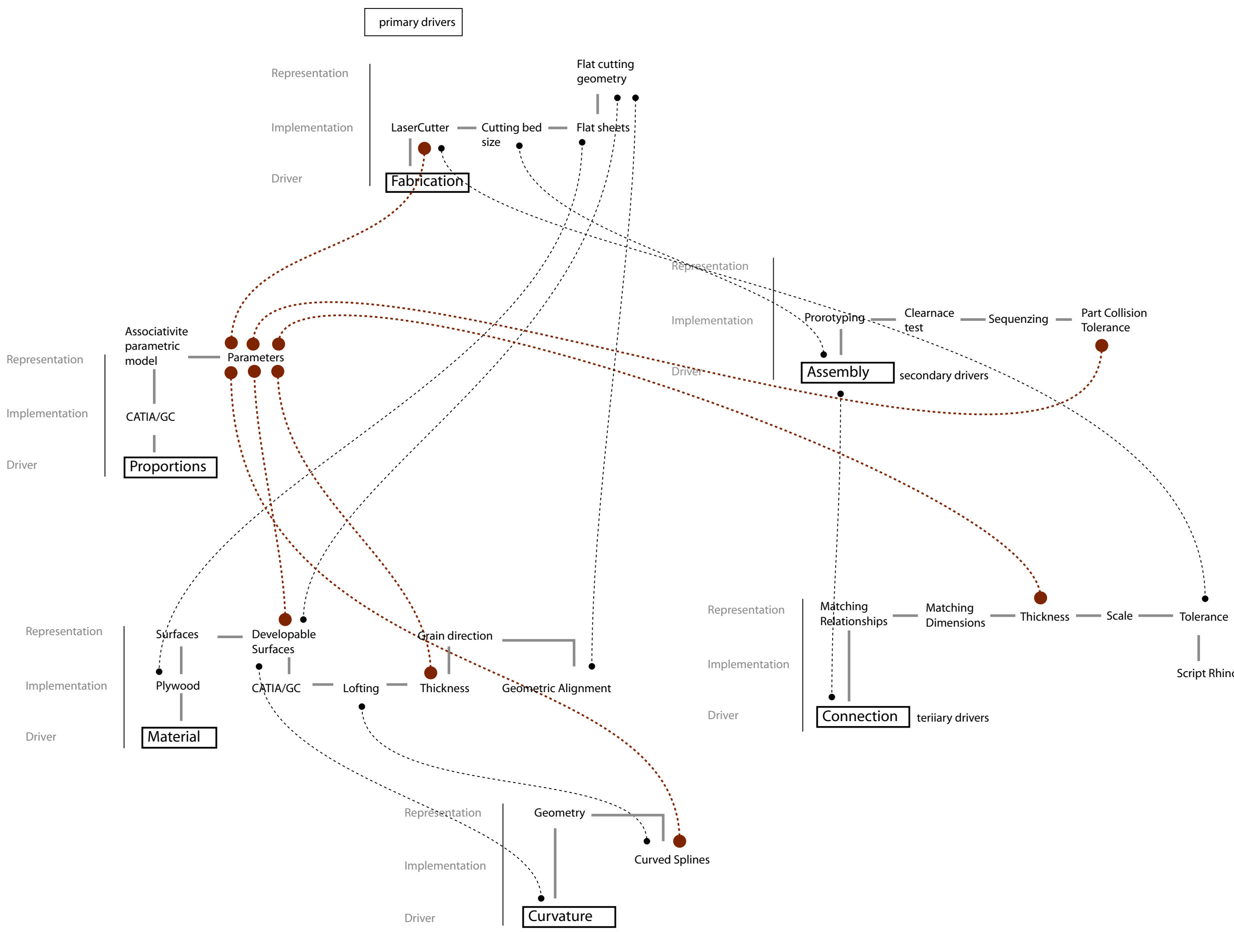

Representation
Implementation
Driver
Geometry
Curved Splines
Curvature
Flat cutting geometry
LaserCutter
Cutting bed size
Flat sheets
Fabrication
Associativite parametric model
Parameters
CATIA/GC
Proportions
Surfaces
Developable Surfaces
Grain direction
Plywood
CATIA/GC
Lofting
Thickness
Geometric Alignment
Material
Matching Relationships
Matching Dimensions
Thickness
Scale
Tolerance
Script Rhino
Connection
teriiary drivers
Prototyping
Clearnace test
Sequenzing
Part Collision Tolerance
Assembly
secondary drivers



5.1.5.10 Assembly

The material properties for the wood chosen mandated the alignment of the grain perpendicular to the direction of curvature to avoid cracks and breakage. The perpendicular alignment meant also that bending the wood required a far greater force and was not possible in many of the tighter locations where many curved parts need to be assembled simultaneously. For those parts light steaming was applied to protect the wood during excessive stress and partially pre-bend the parts in the desired shape beforehand. This treatment does not constitute a full blown shaping of the parts as no molds or steaming equipment was used, it was merely a way to prevent time consuming reproduction of parts. The assembly worked relatively smoothly in the range of several hours. A proper

Partial feedback loops make the exploration more challenging. The loops emphasize the circular nature of the exploration. Changes in a subpart of the design exploration can affect the whole causing adjustment cycle that can be hard to control.

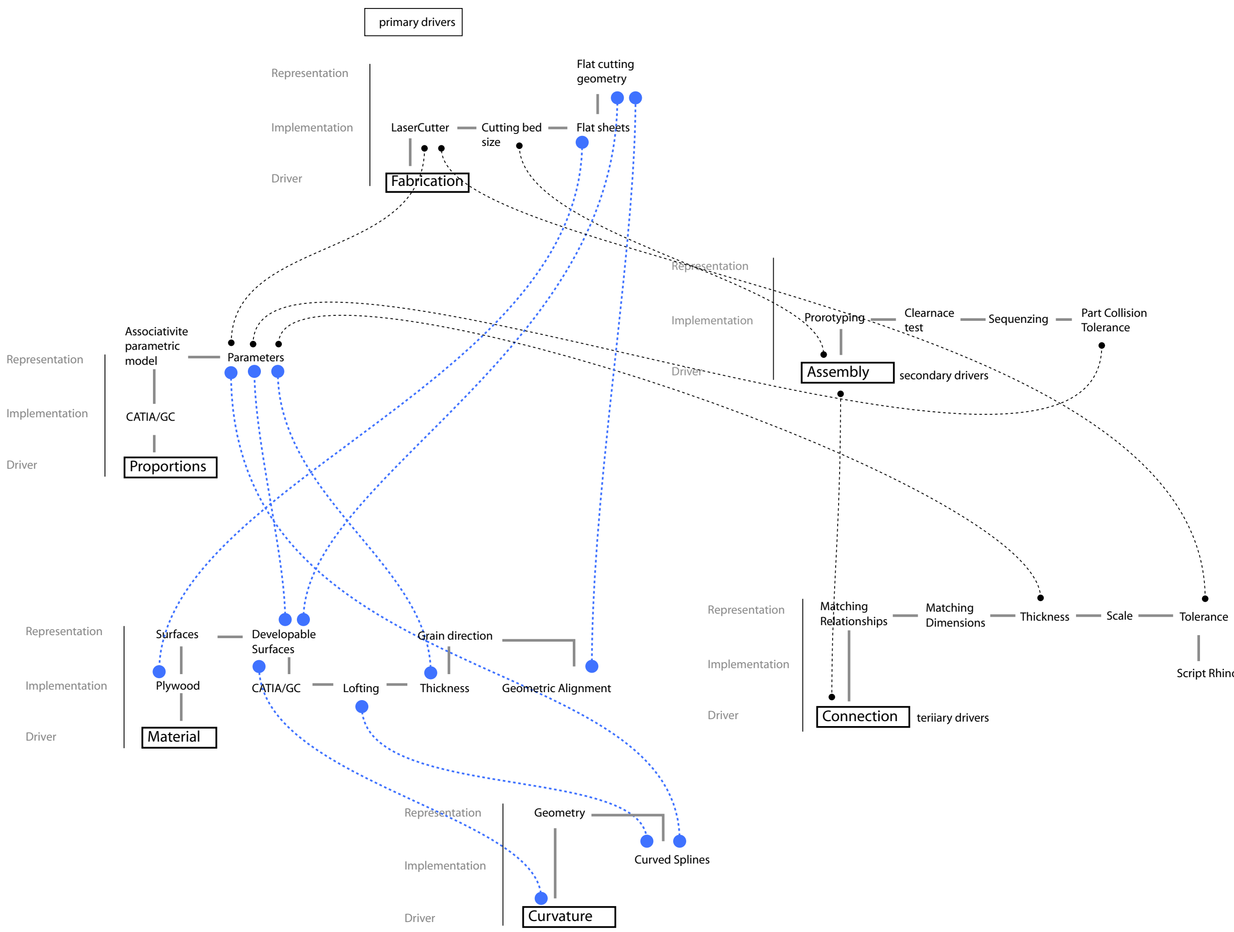


Highlighting individual constraint connections helps to detect the main design drivers among the constraints and the role of representations in gathering or distributing information to other representations. In this case the proportional control skeleton acts as an integrator among several other constraints that are otherwise not directly connected.

assembly sequence needed to be developed from the geometric constraints and eventually worked very well for a single person to assemble the over 140 parts. The biggest challenge were the final edge strips since they required the connection of all parts at the same time. The legs were only partially successful due to them not being fully modeled parametrically. In further iterations they would be fully integrated into the parametric representation.

5.1.6 Analysis of the Projects

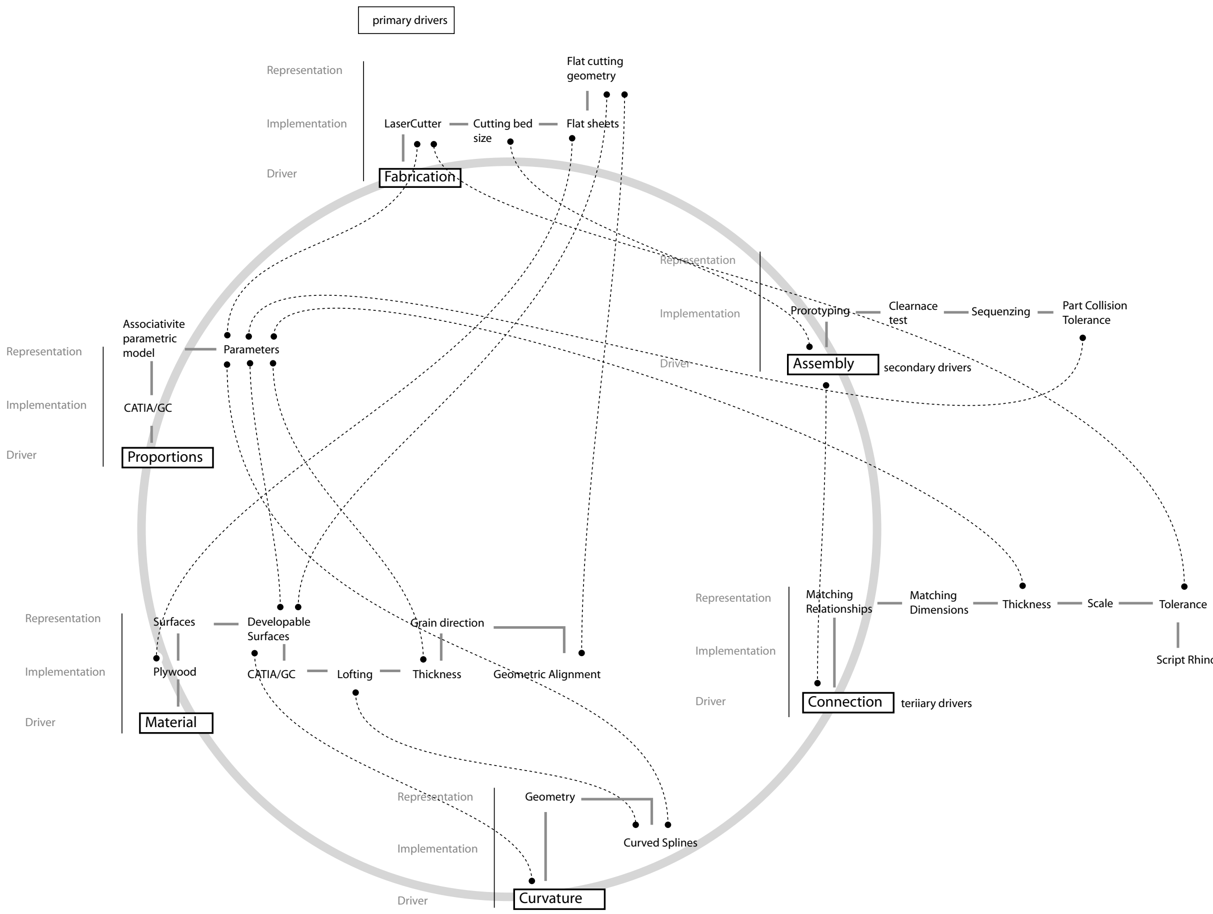
Surfaces play an increasingly important role in design representation and also led to the increased reliance of architectural representation on geometry overall. This has led to many of the frustration and misunderstandings in design in the digital context. At the core



of the geometric representation stands the surface. Despite the importance of solid based modeling in engineering and architecture the surface is still at the core of design representation as even solids are composed of surfaces and a considerable percentage of design software does not implement full solid modeling function bout rather relies on surface representations The surface condition has many facets and only few of them are captured in digital surface representation.

The separation of sculptural expression and translation into a buildable system led to some of the most stunning buildings

The material connections act as integrators between curvature and fabrication.

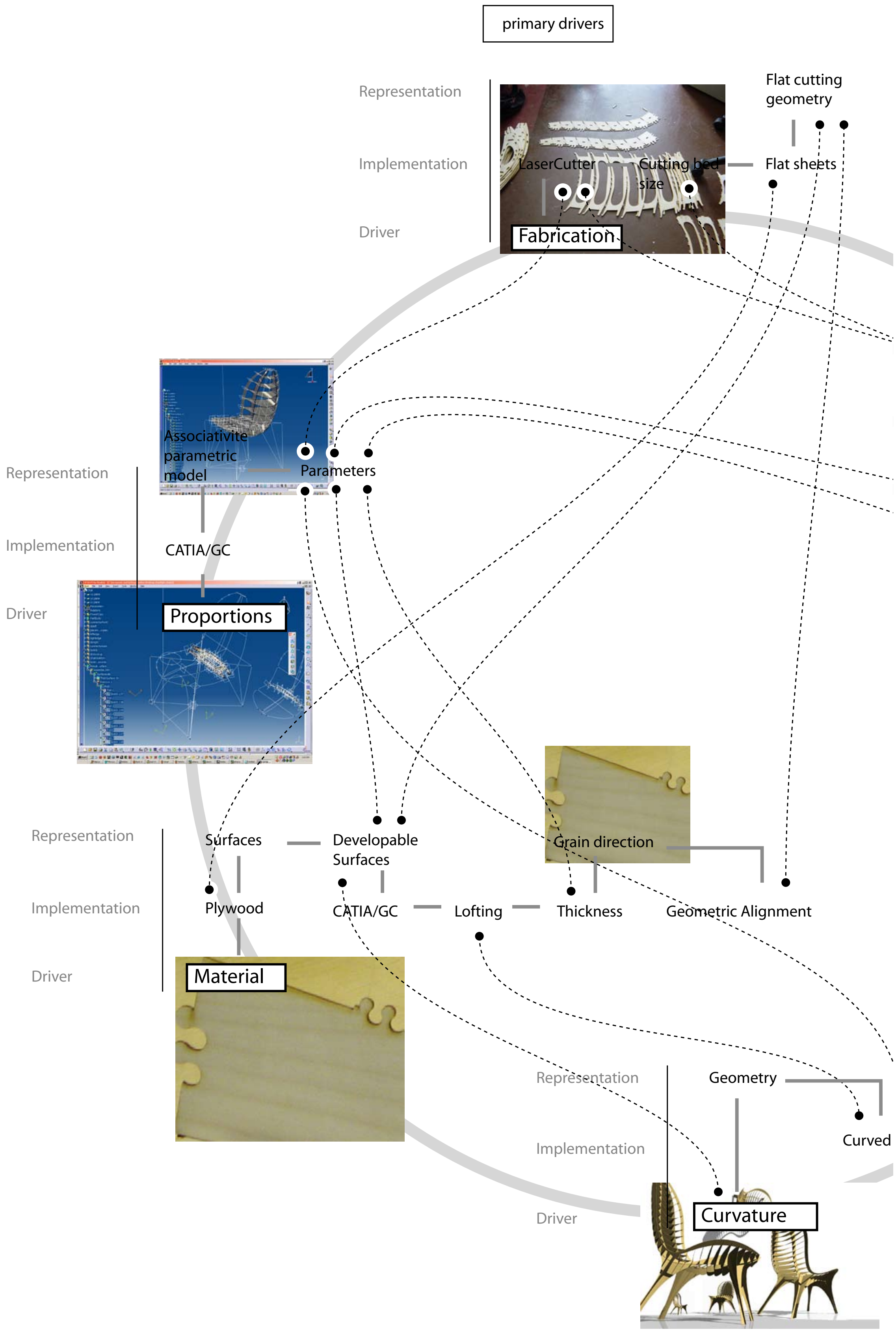


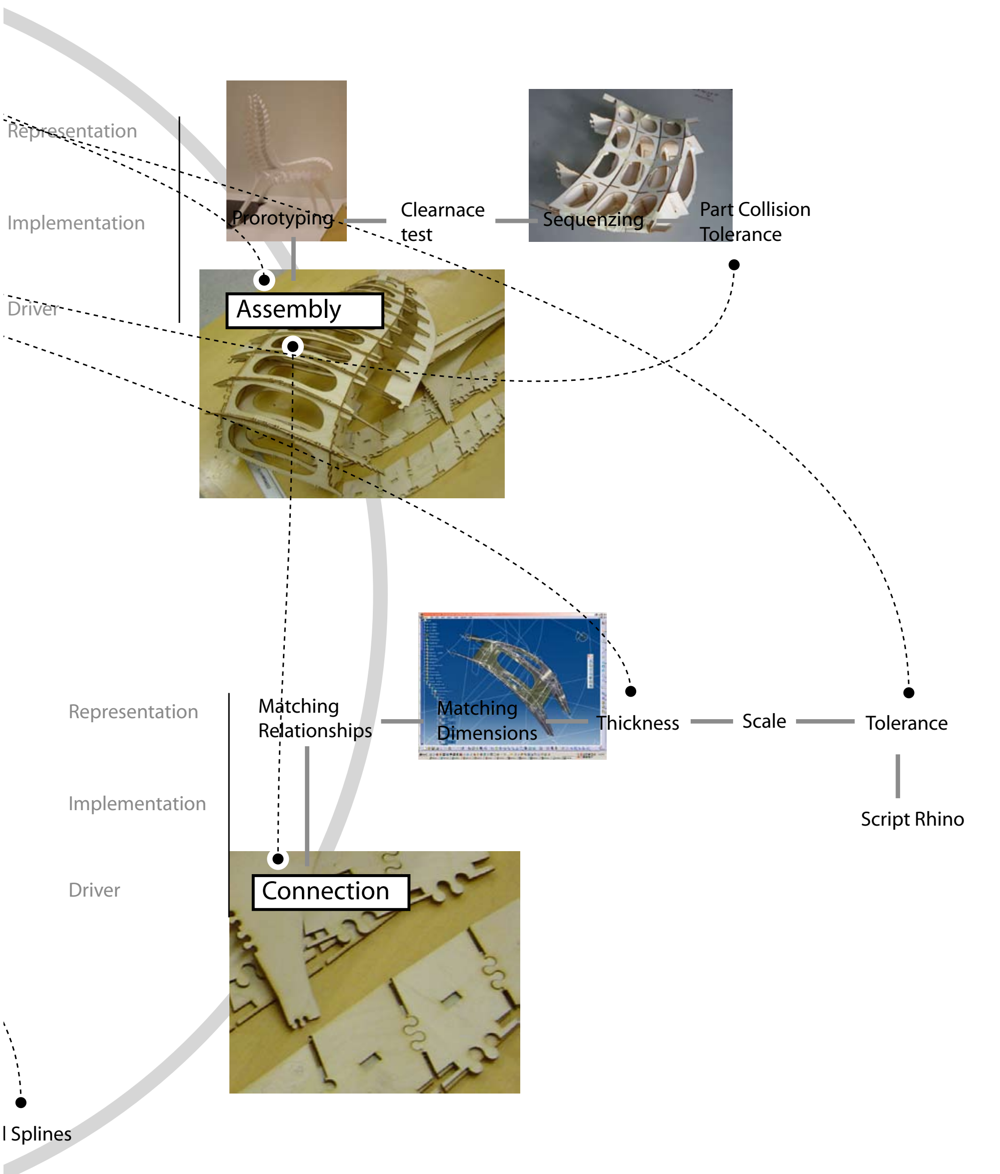
If one follows the links several circular dependencies can be identified amongst the constraints. This makes it harder to model them as design drivers as there might be feedback loops.
For instance a change in material may require a different fabrication choice which may affect connection and assembly

such as Bilbao or Walt Disney Concert Hall. But at the same time it demonstrated the lack of domain integration into the initial design process. And I am not arguing for taming the design process but rather to support and embedded a structural and material sensitivity into the design process. Not so much in its imagined form but in its manageable form that allows for the precise capturing and post-processing of the desired form in a digital system.

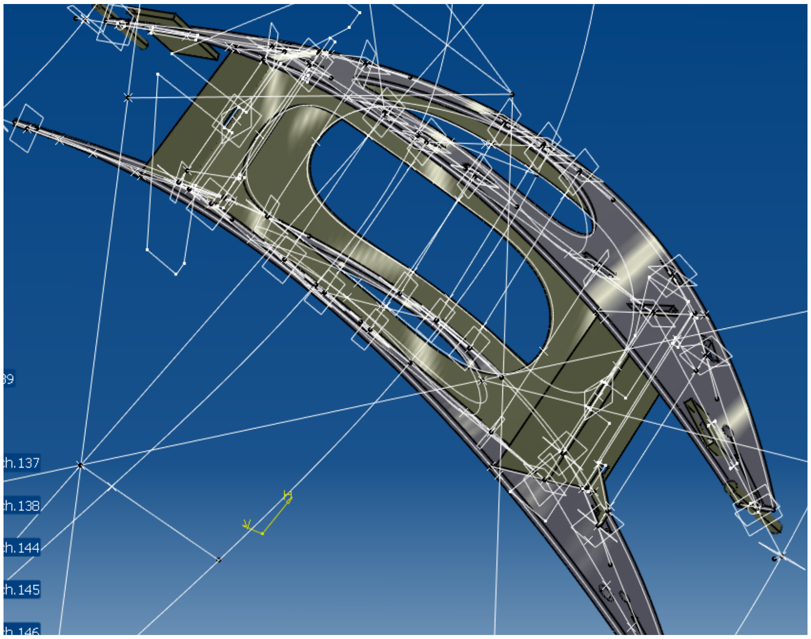
5.1.6.1 Conclusion

The chair case study was successful in developing a partially functioning prototype of a chair exploiting the possibilities of curved wood surfaces in a structural assembly. The parametric model as the repository for the prototyping revisions proofed


primary drivers
Representation
Implementation
Driver
LaserCutter
Cutting bed size
Fabrication
Flat cutting geometry
Flat sheets
Associativite parametric model
Parameters
Representation
Implementation
Driver
CATIA/GC
Proportions
Representation
Implementation
Driver
Surfaces
Developable Surfaces
Grain direction
Plywood
CATIA/GC
Lofting
Thickness
Geometric Alignment
Material
Representation
Implementation
Driver
Geometry
Curved
Curvature



The complete constraint network with implementation and representation choices.



Parametric component with full connection details.

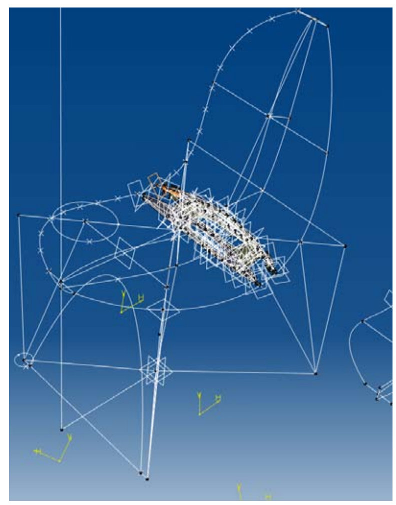
Final full scale chair assembly in plywood.

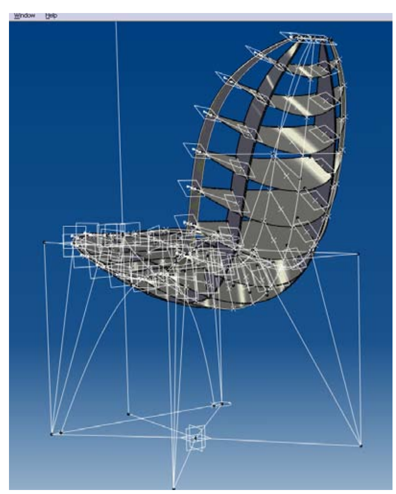
Parametric models of the control skeleton and the parts riding on it.

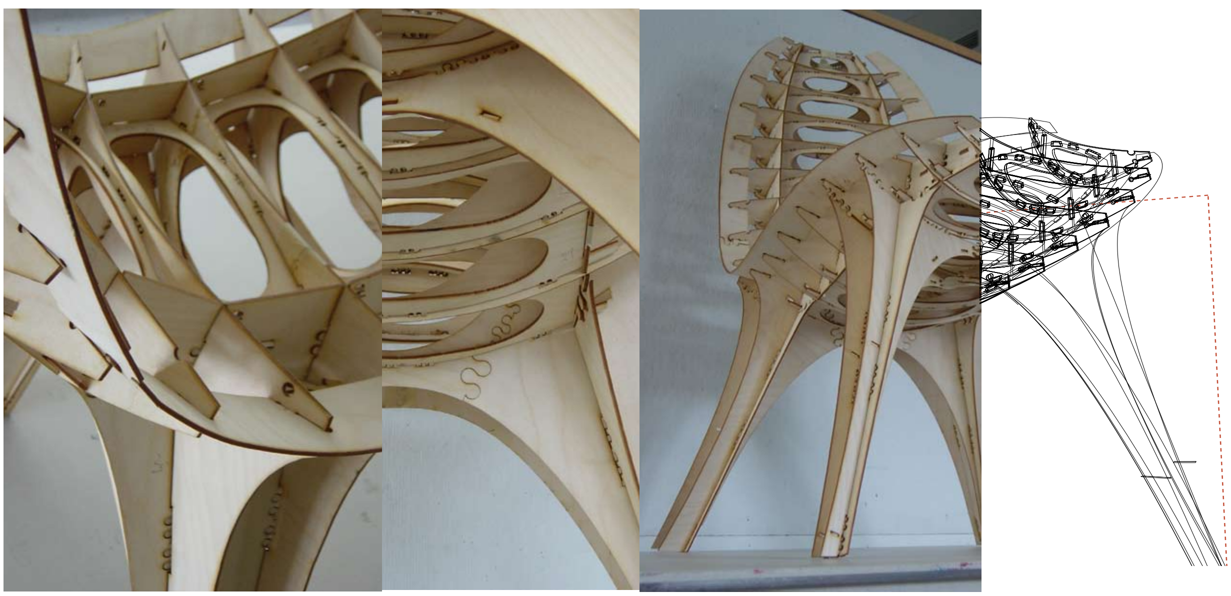
A surface model with reduced detail for evaluation of proportions and positioning of parts.

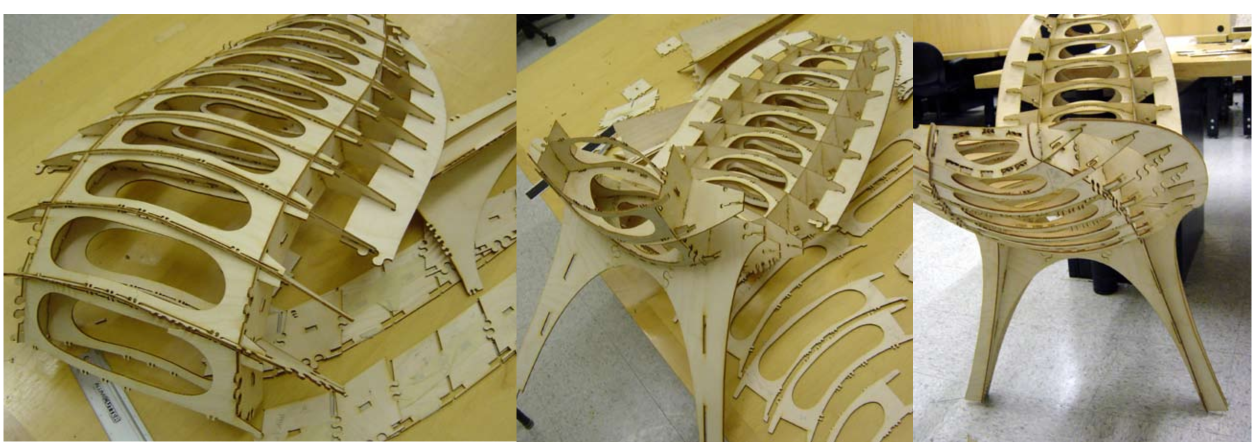
Detail studies and physical-digital comparison

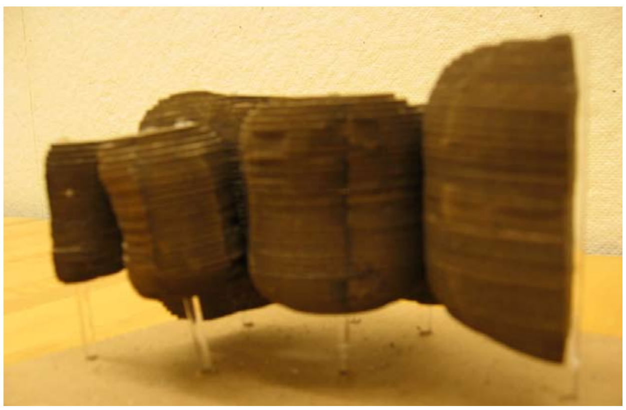
(right) Assembly steps for legs and seat part.

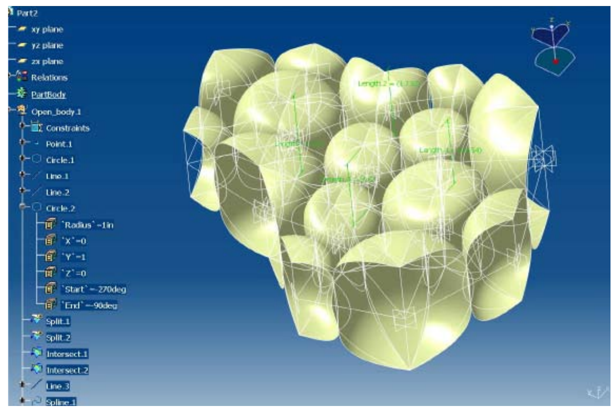












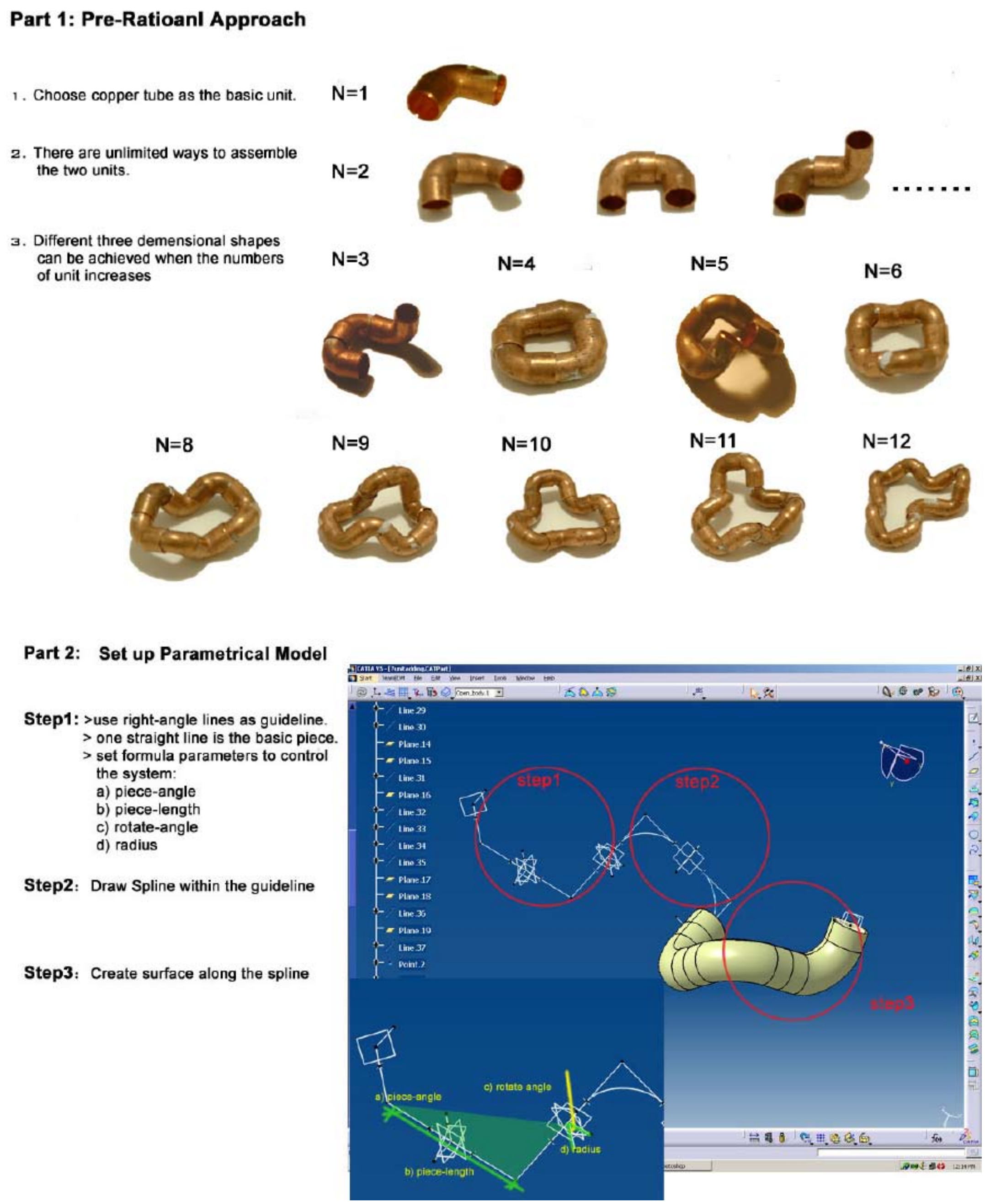
Ceramic workshop in collaboration with IST Lisbon, 2002.
Instructor: Larry Sass , TA Axel Kilian, RA Carlos Barrios.
The exercise called for the description of a system of interacting parts to be parametrically modeled in CATIA based on a physical precedent.
Min Cho worked on a "mushroom like" tile study modeled in CATIA.

Image: Min Cho

successful. The parametric model's functionality to support the circular properties of the project are limited though, as it can only receive information fed to it through the input parameters but was not connected to any other representations.
It is difficult to argue for a general model for instance for material simulation. This specific case did not use the material for the realism of its representation but rather as a design rational.

5.1.6.2 Circular dependencies – effects on design – disciplinary barriers

The case study is part of this thesis as the circular dependencies observed in the study and in the studies leading up to it are present in all projects no matter what scale or domain and managing them properly can make the difference between success and failure of a project. Beyond the management of a project the circulars main



4.182 ceramic workshop in collaboration with IST Lisbon. Fall 2002
Study of a parametric relationship, for which she chose a copper tube fitting system and developed an interlinking mesh from it.
Image: Xin Tian

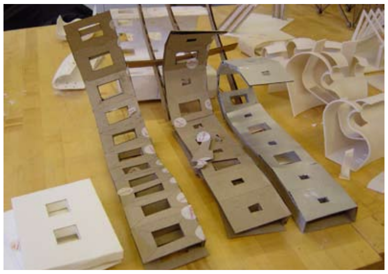
research interest is the potential for triggering aesthetic potential and synergies between the different domains adjacent to the design.

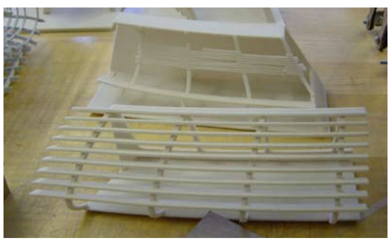
5.1.7 Teaching Examples

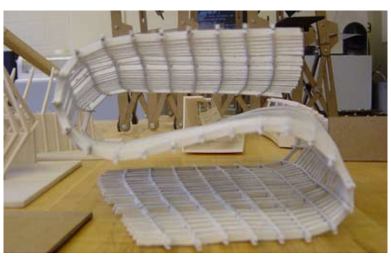
5.1.7.1 Workshop on Parametric Modeling and Ceramic

Construction – MIT in collaboration with IST Lisbon

A workshop focused on conveying principles of parametric modeling in the context of ceramic construction systems was taught in the fall of 2002 in collaboration with the *Instituto Superior Technico* in Lisbon, Portugal. The workshop was headed and organized by Larry Sass and was taught in collaboration with Jose Duarte, Luisa Caldas and Joao Rocha (IST). The author contributed



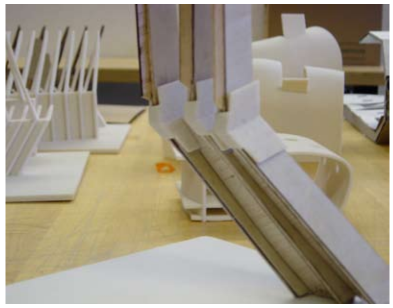


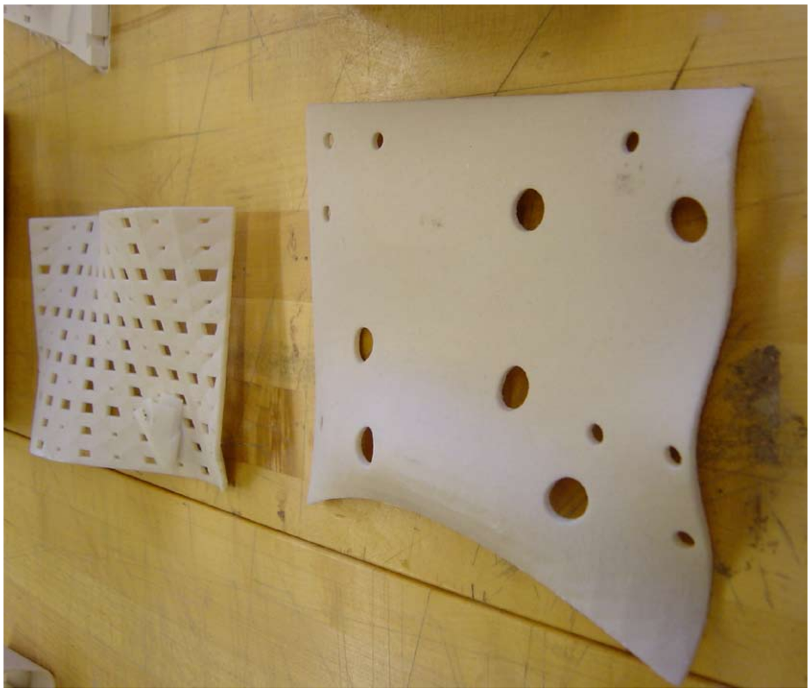


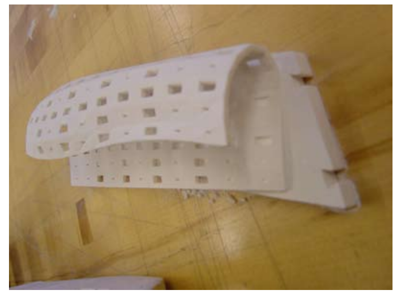
as a teaching assistant. He was responsible for the first exercise, a system analysis based on CATIA©.

The author structured the first half of the semester with design exercises that focused on the analysis of systems and the modeling of these systems and their interaction in a computational fashion. One student chose as her building component copper tubing bends and developed form the analysis of their interaction different schemes of how to extend their intended use into that of a building skin. She developed different types of meshing strategies to achieve this. The goal of this exercise was to teach the students parametric associative modeling through the careful analysis of an existing system, thereby challenging them to adopt an alternative digital modeling approach to geometric descriptive modeling than

Fabrication workshop Spring 2003. "Light in architecture" using parametric and generative tools. Instructors: Axel Kilian, Larry Sass, Terry Knight. RA Carlos Barrios, Yanni Loukissas, TA Aaron Greene, in collaboration with Hugh Whitehead, Judit Kimpian and Francis Aish, from Foster and Partners..
Work shown by Sawako Kaijima, Alejandro Zulas, Stelios Dritsas and Sameer Kashyap.
On the right the lighting test box for the facade exercise.





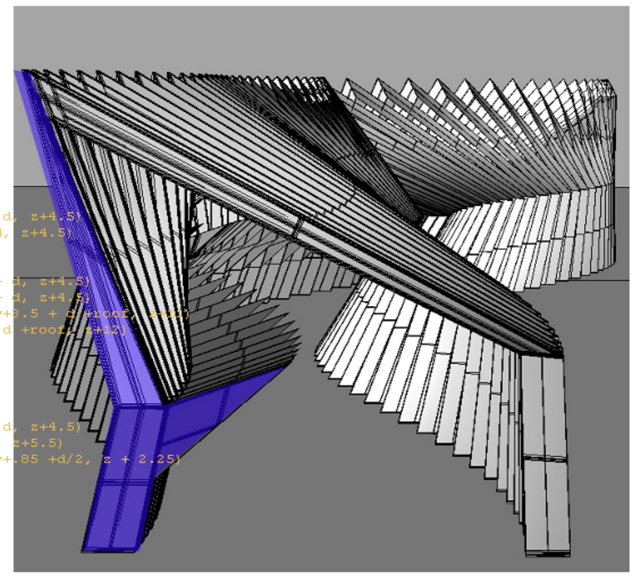


the one common in standard CAD platforms.
This approach was partially successful in introducing students to the following assignment which required them to work in groups in the much more complex context of a full building façade. The danger in combining a complex design task with a novel software approach is that there is no time to understand both simultaneously

Fabrication workshop Spring 2003. "Light in architecture" using parametric and generative tools. Instructors: Axel Kilian, Larry Sass, Terry Knight. RA Carlos Barrios, Yanni Loukissas, TA Aaron Greene, in collaboration with Hugh Whitehead, Judit Kimpian and Francis Aish, from Foster and Partners.

Three dimensional prints of the parametric and script generated structures for testing in light. Work by Rita Saad and Maria Thompson, Sawako Kaijima, Victor Gane.

```
function louver1 (point, roof)

  redim toptemp (1)
  dim strObject
  dim strObject2
  dim strObject3

  dim x : x = point (0)
  dim y : y = point (1)
  dim z : z = point (2)
  dim d : d = 2.5

  redim blade (3)
  blade (0) = array (x, y, z)
  blade (1) = array (x, y + 1.2, z)
  blade (2) = array (x + d + cos(x/10.0), y+.5 + d, z+4.5)
  blade (3) = array (x+ d + cos(x/10.0), y-.5 + d, z+4.5)

  redim blade2 (3)
  blade2 (0) = array (x + d + cos(x/10.0), y-.5 + d, z+4.5)
  blade2 (1) = array (x + d + cos(x/10.0), y+.5 + d, z+4.5)
  blade2 (2) = array (x + .5 + d + cos(x/10.0), y+3.5 + d +roof, z+11)
  blade2 (3) = array (x + d + cos(x/10.0), y+3 + d +roof, z+12)

  arrtoppt (0) = blade2 (2)
  arrtoppt (1) = blade2 (3)

  redim blade3 (2)
  blade3 (0) = array (x+ d + cos(x/10.0), y+.5 + d, z+4.5)
  blade3 (1) = array (x+ d+ cos(x/10.0), y+3 + d, z+5.5)
  blade3 (2) = array (x+ d/2 + (cos(x/10.0))/2, y+.85 +d/2, z + 2.25)

   strObject = Rhino.AddSrfpt (blade)
  strObject2 = Rhino.AddSrfpt (blade2)
  strObject3 = Rhino.AddSrfpt (blade3)

  call thickness (strObject, arrtoppt)
  call thickness (strObject2, arrtoppt)
  call thickness (strObject3, arrtoppt)
  Rhino.DeleteObject strObject
  Rhino.DeleteObject strObject2
  Rhino.DeleteObject strObject3
```



Project by Maria Thompson, RIta Saad shows script and one generated structure .
Image: Maria Thompson and Rita Saad

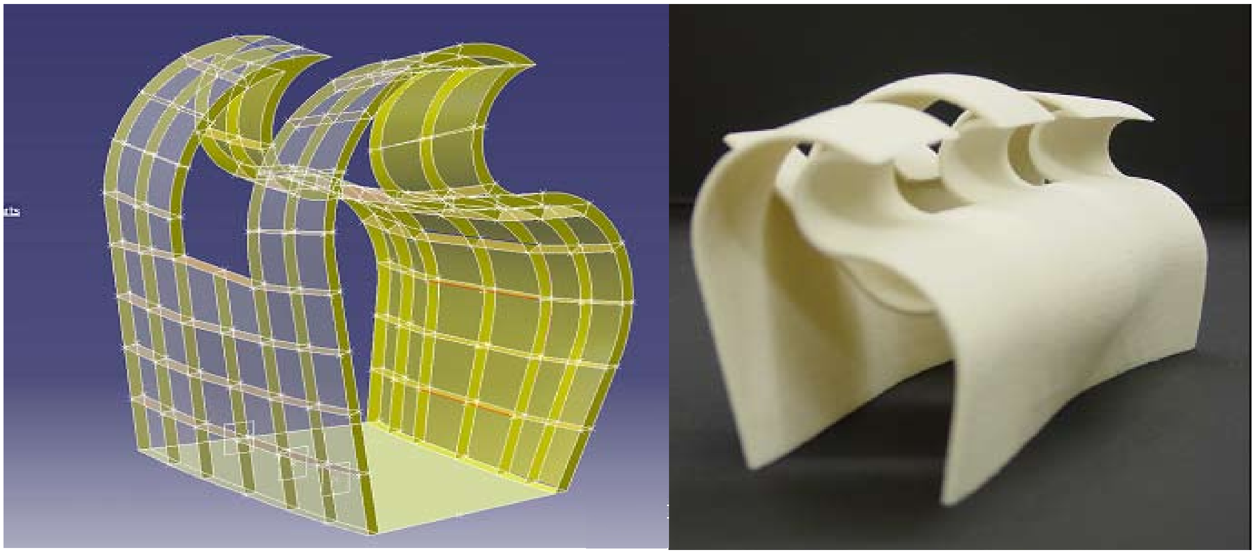
(top)Video conference session with students and Hugh Whitehead, Judit Kimpian and Francis Aish.

Image: Federico Casalegno

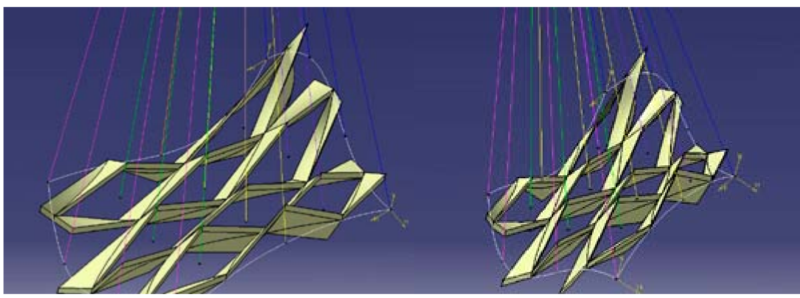
and students revert to the processes they are accustomed to without learning the potential and shortcomings of the new approach.

5.1.7.2 Workshop on Parametric and Generative Methods – MIT in collaboration with Foster and Partners

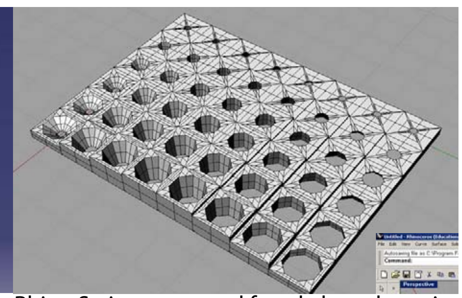
In the spring of 2003, a second workshop was taught that focused on the teaching of parametric and generative methods in design.



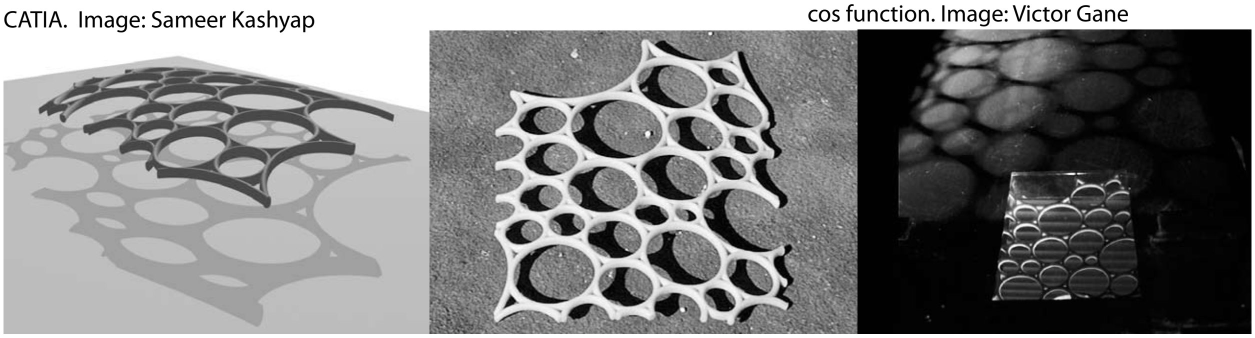
Project by Victor Gane. A gallery section showing light modulating louvers. The corridor is parametrically adjusts to context and artwork. Image:: Victor Gane



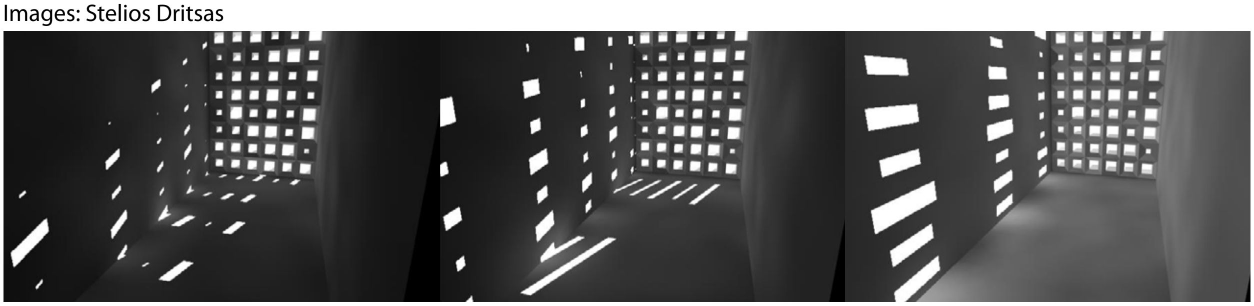
Project by Sameer Kashyap louver studies responding to daylight direction in CATIA. Image: Sameer Kashyap



Rhino Script generated facade based on sin cos function. Image: Victor Gane



Project by Stelios Dritsas using CATIA to model a water like shading surface. Light study in the lighting box
Images: Stelios Dritsas



Project by: Sawako Kaijima, a script generated facade panel that is based on neighbor inhibition. If one cell is open it surpresses its neighboring cells to be open as well. The emergent pattern were studied in the light box. Image: Sawako Kaijima

Smart Geometry workshop series 2003-2006. Image above shows the first workshop in June 2003, at the Moeller Center, Cambridge, UK. Founding members of the Smart Geometry group: Lars Hesselgren, Robert Aish, Hugh Whitehead, Jay Parish. Tutors: Axel Kilian, Chris Williams, Marty Dorscher, Francis Aish. Image: Smart Geometry.com

The author was co-instructor together with Larry Sass and Terry Knight. Research assistants were Yanni Loukissas and Carlos Barrios, Aaron Greene was the teaching assistant. At Foster and Partners, Hugh Whitehead, director of the specialist modeling group, Judit Kimpian, and Francis Aish were collaborators.

The second workshop was much faster paced and explicitly focused on teaching parametric and generative modeling techniques in the context of fabrication. The first exercises focused on the theme of the workshop, the use of daylight as a design driver in architecture. The goal was to generate façade panels that modulate light and fabricate them for physical testing in a lighting box. A range of approaches was taken and it was particularly interesting to see how parametric models and script generated models resembled each

Free form roof exercise with hierarchical control geometry and component population
–by Axel Kilian September 2005.
–edited by Rob Woodbury October 2005.

The goal is to demonstrate a number of Generative Component features acting in concert to produce a relatively complex model.

The objective is to create a curved roof following a spatially curving path in the landscape using only four top level control points. The resulting model has several levels of structure or hierarchy demonstrating how parametric modeling can yield conceptually clear and easy to manipulate models comprising complex geometric features.

Step 1. Create a new GCScript file by press the new script file button

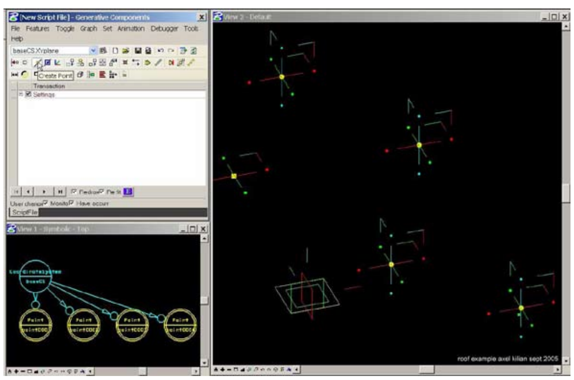
- Arrange the windows in such a way that they don't overlap and both the symbolic view and the geometric view are visible as well as the GC control panel

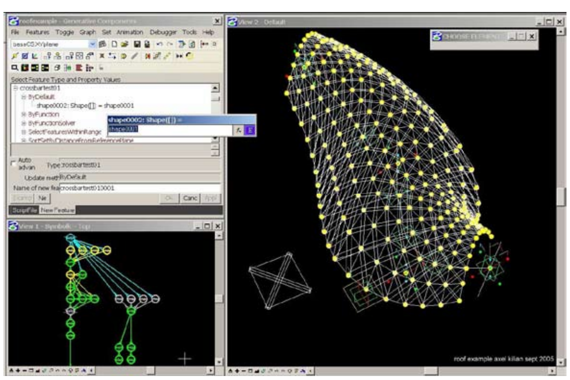
Step 2. Save the file with an appropriate name for instance *roofexample01.gcs*. Do this by clicking on the save icon.

Step 3. Familiarize yourself with the view rotation in the bottom tool bar of the geometric view window

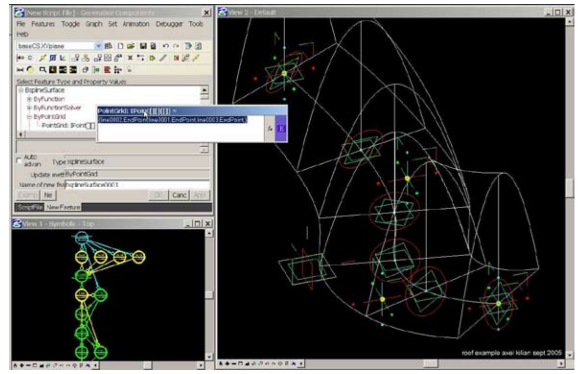
- First, zoom to fit the model to the window. The icon that looks like a mountain in profile does this. With an empty model, you will get a single coordinate system. If you don't do this you are likely to create models far from the origin, which makes futures viewing manipulation awkward.
- To zoom in and out there are the + and – buttons at the bottom of the window pane. There is also an arrow that pans. You should zoom out a few times so that you have space around the coordinate system in which to work.
- The turquoise spinning arrow is for view rotation. To get back to an orthographic view select the spinning arrow and switch from dynamic to "top" in the pull down menu. This is Microstation functionality
- To shade a view click on the lightening bolt and select the desired shading mode. Most users alternate between wireframe and shaded views.

Step 4. To begin the model select the point creation short cut (see highlight icon in image below) and place four points in an axonometric view into the geometric model in a zigzag fashion. These points are the base for the path that winds through the landscape. (We skip modeling the landscape here for time's sake. Such can always be added later and the four points can be tied to the landscape.)





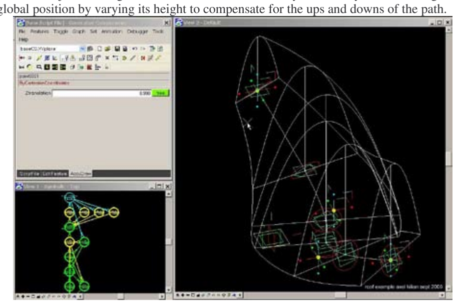
Rendered and hidden



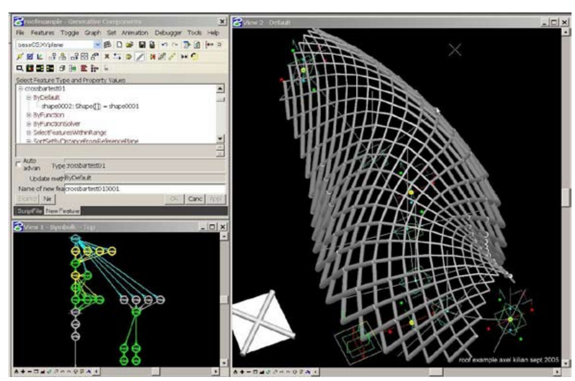
Step 10. Currently the surface is constant height and width, to make it more interesting we can make it responsive t the undulation of the path. To do so we use the edit command and replace the fix length of the vertical line of the sections with an expression that calculated the height as a function between a maximum height of the roof and the height of it base point.

Key Idea:	It is easy to set a fixed initial value and later replace it with a dependency on other values.

- Edit the vertical line by clicking on the edit tool and then the line
- In the length field of the line subtract the Z value of the plane0001 from the fixed value currently ther for instance if it was 4 Length=3-plane0001.Z
- This will adjust the height of the roof so that it will always be exactly reach the height of absolute 3 in global position by varying its height to compensate for the ups and downs of the path.

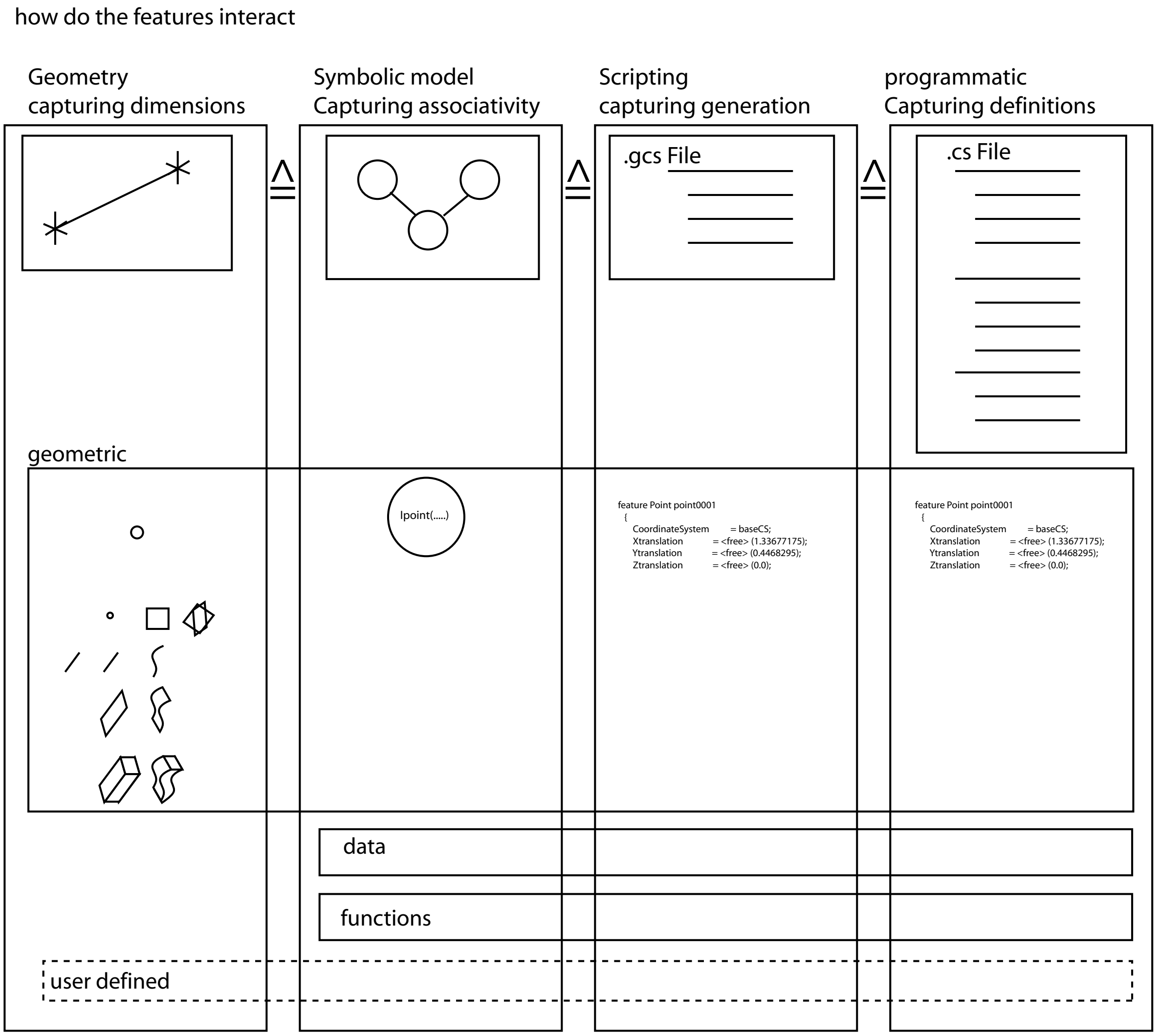


Step 11. Play with the move command to vary the initial four points height and position to see the roof respond. W now have a surface, which we can use to place the elements forming a roof.



That's it for now. In order to be able to move the initial points you may want to toggle the BSpline surface to

Tutorial material for teaching modeling based parametric constructs. Example shown is a responsive roof example.

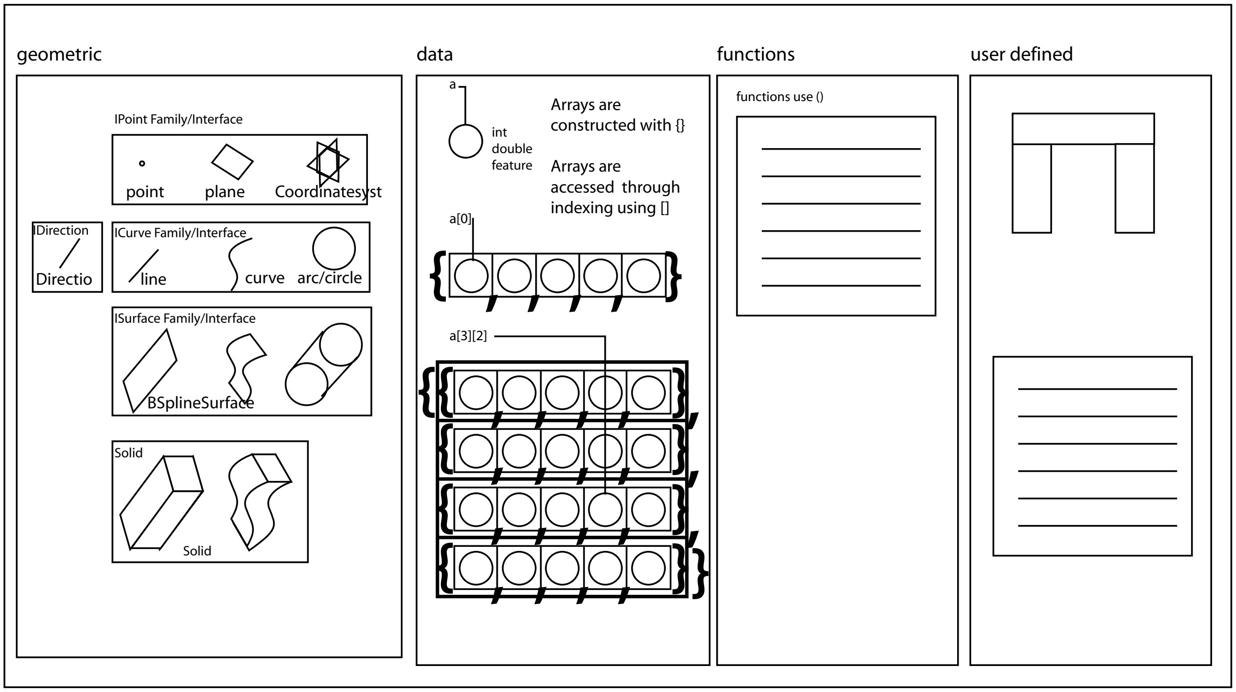


The parallel structure of interfacing with generative components.
The program structure allows the extension of the runtime environment from any of the four interaction modules shown here.
Manual modeling, associative editing, scripting, and programming in C#.

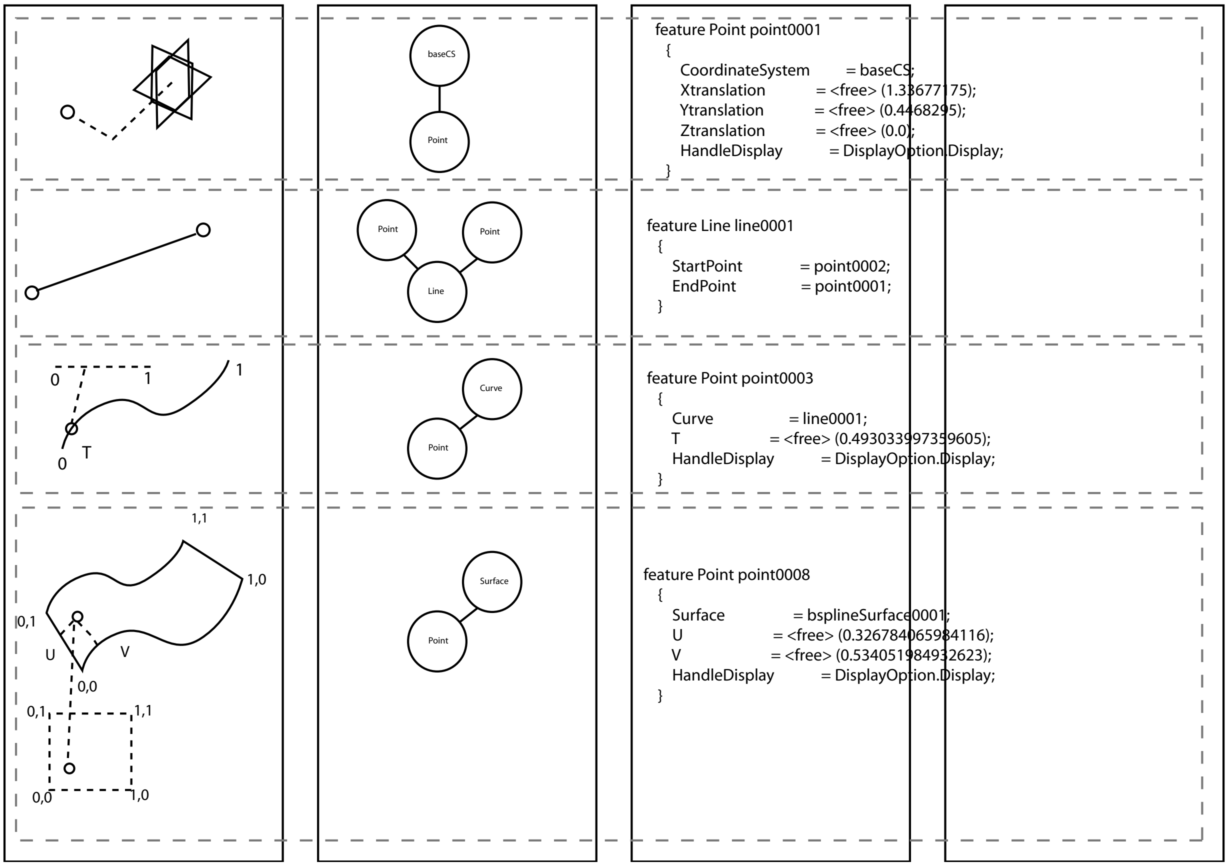
other or differed. The biggest challenge was to translate the design intent into the unfamiliar language of scripting and associative parametric modeling in combination with having to produce geometry that could be used for fabrication. (Loukissas 2003)

5.1.7.3 Smart Geometry workshops

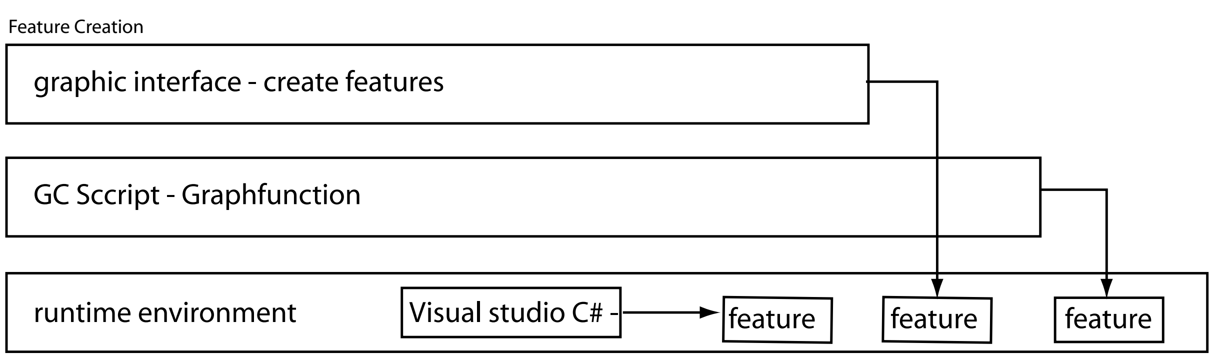
Over the course of several years starting 2003, the author has taught, together with the Smart Geometry Group, generative and parametric modeling concepts to both students and practitioners in universities, workshops, and at conferences. The founding members of the Smart Geometry Group are Robert Aish, Lars Hesselgren, Hugh Whitehead, and Jay Parish (www.smartgeometry.org). The platform for working was Generative Components, a parametric primitives -features



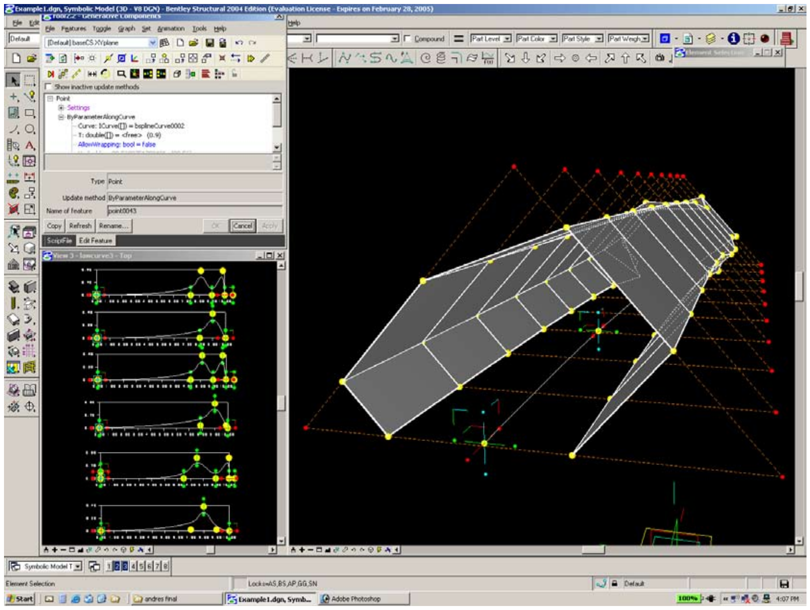
The primitive features in the system available to the designer. There are geometric elements like points and lines , surfaces and solids. There are also data primitives such as single values, and single and two dimensional arrays. Script functions are also primitives in the system as well as programmatic features. All together they define in parallel the interface for the designer and programmer to interact with the system and extend it.



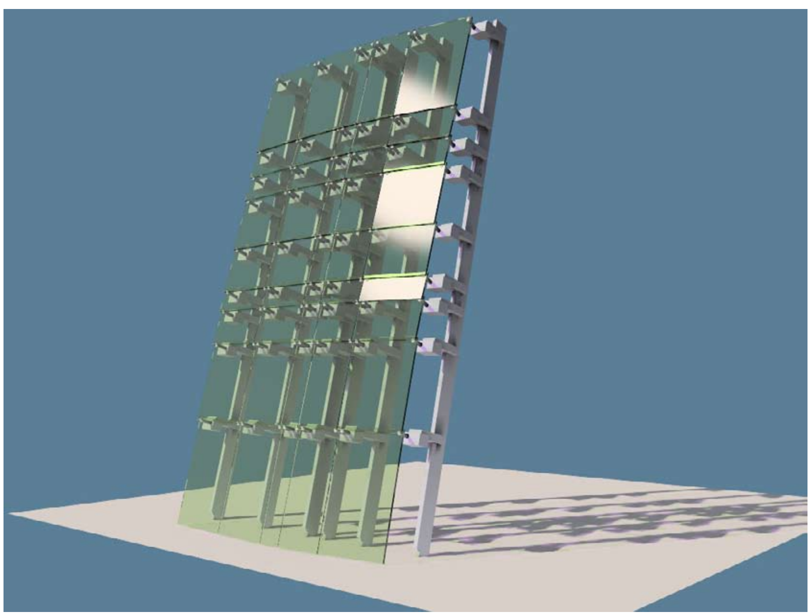
How the approaches differ in defining the same features. The first column shows the geometric relationships, the second one the associativity and the last two the scripting and programming approach for defining a feature.



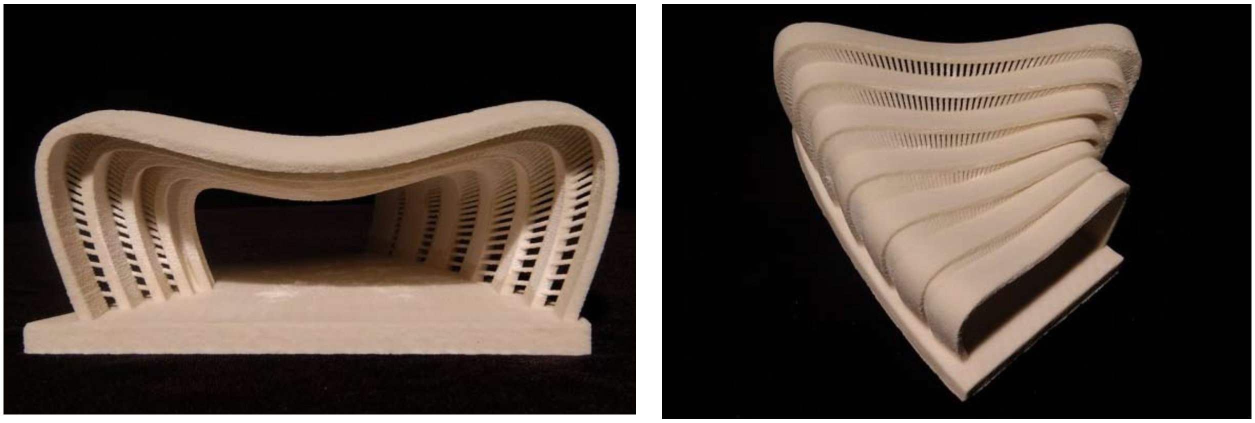
Extending the runtime environment through graphical interface created features, through script ed features and through programming based features.



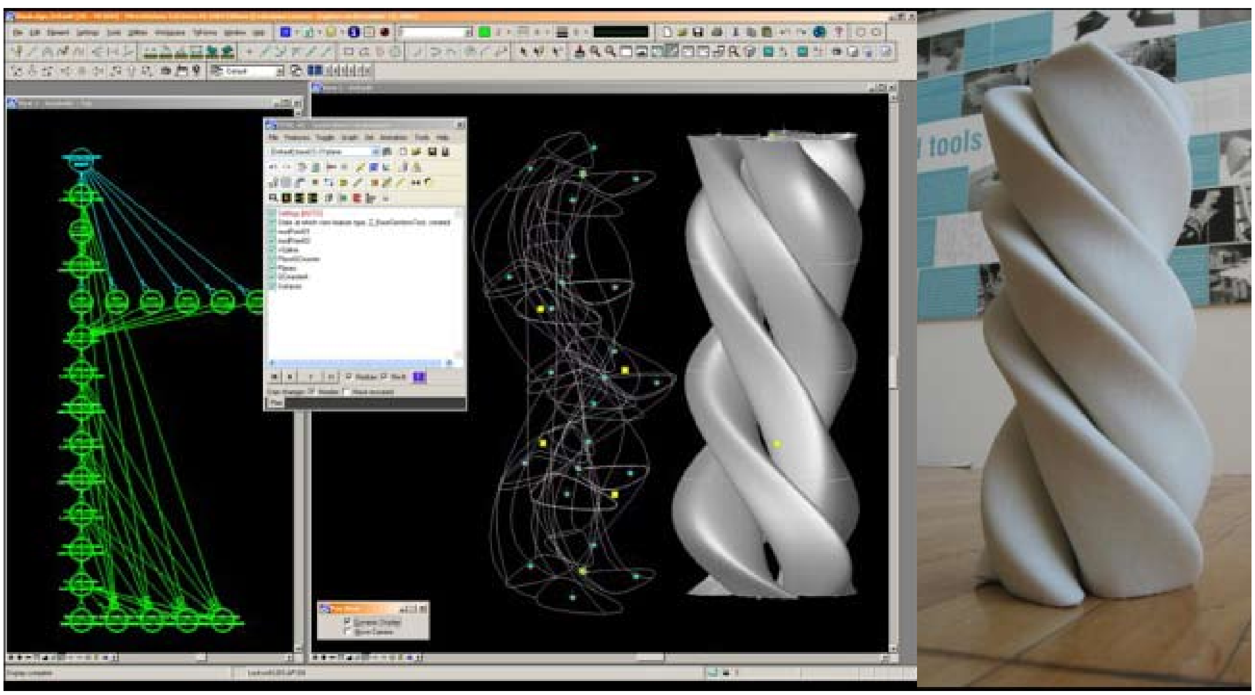
2005 IAP MIT GC Workshop study by Andres Sevtsuk.Interior roof study for an existing building that generates geometry from the position of a set of attractors traveling the length of the roof.
The use of graphs made the interaction with the geometry more diagrammatic.
Image: Andres Sevtsuk



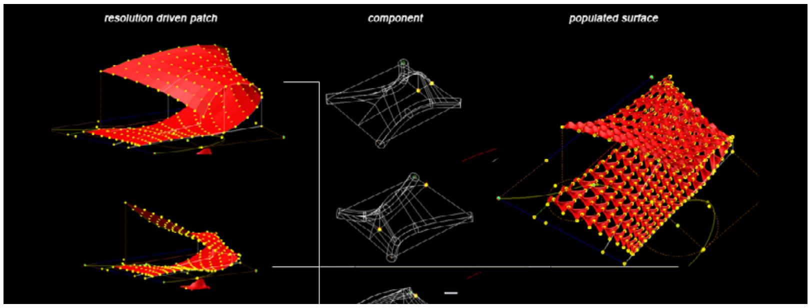
2005 IAP MIT GC Workshop study by Jeff Andersen. Glass facade solution that allows the adjustment of the glass pane subdivisions parametrically.
Image: Jeff Anderson

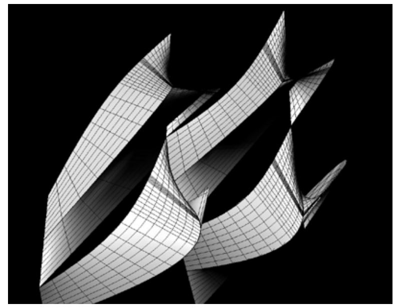


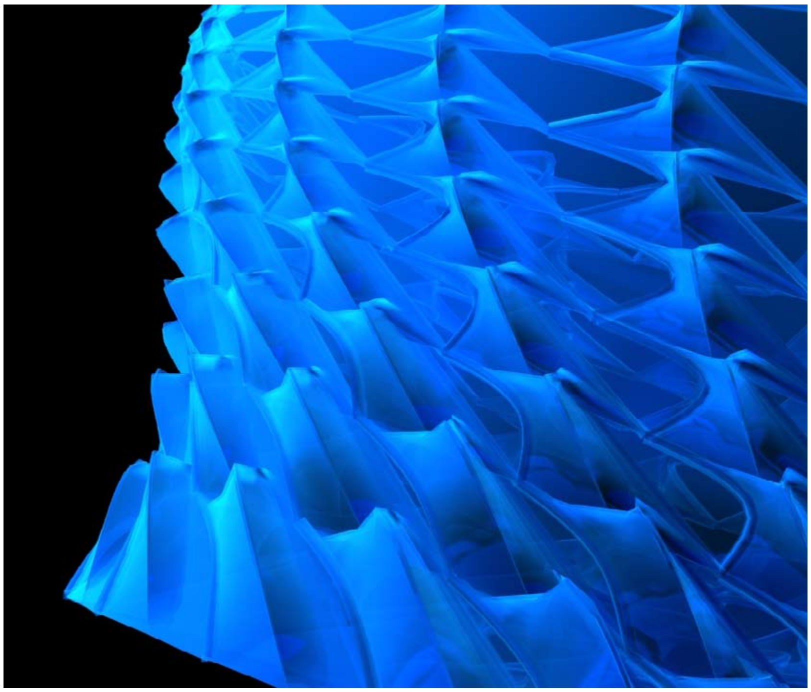
2005 GC workshop, Acadia conference Waterloo., Project by Judit Kimpian, Image: Judit Kimpian

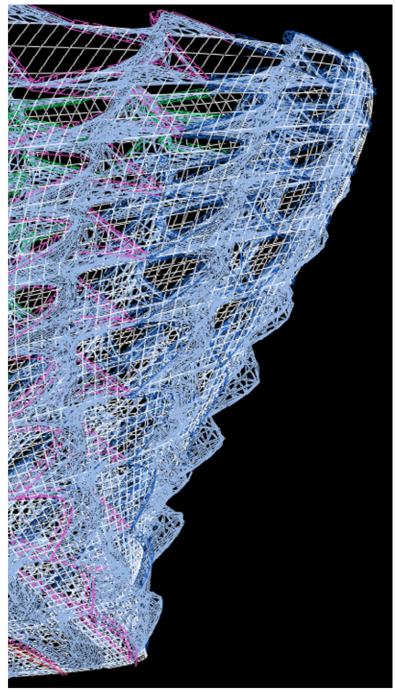


2005 GC workshop, Acadia conference Waterloo., Project by Mark Cichy, Image: Mark Cichy







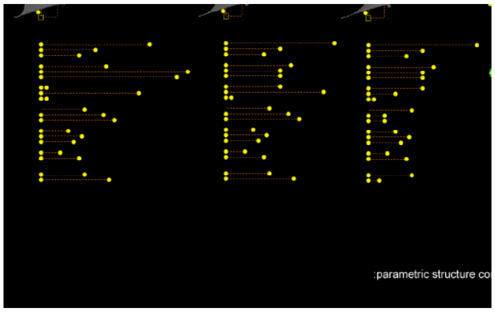


Design workshop at UFM, Guatemala City, Guatemala, summer 2005, Instructors Axel Kilian ,MIT and Axel Paredes, MIT and UFM.

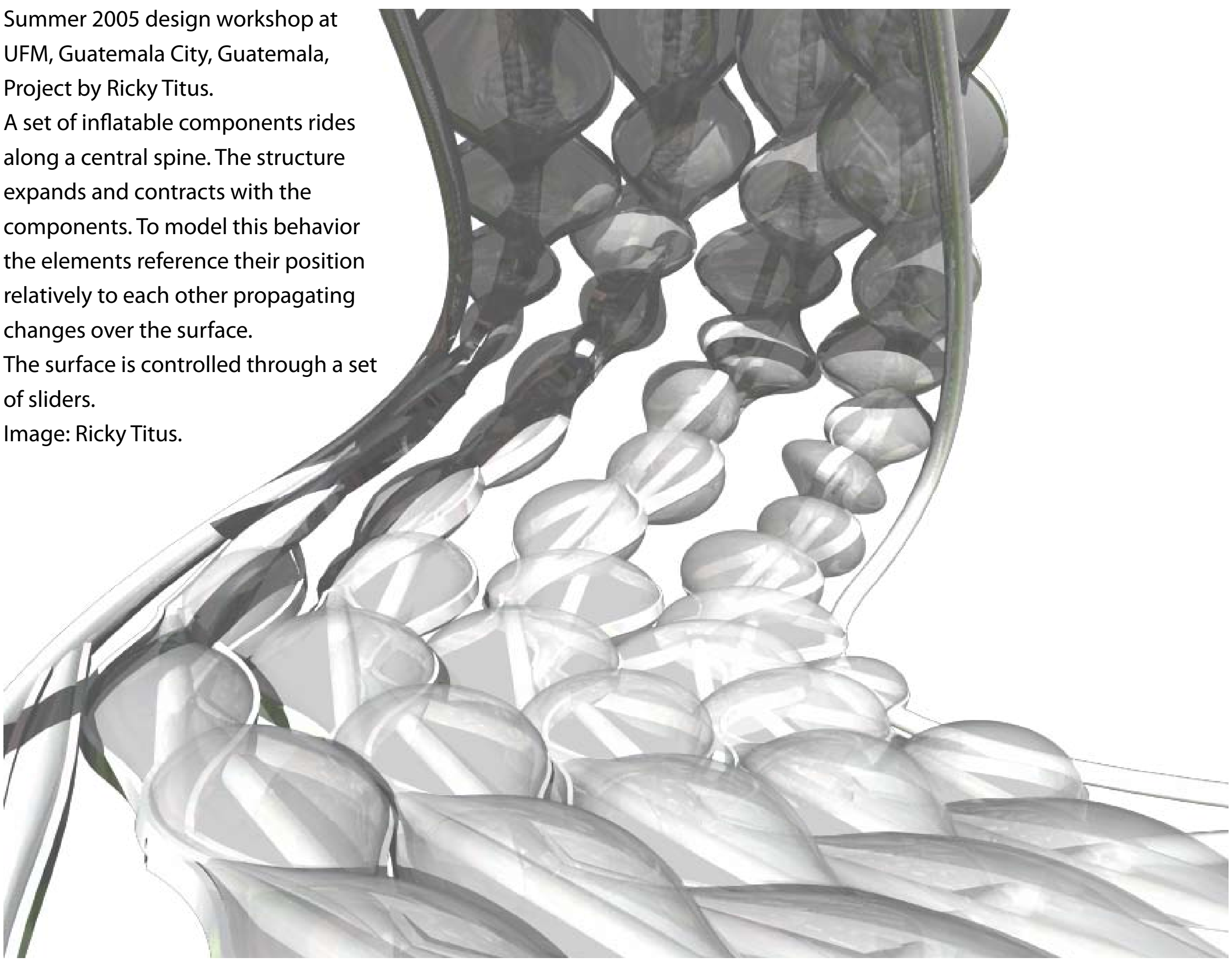
Four sites on campus were chosen and each student had to pick two for them and identify four points to have a structure touch down on it.

Parametric structures were developed that can respond to the varying site conditions both through geometric adaptation and through conditional changes. Paper based prototypes were developed from the design geometries to test the part compositions.

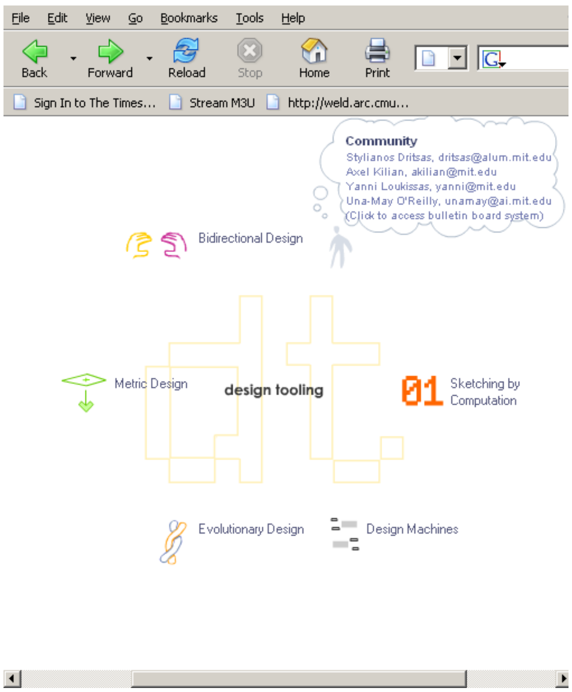
Summer 2005 design workshop at UFM, Guatemala City, Guatemala,. Project by Carlos Castaneda.
A set of components is modeled to ride on a design surface. The strategy was developed to have the roof respond to trees on the chosen site. components are prototyped in paper.
Image: Carlos Castaneda

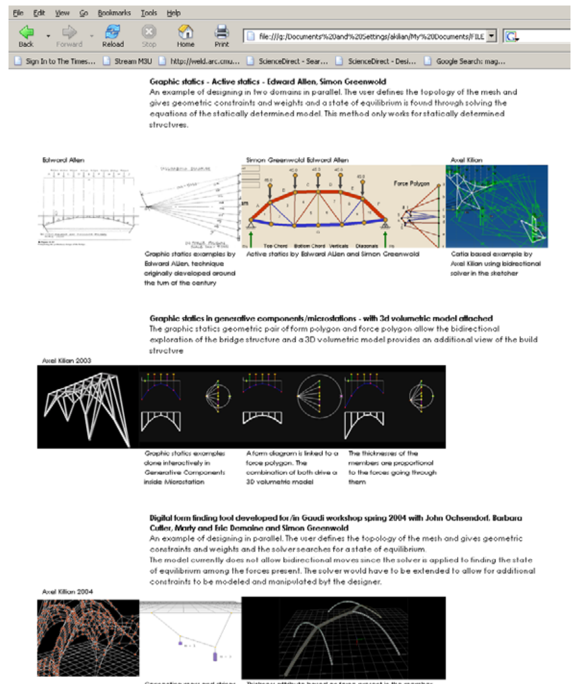


Summer 2005 design workshop at UFM, Guatemala City, Guatemala, Project by Ricky Titus.
A set of inflatable components rides along a central spine. The structure expands and contracts with the components. To model this behavior the elements reference their position relatively to each other propagating changes over the surface.
The surface is controlled through a set of sliders.
Image: Ricky Titus.



and generative extension to the Microstation platform, developed by Robert Aish, director of research at Bentley©. In addition, the workshops contribute to the development direction of the platform in giving user feedback and a platform for discussing the direction of the tool in the context of teaching and practice. The workshops reach a small audience of geometry specialists mostly in Europe and the United States. Eventually it is planned to launch





the platform as a commercial platform.

What makes the platform interesting from a teaching perspective is the parallel structure of the design representations available to work with. There is the geometric track for modeling the geometric dimensions, there is the associativity track that allows storing and editing the relationship between components and there is the design history that records the progress of the design and stores it as script. This script is also the basis for a programmatic expression of the design, which is the forth track. Despite its unfinished and un refined interface this conceptual approach works well in introducing students into the concepts of programming in the context of design.

The workshops have produced a large amount of exercise projects

2004 Design Tooling Initiative by Yanni Loukissas, Axel Kilian and Stelios Dritsas. Project advisor: Una-May O'Reilly.
The study developed and gathered existing approaches to the creation of user based tools in design and created a web site as a forum for people to access and interact with the information. Several larger themes were identified as starting points for learning how to program for design.
Image: Stelios Dritsas
http://destech.mit.edu/akilian/designtooling/inetpub/wwwroot/index.html

and only a few will be shown for demonstrating the general nature of the two day exercises. There is little time to develop full projects in the framework of the workshops.

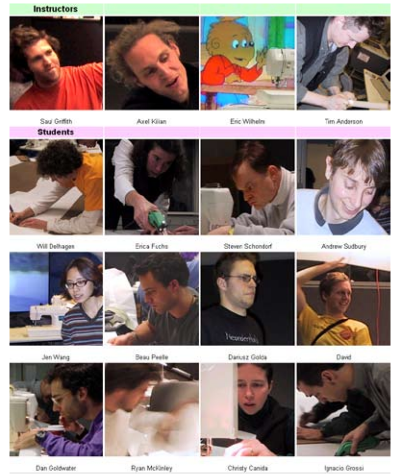
5.1.7.4 Design Tooling Initiative

In the summer of 2003, the author, Yanni Loukissas and Stelios Dritsas, advised by Una-May O'Reilly, developed a website to gather programming and digital design related resources. The name of the initiative refers to the perception of programming as a way to extend the existing methods and tools of designers in the digital context. A similar initiative exists on a much larger scale with the processing.org community started by Ben Fry and Casey Reas from the MIT media Lab that developed the processing platform to support programming based design and create an integration platform for a design community to share and exchange design

2003 IAP MIT Kite design workshop
Co instructors: Saul Griffith, Axel Kilian, Eric Wilhelm, Tim Anderson.
Surface patterns of the pneumatic leading edge tube before sewing. There are about 70 parts in a small kite and all of them are critical within a tolerance of a few millimeters for the performance and finish of the kite membrane.
Organization of parts and assembly sequences are a crucial aspect of the project. Overall structures of large kites reach lengths of 50 feet and surface areas of about 260 square feet.

Images: Saul Griffith, Axel Kilian, Eric Wilhelm projects and approaches. Digitaltooling only developed the beginning of a community during the three month project but it still serves as a resource to start students off with scripting and some of the concepts present in digital computing and design. The site is split into four themes, which reflect the research interests of the participants. The themes are bidirectional design, by the author, and evolutionary design by all participants. The designtooling

initiative addresses an emergent trend in both in the academic and the professional field of architecture of customizing digital tools through programming as well as programming design specific applications. This is not a new occurrence but it is only now that the distinction between designers and toolmakers is slowly becoming more blurry. It could have larger implications for how design is supported in the digital environment

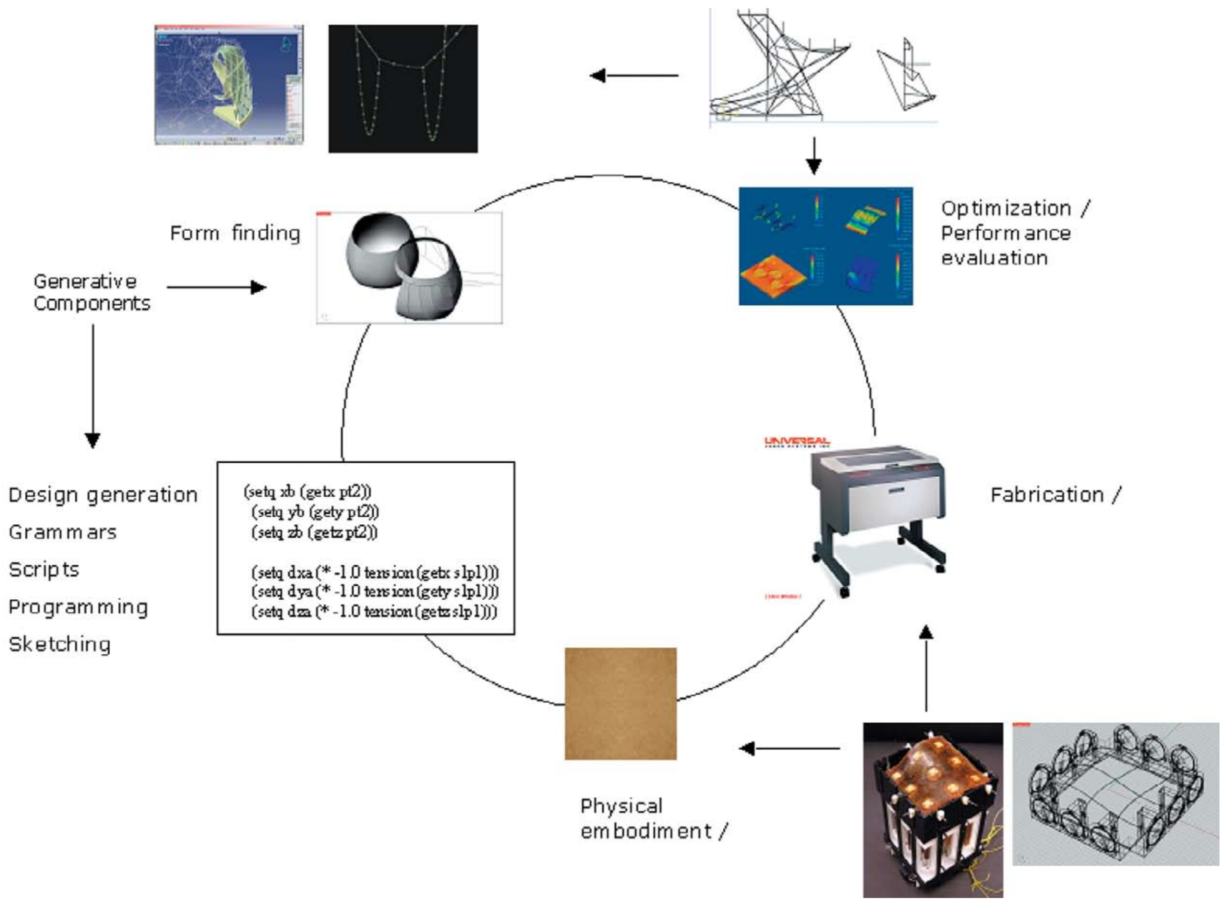


2003 IAP MIT Kite design workshop
Co instructors: Saul Griffith, Axel Kilian, Eric Wilhelm, Tim Anderson.
The workshop focused on the fabrication of full scale power kites directly from surfplan, a parametric kite design system by David Aberdeen.
Images: Saul Griffith, Axel Kilian, Eric Wilhelm

2003 IAP MIT Kite design workshop
Co instructors: Saul Griffith, Axel Kilian, Eric Wilhelm, Tim Anderson.
The kite design were cut from the digital geometry and unfolded onto paper as cutting templates.
The assembly methods had to be explained in detail as they are not integrated in the digital process.
Final full scale test by the author on a frozen lake. On the left, comparison between digital design geometry (bottom) and actual built geometry in flight (top).
Images: Saul Griffith, Axel Kilian, Eric Wilhelm

5.1.7.5 Kite building workshop IAP 2003

This independent activities period workshop at MIT focused on teaching surface based kite design using surfplan©, a parametric, fabrication-oriented and structural form finding software developed by David Aberdeen. Co-instructors were Saul Griffith,



Circular dependencies in the generative process between form generation, performance evaluations, materials, and fabrication.

Axel Kilian, Eric Wilhelm and Tim Anderson.

The kite building workshop benefited in many ways form the experience developed during the architecture workshops centered on surface based fabrication. The geometries of the fabric canopy and that of a double curved façade are not too different before material based adjustments are made. Similar machines and processes are used and the scale of a kit can easily reach that of a small pavilion or a small shelter with length of over 30 feet and canopy areas of 200 square feet. Teaching proper assembly techniques and keeping track of parts proved to be the biggest challenge in the project with 10 participants building one kite each which consists of approximately 40 different fabric pieces each. Of interest was also the influence of different assembly

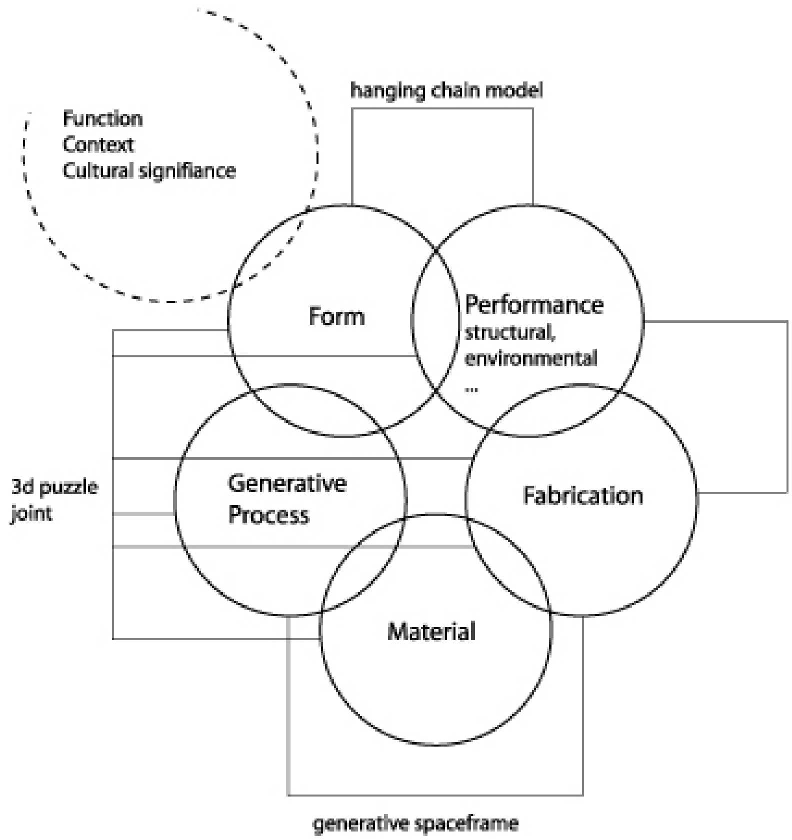


Chart sketching out design at the intersection between different constraints. These charts are preliminary studies leading up to the full experiments. The abstract modeling of the constraints did not prove to be useful. Later diagrams depict the exact dependencies rather than higher level abstractions.

accuracies on the resultant canopy shapes. Small errors can have disastrous results if they occur in crucial areas of the assembly as for instance in the pneumatically inflated airframe which has the smallest tolerances. The workshop also demonstrated the strength of a parametric design program with integrated fabrication output as it is the case in surfplan that was sued for the calculation and pattern creation for the workshop kite design.

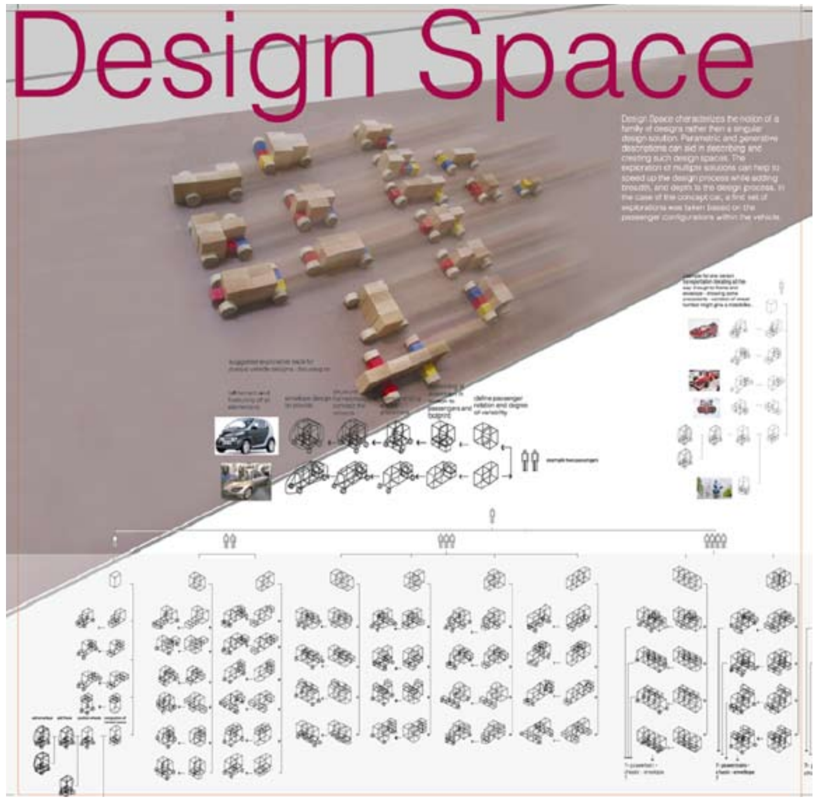
5.1.8 Conclusion – Circular Constraints, Material and Surface

Coordinating material properties and design geometry has vast unused potential for architecture. These opportunities have not been used in part due to the lack of digital material simulations suitable for design, but also due to the lack of the integration of such methods into the design process.

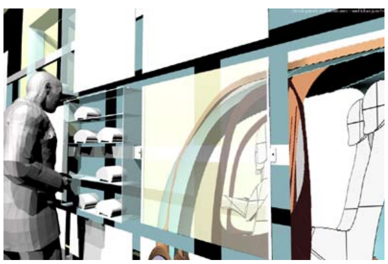
The experiments address this lack of suitable simulations with varying success. The creation of realistic material simulations is only of limited use as the simulation is useful for evaluation but difficult to use as a design driver. The simulation of a behavior needs to be translated into a generative design principle in order to be useful as a design driver. For instance, for single curvature there are geometric primitives such as cones that allow the creation of material conform geometries.

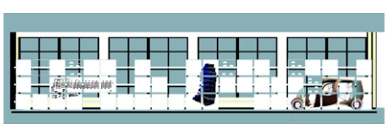
Another aspect that remains largely unexplored are the cross dependencies between parts of an assembly. The puzzle joint gets

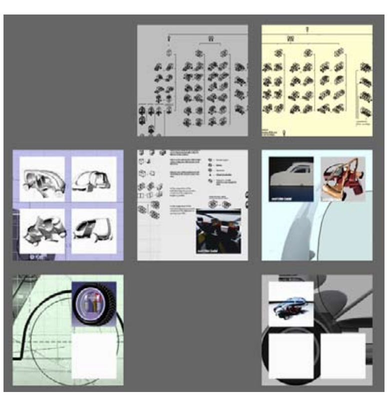
A poster design by the author for the June 2004 exhibition of the concept car design workshops. It depicts a set of block diagram studies conducted for the set of concept car experiments.



The exhibition concept by the author for the 2004 concept car show at the Wolk gallery. Exhibition development with the Smart Cities Group.







close to such an aesthetic for surface materialization, but has too many problems related to structural robustness to be usable. But it hints at the changes possible in the industry if generative methods are combined with digital fabrication and innovative material use. Ideally this can happen on all scales of design, from the tactile to the overall structure.

A similar potential lies within the explorative power of programmatic and functional descriptions of design when linked with other constraints present in a design problem.

The examples discussed in the following section are outside the domain of architecture, but were approached from an architectural perspective of design and innovation. Program played a role in this approach. But most importantly, it was a design exploration that set up and defined a design problem through the process rather than refined one with a clear aesthetic target in mind. The design target emerged through the exploration.

5.2 Branching Exploration: Defining the Constraints

The second set of experiments centers around the definition of a design problem. The term branching refers to the expanding of a design space through the exploration of different constraint directions as branches of the evolving design problem. The experiments took place around a series of workshops at the media lab.

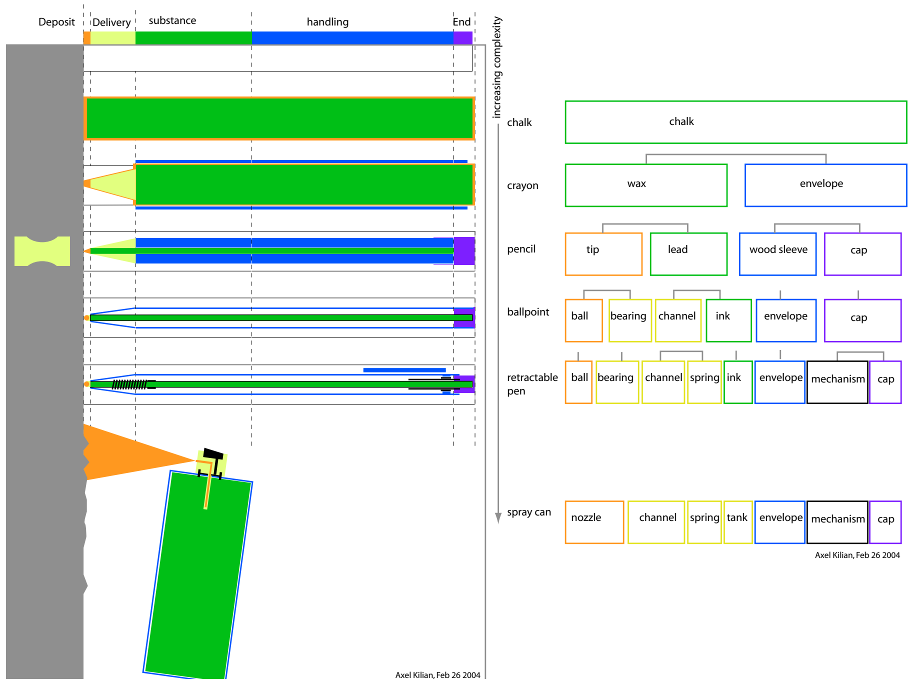
5.2.1 The Concept Car Workshops: Context

The car design discussed in this thesis was developed in the context of the concept car design studio headed by William J. Mitchell. The studio is a collaboration of the Smart Cities Group at the MIT Media Lab, General Motors represented by Wayne Cherry, as well as Jim Glymph and Frank Gehry of Gehry and Partners (in 2003/04). The car project was initiated by William J. Mitchell, Ryan Chin and Betty Lou McClanahan in 2003. The author and a group of core participants collaborated on the project from the beginning up to date. They were the graduate students in the Smart Cities Group: William Lark, Ryan Chin, Phil Liang, Patrik Künzler, Susanne Seitinger, Peter Schmitt, and Raul-David "Retro" Poblano. Ryan Chin from the Smart Cities Group coordinated the concept car workshops and the exchange with General Motors. The core group of students outside the lab was Mitchell Joachim, Franco Variani, and Marcel Botha. The author has been a collaborator and teaching assistant for the class since its inception together with many of the students mentioned above. The work shown in the experiment is the work of the author or work done in collaboration with the persons as credited.

5.2.2 Branching Design Exploration: Defining Solution Space

The branching exploration works through the linking of design representations. Step by step this process defines the design space of the emerging design task. In a set of preliminary experiments exploration experiments were tested aimed at expanding the

solution space.
Changing solution space can be achieved by regrouping existing function combinations. For instance, Patrik Künzler from the Smart Cities Group developed a wheel that integrates suspension and propulsion. This design move originated in redrawing the packaging boundary of the drive train of a conventional car to be



Writing device study project by the author for an exercise in the concept car workshop Spring 2004.
The analysis of the existing set of devices is translated into a function chain based on the core functions.

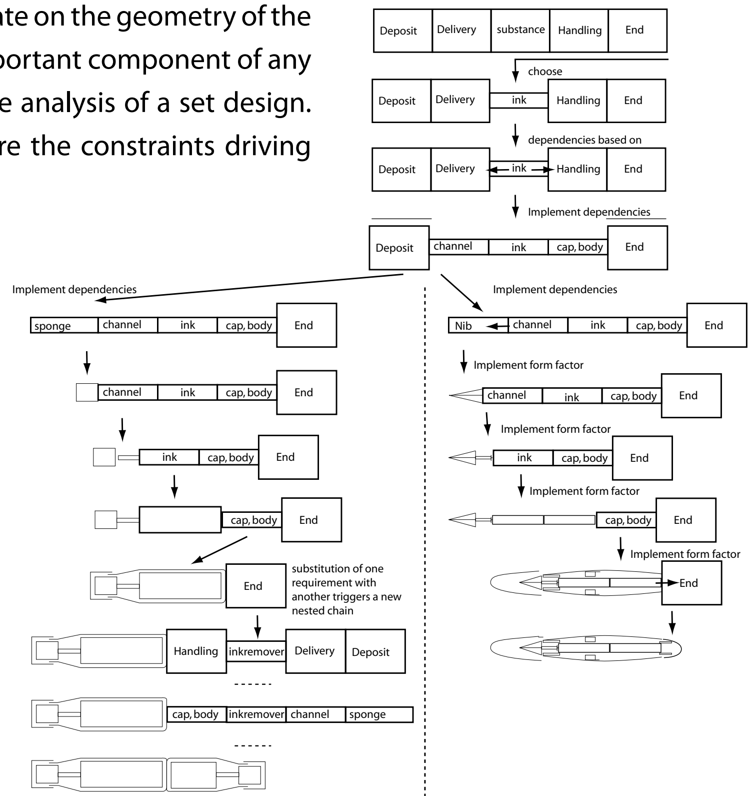
located in the space of the wheel.
This approach to innovation from functional groupings was further studied in a series of small exercises before applying this principle to the car design domain. The examples shown here were developed by the author.

- Writing devices.
- Laptop study – sketch only

5.2.2.1 Experiment 1: Writing Device – Functional Chains

An early assignment in the workshop was the analysis of existing writing devices as a starting point for novel designs. The study is discussed in chapter three but will be discussed here in more detail.
One approach to creating novel designs from a set of existing

designs would be a shape grammar approach. The extraction of rules from the existing designs in the set could be used to regenerate the existing set of designs as well as generate novel designs. This approach has been demonstrated many times in both architectural and product design. The problem of the approach in the context of this thesis is that the rules primarily operate on the geometry of the samples of the design set. Form is an important component of any design and can serve as the basis for the analysis of a set design. But particularly in product design, where the constraints driving



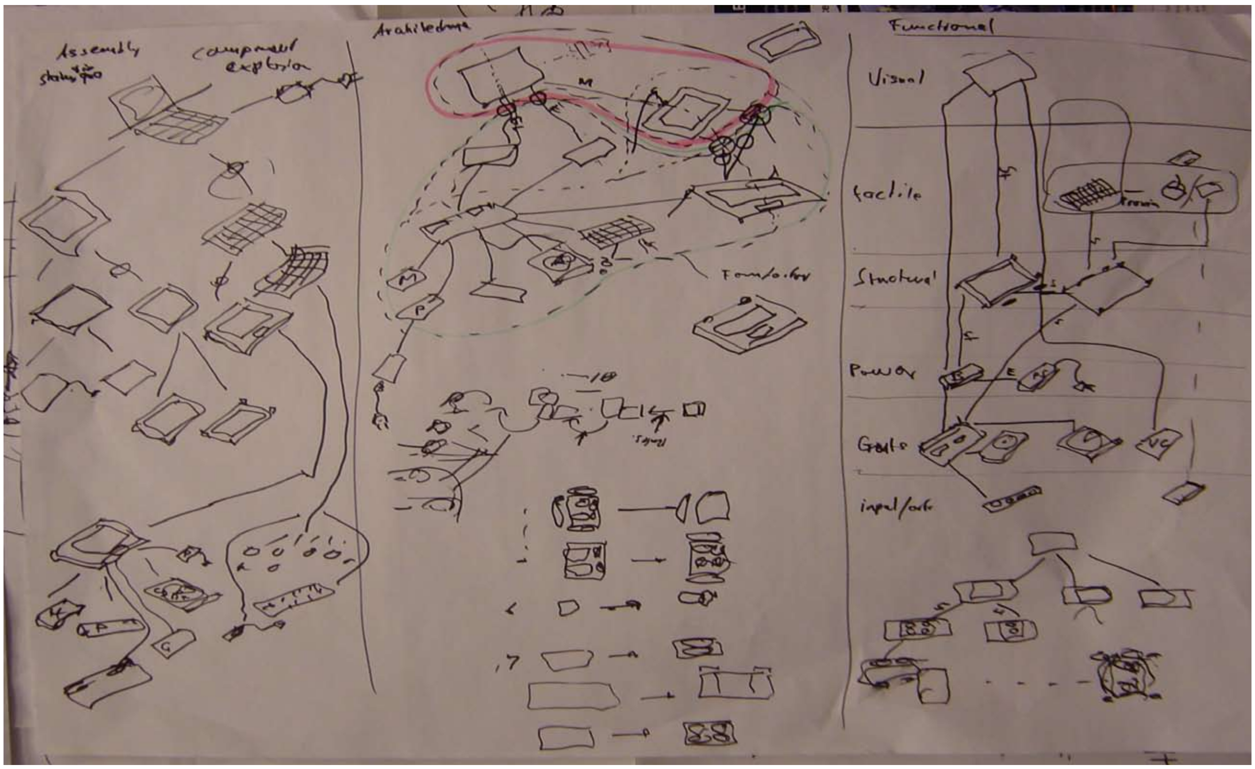
The function chain as a design generator for two different designs. The motivation for the *function chain* was the conceptual flexibility it affords in the progressive implementation of a concept.

a design are overlapping, the dependencies of the functional requirements are more important than geometric shape based compositions. At least the functional relationships play a more significant role in shaping form through physical components than in architecture.

The author would like to suggest an alternative approach referred to as *functional chain* approach. A *functional chain* can be described as the set of essential functional dependencies triggering a set of designs. It is a relatively loose functional description that triggers design choices.

Functional analysis

The analysis of the writing devices started by identifying the most common denominator amongst the set of devices studied: the process of leaving a mark on a writing surface. Sorted from the simple to the most complex, the devices studied ranged from a chalk to a retractable pen to a spraying can used for graffiti. At the core of every device there is a substance used to leave a mark with. This substance may be the device itself, as in the chalk, but it may also be the writing surface, as in the case of a stick being used to draw a line in the sand. In both cases though, the device is about



A further sketch based study of a similar analysis for a laptop computer. The importance difference to conventional, component based decompositions is that the function chain approach triggers novel ways to design for a similar function chain without confronting the designer with existing components to start from.
Sketch: Axel Kilian from a brainstorming session of the author with Franco Variani, Ryan Chin, and William Lark.

leaving a mark with a substance, even if the implementations are quite different. It is important to note that the examples differ in complexity but that difference is a function of implementation choices not of the basic functional requirements.

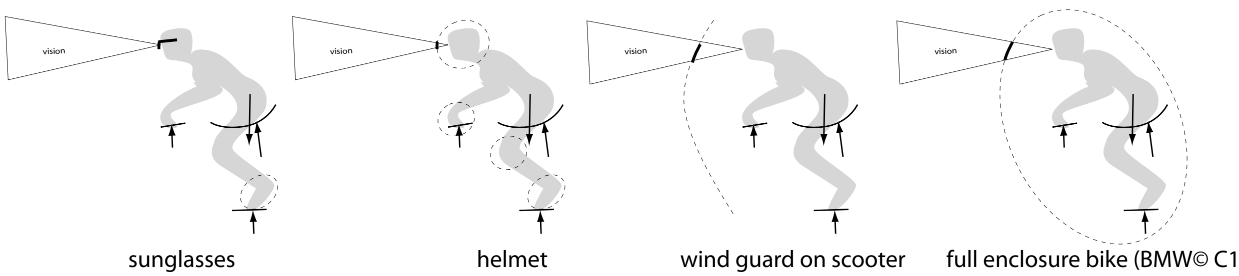
Other core functions implemented by all members of the set are that of handling the device to guide the marking. There is delivering the marking substance to where it is used, if the place of storage is not the same as the place of delivery. There is depositing the substance on the writing surface. This results in a reasonably detailed function chain, which in this case is more or less linear:

surface -deposit – delivery – marking – handling

There seems to be in some cases an additional function at the end tip like an eraser or a cap holder. This could be generalized into two additional functional requirements: removal and storage. The expanded chain would then look like this.

surface-deposit – delivery – marking – handling – removing – storing

It becomes clear that the complexity of the functional requirements quickly grows as the analysis moves forward. The study reveals that even a simple device has considerable complexity. In later



studies connected to the car, alternatively functional networks are introduced to address this issue, but complexity remains a challenge. This applies all the more to architecture of course.
The function chain resulting from the analysis now provides a starting point for addressing the functional requirements involved in creating a novel writing device. This process takes place before considerations of form, addressing the base architecture of the device. Only then will the base architecture drive the development of the geometry.

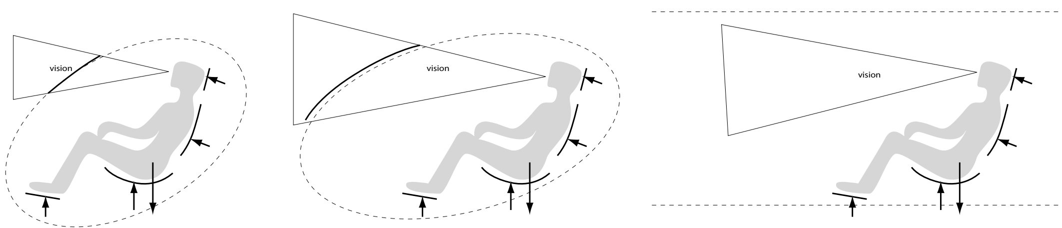
Relationship between design triggers and component implementations. The windshield can be viewed as a secondary design response brought about by the need to shield from the wind while driving and the need to see. Those two variable geometric entities create the demand for transparent wind protection.
With varying scale of the enclosure and position of the eyes to the enclosure different design solutions have emerged. To describe this set with a dimensional variable component would not satisfy all design variations.

Implementation

After creating the functional chain the first step in creating a novel design is the implementation of one of the functions. For instance a first step could be an implementation choice for marking, for instance a substance. The first implementation choice can be considered the design driver. A design driver is a constraint that is applied to a design exploration that is left unchanged and guides the exploration.

In the example, a standard substance like ink might be chosen. Substituting the functional requirement with an implementation choice triggers neighboring functions of the chain, in this case delivery and handling.

For the handling function, a simple choice is a tank. For delivery, a capillary tube might work. The next step, the deposit requirement, might be answered by branching into a variety of choices: a felt tip, a sponge pointing towards a felt tip pen, or a magic marker-like device. Or it might be left unimplemented by turning the design into a quill. We are left with the remaining links of the functional chain: removing and storing. Choosing an ink removing substance



Small car like Smart	Large car like a van	Inside a bus or train

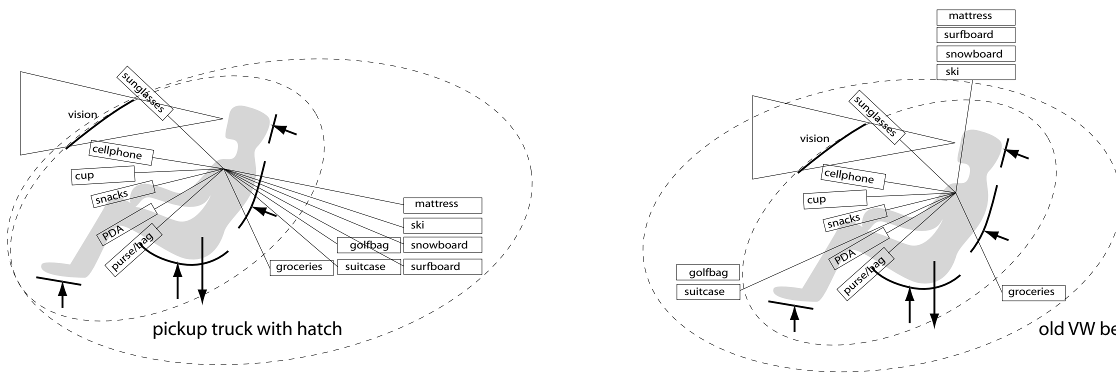
for removing function would lead to a symmetric implementation of the function chain with two back to back marking choices one for ink one for ink removal. The storage functional could be fulfilled with a simple protector cap.

Design choices

At this point, the established implementation chain needs design choices. Up to this point, form factors have not yet been considered.

Through the design choices, the implementation choices can be dimensioned. This happens according to the same process described for the implementation choices. The design choices ripple through the implementation chain. This ensures that dimensional dependencies are propagated correctly. Once established, the dimensions can be controlled parametrically to create dimensional variations. The chain sequence can be geometrically altered or packaged differently as long as the functional dependencies are fulfilled.

In conclusion, the function chain approach provides a robust, high-level approach to design generation from a set of existing designs with a focus on functional requirements as design drivers. Once established, the function chain drives the geometric design implementations.The principle of function chains is not fully externalized at this point. But function chains help to capture known design approaches to a problem in an implementation

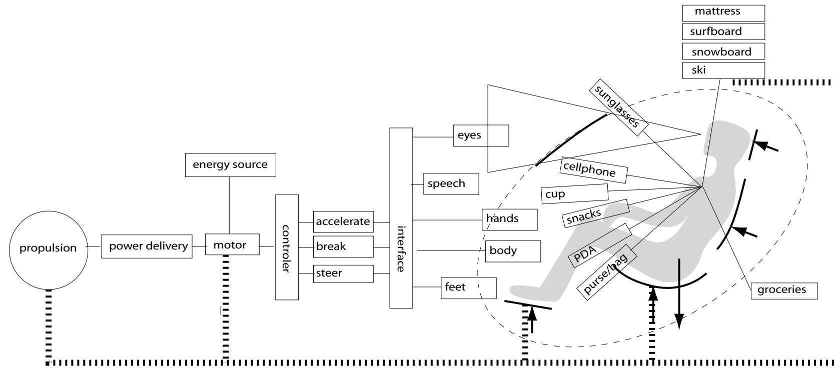


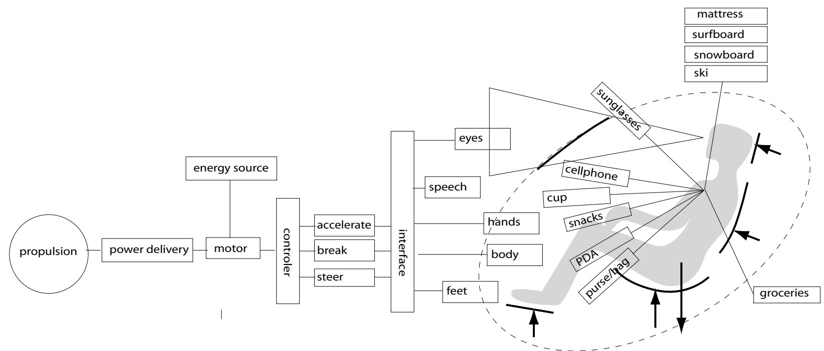
independent way. By unpacking and regrouping the functional descriptions innovative solutions can be generated.

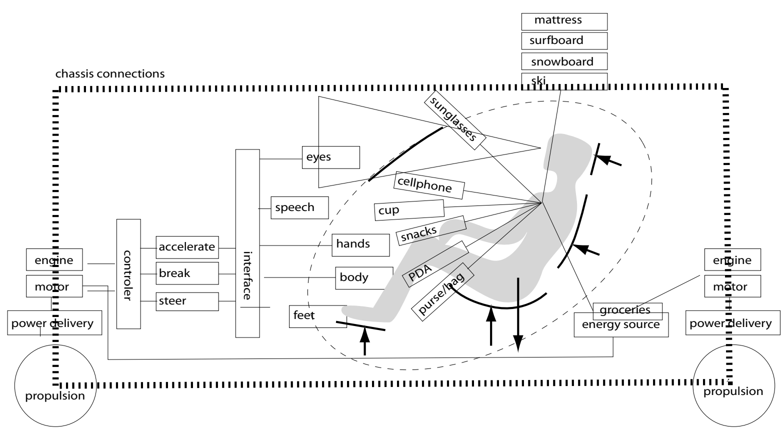
The development of a functional network was continued for cars. The grouping of driver centric functions and functional needs is the first step for a design response.

5.2.2.2 Experiment 2: Functional chains for the car design domain

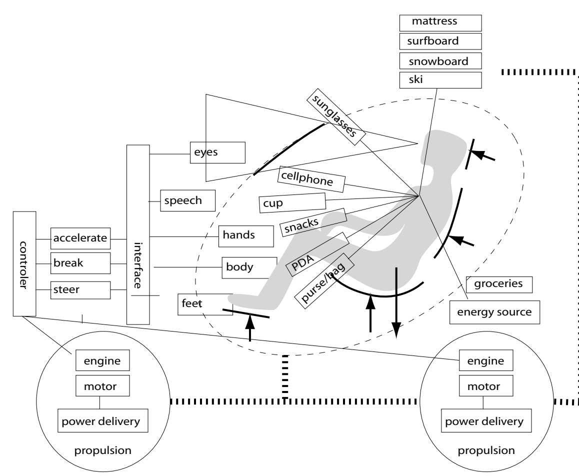
The approach of extracting function chains, or in the case of more complex dependencies, functional networks from existing designs proved very powerful. There are several ideas for design exploration that developed from it. First the reduction of existing design solutions down to their lowest possible functional denominator that captures the essence of the existing solution but goes beyond a component description. For instance in the simple example of the windshield for a car it is visible how the wrong kind of abstraction of the functional description will prevent design variations from emerging. The windshield as an object does not really exist in a functional description. It rather is a secondary reaction to the overlap of the primary functional requirements of vision of the driver and enclosure from the elements. The windshield occurs in most existing designs at the overlap of the vision cone demand







Different configurations of the same functional networks to serve as starting points to formulate functional demands and implementations.

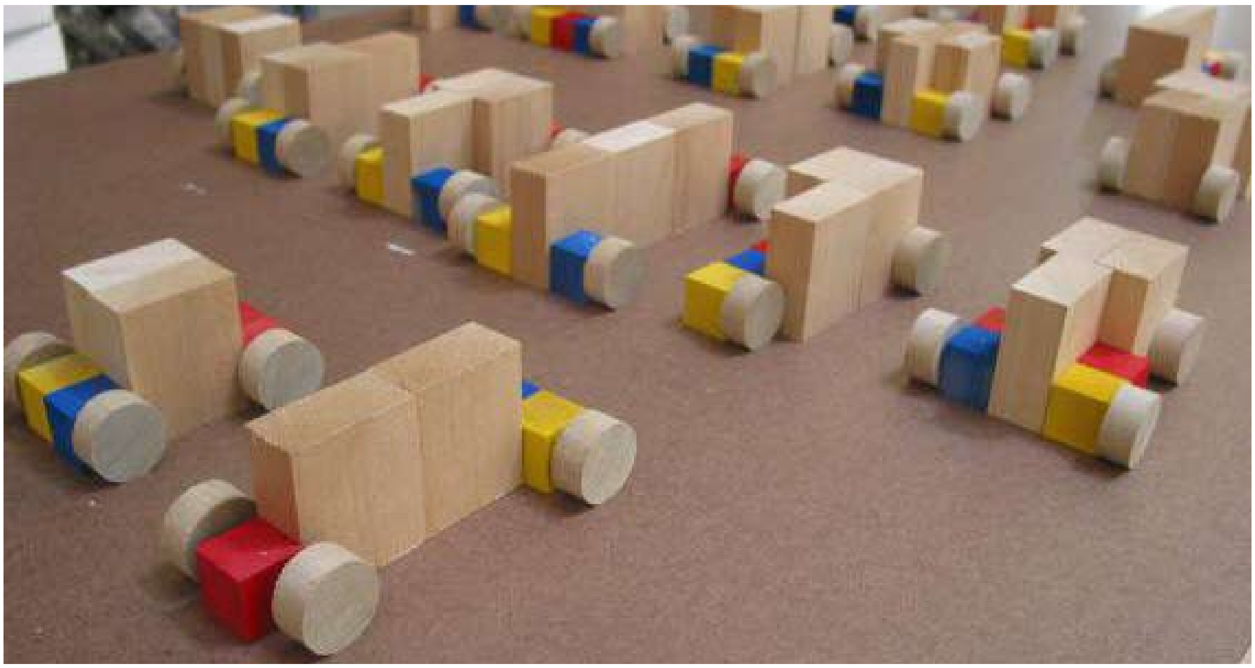


and the non transparent envelope enclosure. The emergent function requirement trigger a new secondary requirement of a transparent, wind resistant panel that conforms to the vision cone and replaces the envelope with a structural transparent material. In a linear interpolation of enclosures from a minimal solution of sunglasses over motorbike helmets, motor scooter shields to cars and eventually fully enclosed spaces like trains or busses the object windshield does not persist. Rather the design requirement from the overlaps triggers different design solutions from a pair of glasses to the disappearance of the object all together. A functional description with the node "windshield" would fail to capture this variety or worse, prevent an exploration of different solution by enforcing a design convention of existing solutions.

The functional graphs of the door studies follow a similar pattern. The drive train and passenger ones show how the increased complexity creates a bigger challenge in reading and working with the method. Here it is very well understandable how the graphic bundling of existing functions into new functional units, such as Patrik Künzler's initial robotic wheel design, can be triggered from diagramming the function graph.

5.2.2.3 Experiment 3: Exploration driven by functional constraints

The preliminary experiments leading up to the car experiments were testing how to set up an exploration for defining a design problem. The experiments helped to establish the notion of the branching exploration type. The branching exploration would



Block study of car configurations built by Gaston Nogues in collaboration with the author. at Frank Gehry and Partners..

The work area dedicated to the car during the visit of the author in the office in November 2003.
Images: Axel Kilian

identify the functional constraints and possible design drivers for defining the design problem. In case of the car it was the type of program and usage scenario and the functions a vehicle would have to fulfill. The first set of experiments focused on product design examples to test less complex entities first.

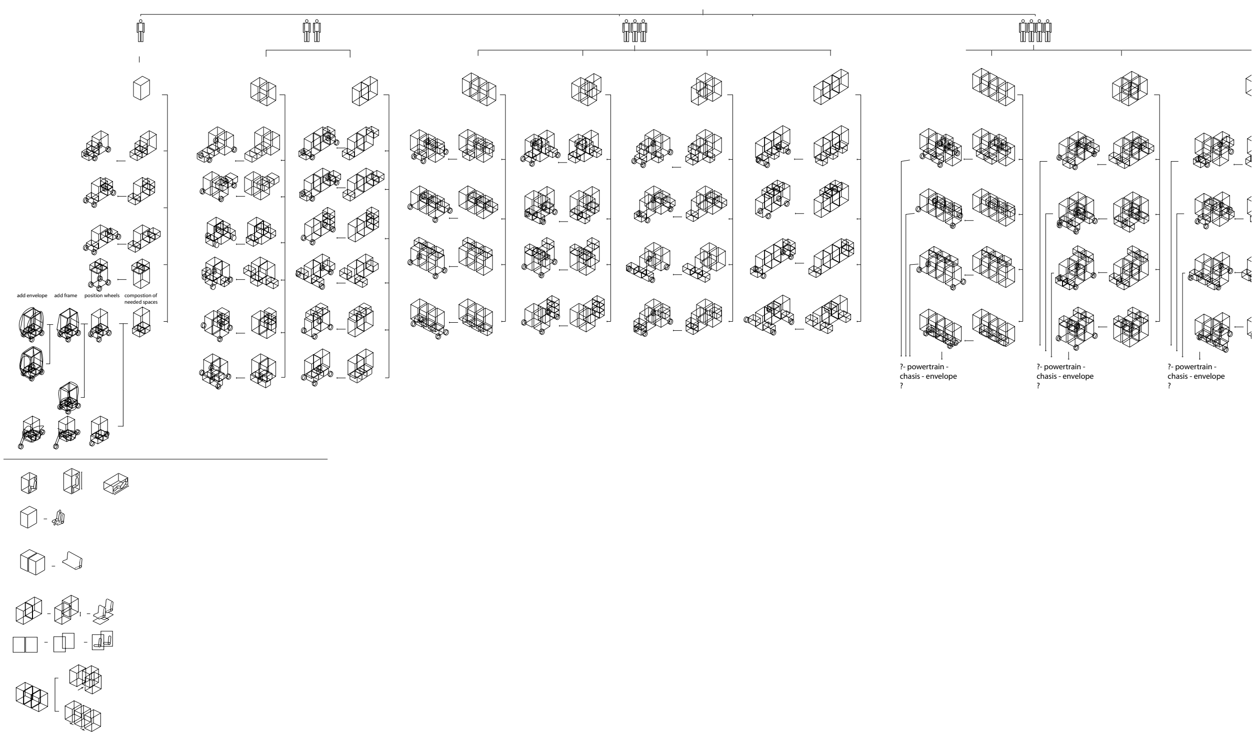
The motivation in working with functional descriptions was to overcome the component-based decomposition of products, typically used as a starting point for new design generations. In contrast, a functional analysis implies careful choices of abstractions in order to trigger possible innovation. This is different from the component-based approach leading to incremental variations in the design.

The preliminary design experiments focused on a number of exploration techniques to trigger the rethinking of existing design solutions.

- spatial functional recombinations - seating arrangement studies
- ingress/egress studies – door studies
- chassis integration - frame orientation

5.2.2.4 Experiment 4: Block massing studies.

Prior to the detailed design studies, precedent research was conducted within car design, focusing on the potential of key functional and architectural components for pushing the development of the architecture of the car. The car project was originally set up in collaboration with Frank O. Gehry with the intention to introduce aspects of the architectural design process into the car design process. The first attempt was to test massing studies similar to block diagrams in architectural sketch design for the programming of a car. Since drivers and passengers constitute a major factor in the layout of cars, the person envelope was chosen as the starting point. The physical building block models proved useful for quick physical interaction with the design variations but had limited flexibility of the very tight programmatic overlay of car design. The digital version proved to be more flexible. The massing wood model was built by Gaston Nogues from Gehry and

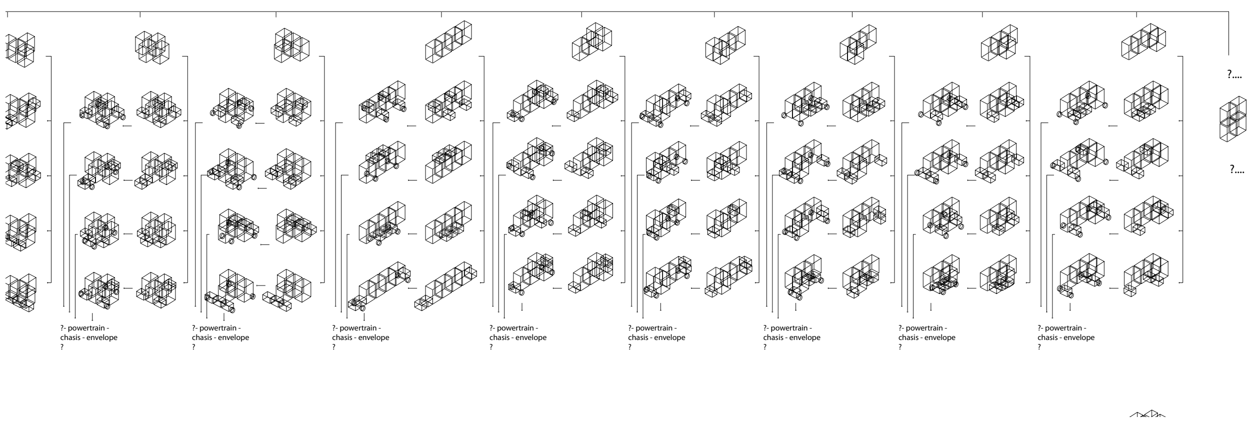


Combinatoric tree study putting the passenger configuration at the center of the design exploration.
Tree study by the author from a brain storming session together with Ryan Chin and Franco Variani.

Partners in collaboration with the author. The initial block study was developed by the author, Ryan Chin and Franco Variani.

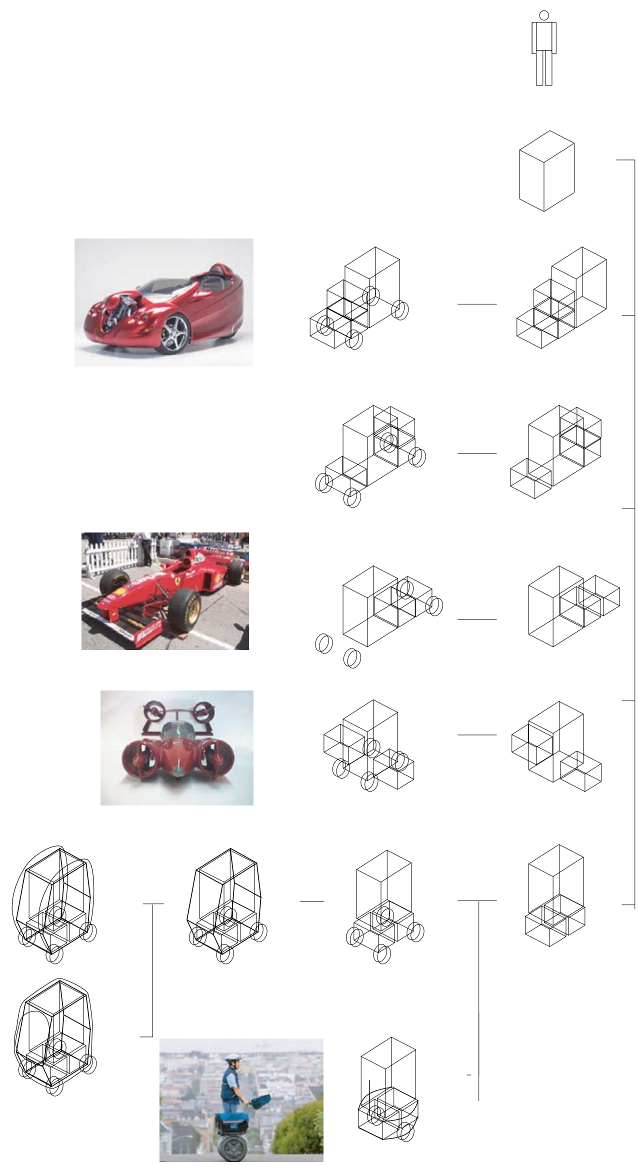
5.2.2.5 Experiment 5: Seating arrangement as a driver for car architecture

In parallel with the physical block studies, digital combinatorial studies were done by the author. The explorations were driven by



an interest in grammatical explorations of car designs using rule-based methods related to shape grammars. To simplify the exercise, the space required for an individual person was initially abstracted as a simple box.

One outcome was a tree of car configurations driven by the social configuration of the driver and passengers within the vehicle. The majority of current cars are situated in the square, four passenger seating arrangement. The combinatorial approached proved to be useful to quickly iterate through a large number of variations of car architectures without investing into design details and styling up front. A parametric model was built in CATIA® that allowed the reconfiguration of the vehicle based on the positioning of the passengers. This was a powerful proof of concept, but far too



A section of the tree shows the different combinations of a single person represented by a volume of occupancy box with the different combinations of wheel and power train. After these combinations were generated it was easy to find designs that already implement them. It was interesting to see that the range of implementations went well beyond a conventional car.

Two passenger iterations - showing how two existing designs can be derived from passenger combinations. All studies by the author

Examples of existing implementations SMART and JETCAR

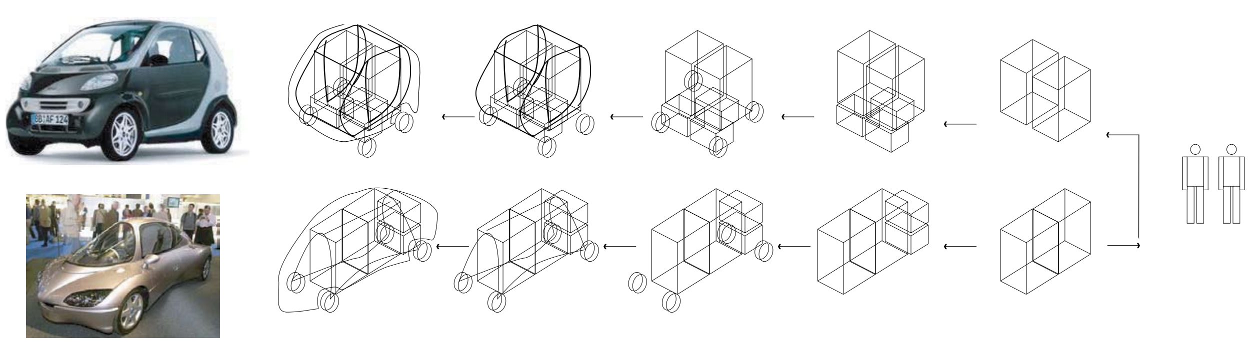
Envelope design

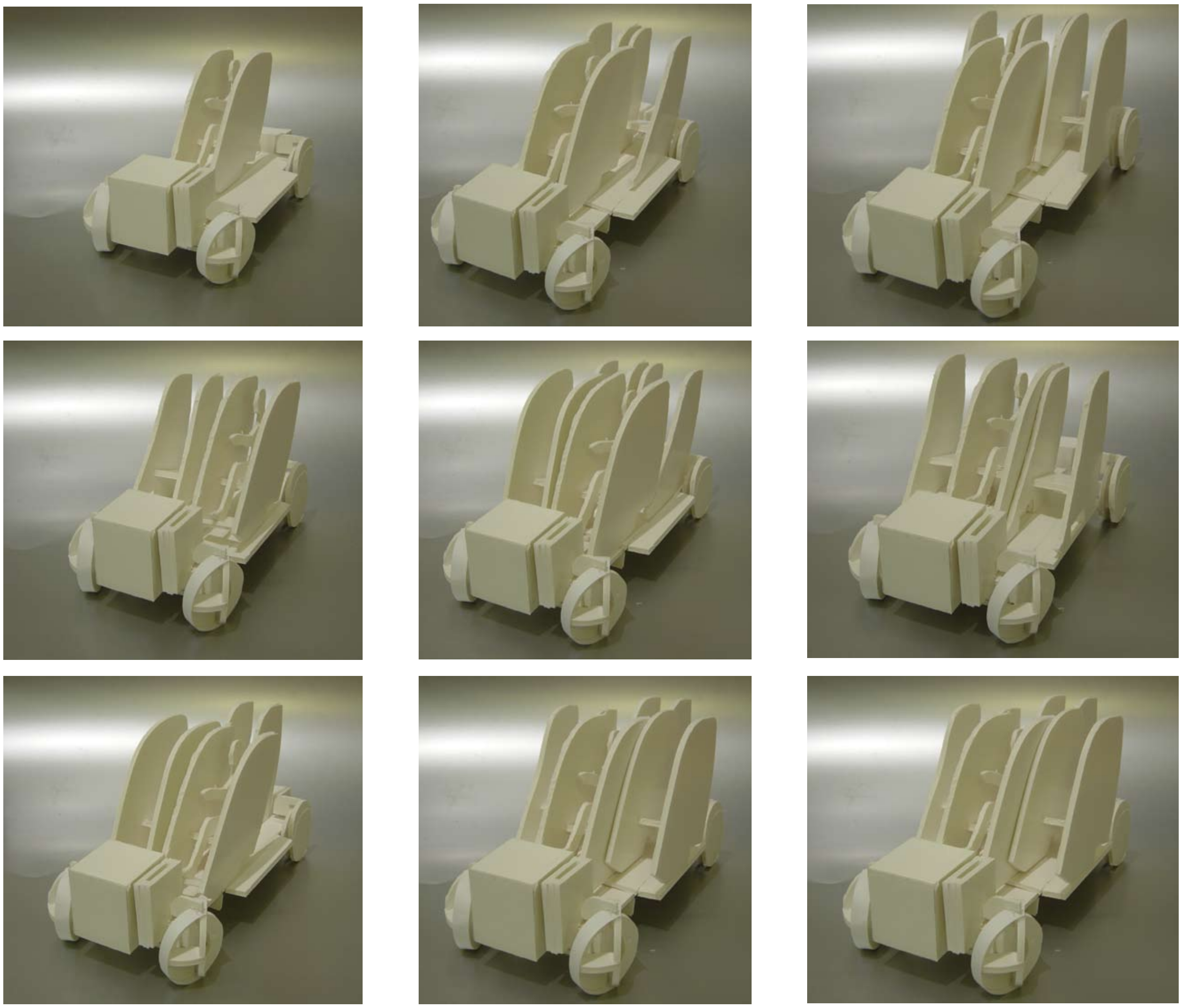
Structural frame/ chassis

Corresponding wheel placement

Power train

Passenger positions





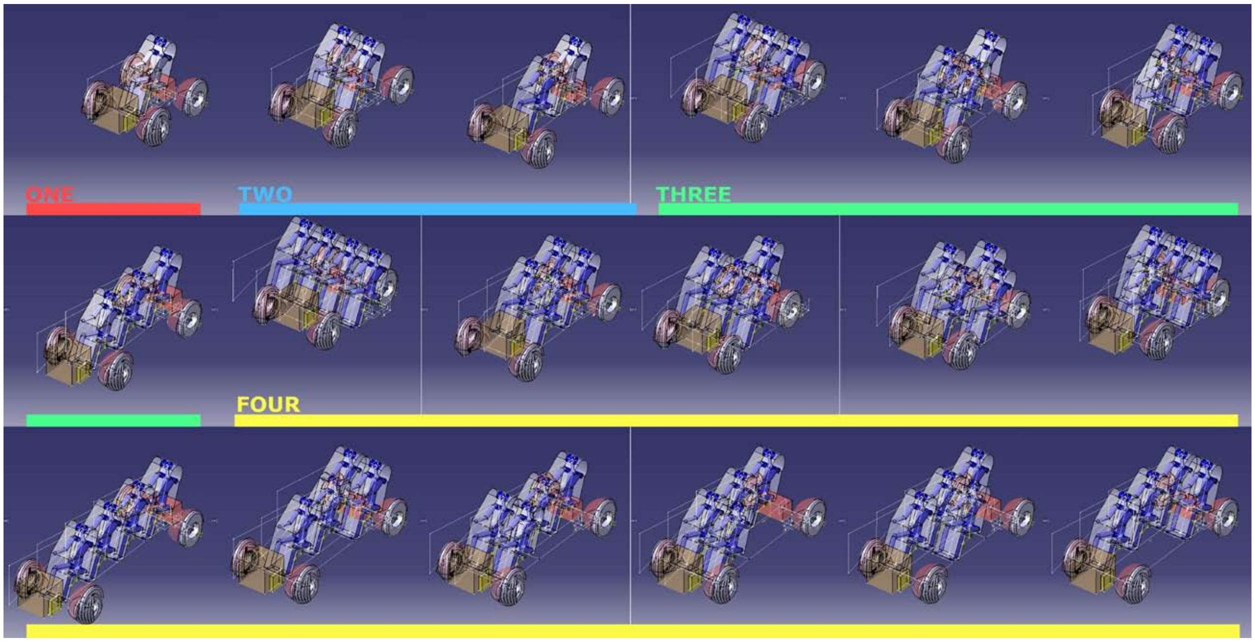
Foam core studies of different passenger wheel configurations. Both passengers and wheels are modeled as minimal occupancy volumes. The wheels take into account steering and suspension movement..
The structural platform is built to allow for extension of the chassis in both length and width to fit the different configurations.

limited as a design explorer due to the rigidity of the parametric model. The model lacked the necessary generality in the definition of the parametric relationships.

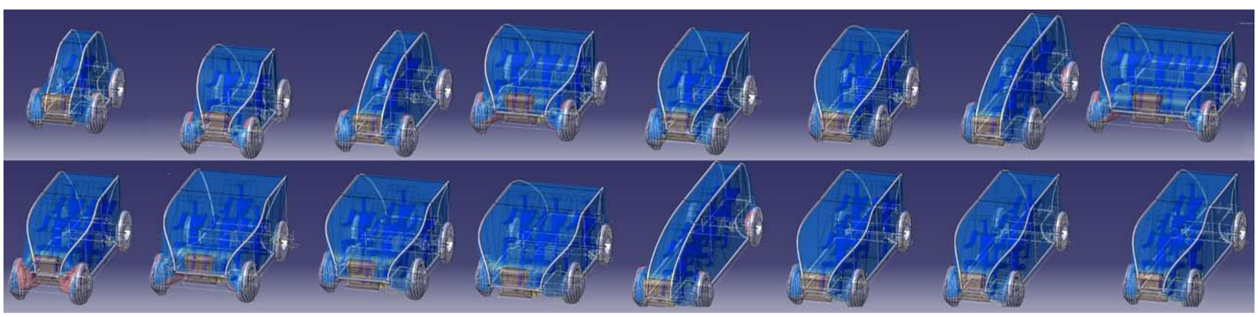
The initial experiences with grammar-based, explorative and relatively unfocused design approaches later led to more specific design tasks responding to groups of functions. Two of these designs developed within the concept car studio were the “City Car” and the “Athlete”.

5.2.2.6 Experiment 6: Block diagrams – seating charts

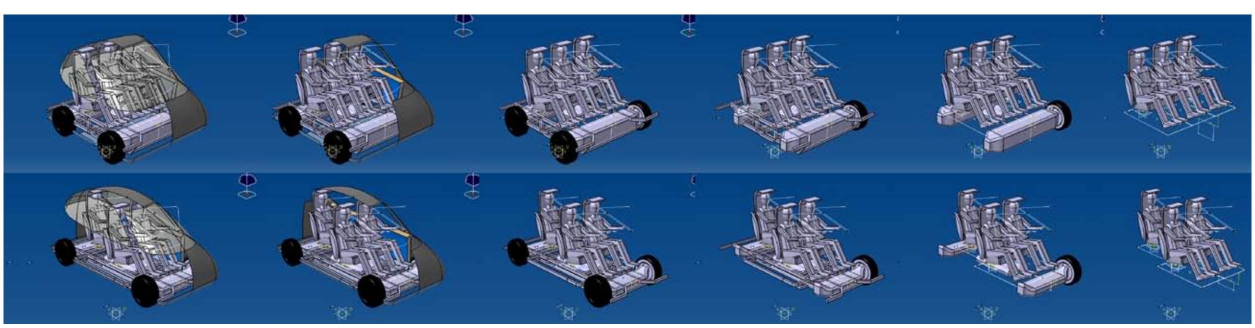
From the combinatorial arrangements of the seating components, a quick tree of possible seating arrangements was generated, allowing a fast overview of the possibilities. The possibilities are by



The first schematic parametric models allowing for the variations of the passengers. The graph shows the same model with passengers distributed differently.

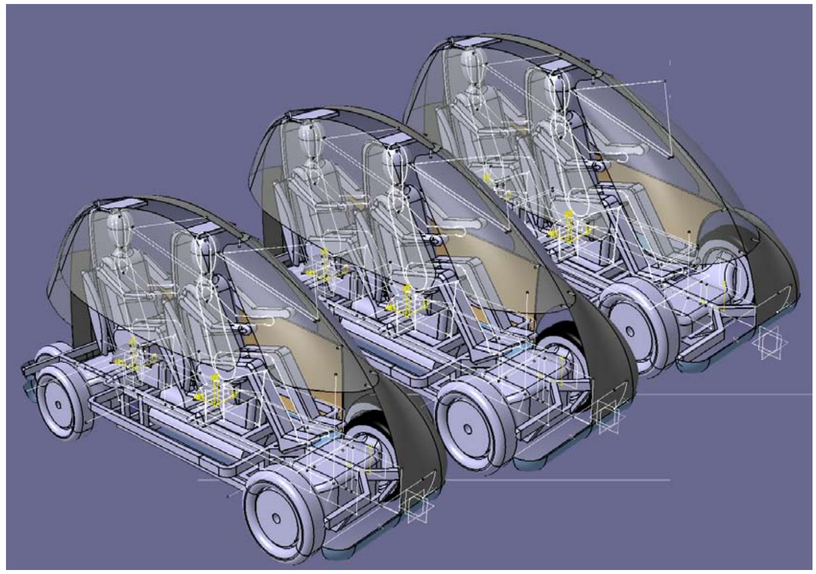


First studies of modeling frame and envelope details to respond to the passenger configurations.



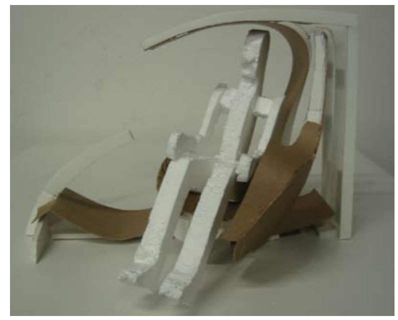
The latest model series with fully modeled surfaces and chassis frame structure and hydrogen storage tanks in the sandwich type floor.

Detail of the CATIA parametric study. The model allows for the reconfiguration of passengers all the way from one to four passenger in any front facing configuration. The car envelope is parametrically adapted to fit the new minimal volume to enclose the passenger. This also affects the storage hydrogen tanks in the sandwich floor. If the car width is sufficient to store them side by side the are placed in parallel, if for a one person version the car is too narrow, they are stacked in a pyramid. In addition all proportional parameters such as window lines or roof heights are editable.



Two variations of the same parametric model showing the adaptation of the chassis with the additional passenger. The biggest challenge in this parametric study was to model all geometry in a way that consistently functions in all cases and without variations in the continuity of curves and surfaces. This model was developed in the spring of 2004 by the author using CATIA V5R11 and a Z-corp three dimensional printer.

no means exhaustive, nor are they all feasible, but they provide a fast way to generate possible starting points for investigations. After the tree was generated, examples were quickly found for most



One of many ingress/egress studies developed by the author. This one is a selective prototype to challenge the notion of a car seat as a piece of furniture. Instead the seat bucket is attached to the door to aid in getting in and out of the vehicle, and the back support is attached to the vehicle roof. This split of the seat offers interesting alternatives to interior and exterior designs of vehicle. It was triggered by thinking in terms of functional chains of the functions of weight support, egress/ingress and safety from side impacts.

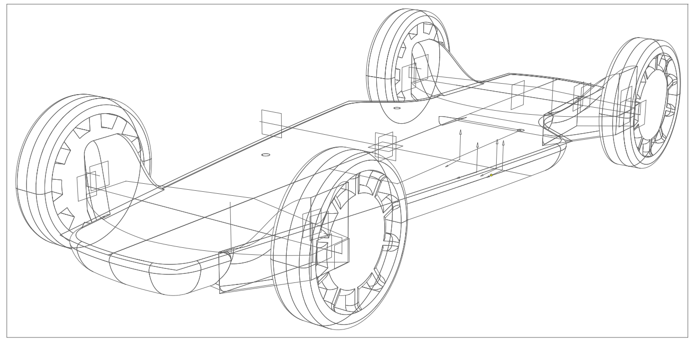
of the generated configurations. This proved the validity of the approach as way of sorting and triggering different combinations for further study.

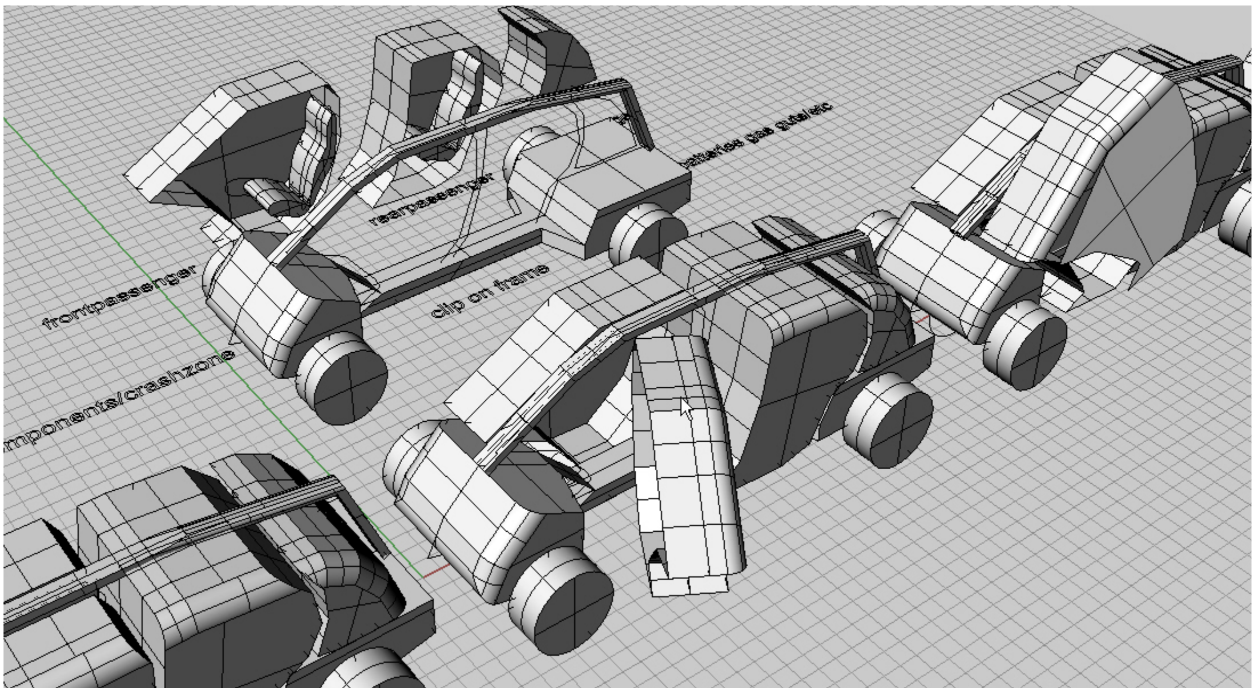
A fully parametric model was constructed for the tree combinations using a simultaneous representation of all four passengers. A so-called rubber band structure allowed for the use of a fixed topology while changing the number of passengers within the same parametric model.

This approach proved still too limiting for the explorative, idea generating approach. In order to be explorative the process could not just be limited to top down combinatorial approach.

An alternative emerged from several different projects in the class most notably from the first designs for a hub less wheel with

The autonomy concept by GM modeled by the author in CATIA. The platform integrates power train and energy storage and frees up the passenger compartment from those functions.
The author proposes an alternative by suggesting reintegrating the vehicle components vertically while still offering similar flexibility.



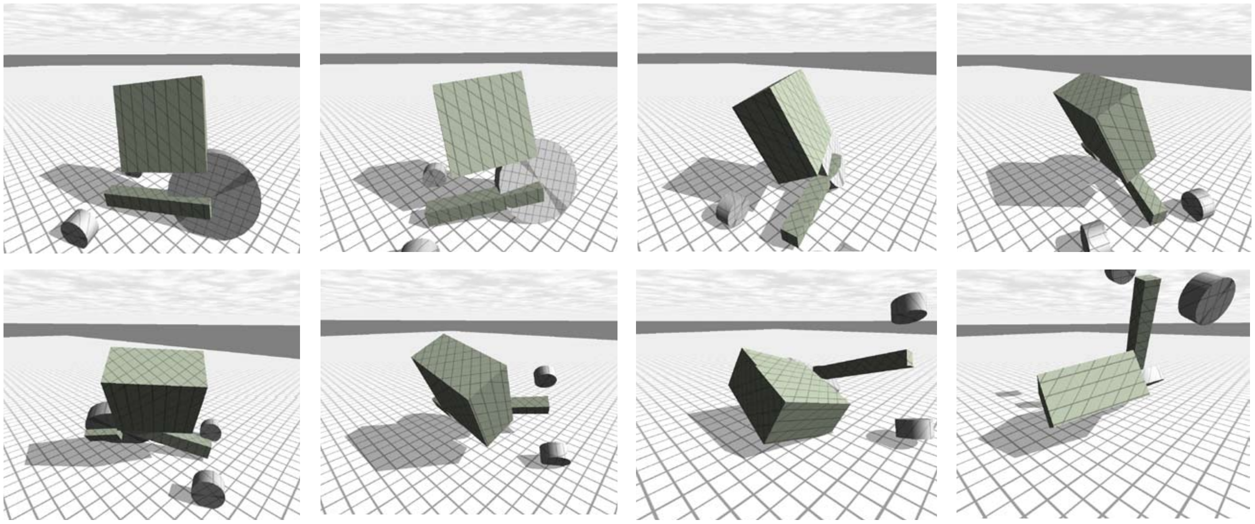


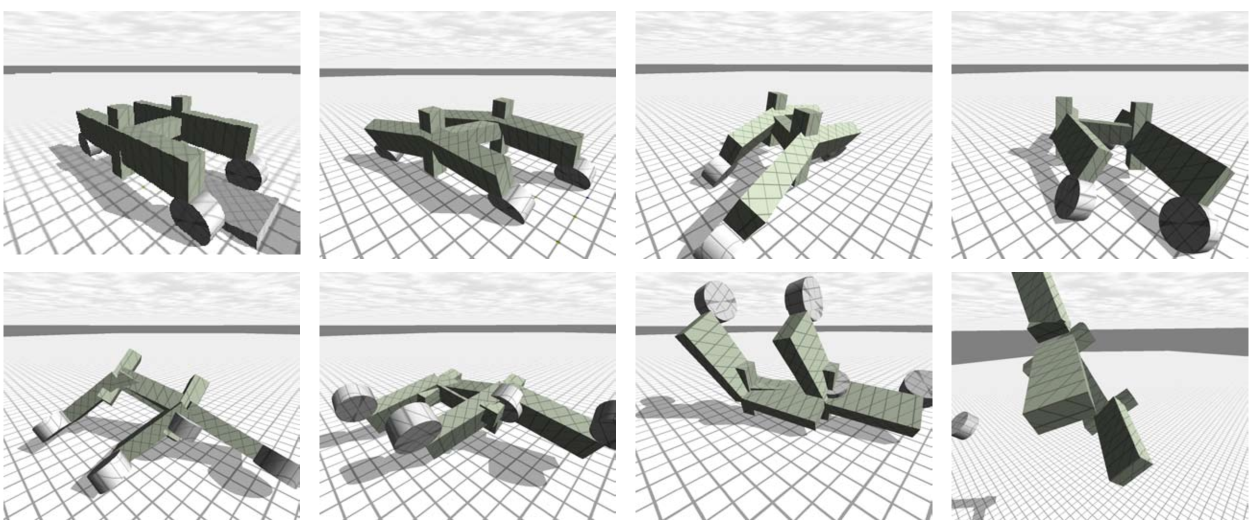
A design exploration by the author challenging the notion of horizontal frame structures for modular car designs such as the GM "Autonomy". skateboard like car platform. Instead it is proposed to stand up the structure vertically and attach seat door combinations as snap on components from the outside.

integrated motor and suspension by Patrik Künzler from the Smart Cities Group. The approach can be generalized as the regrouping of existing functions in to new functional components. The new groupings of functions define the requirements for any needed components. Packaging and spatial relations of the functions change as well as the overall function of the design solution.

5.2.2.7 Experiment 7: Door-seat relationship

In a series of studies for ingress and egress, functional regrouping led the author to rethink the role of seat as a functional entity similar

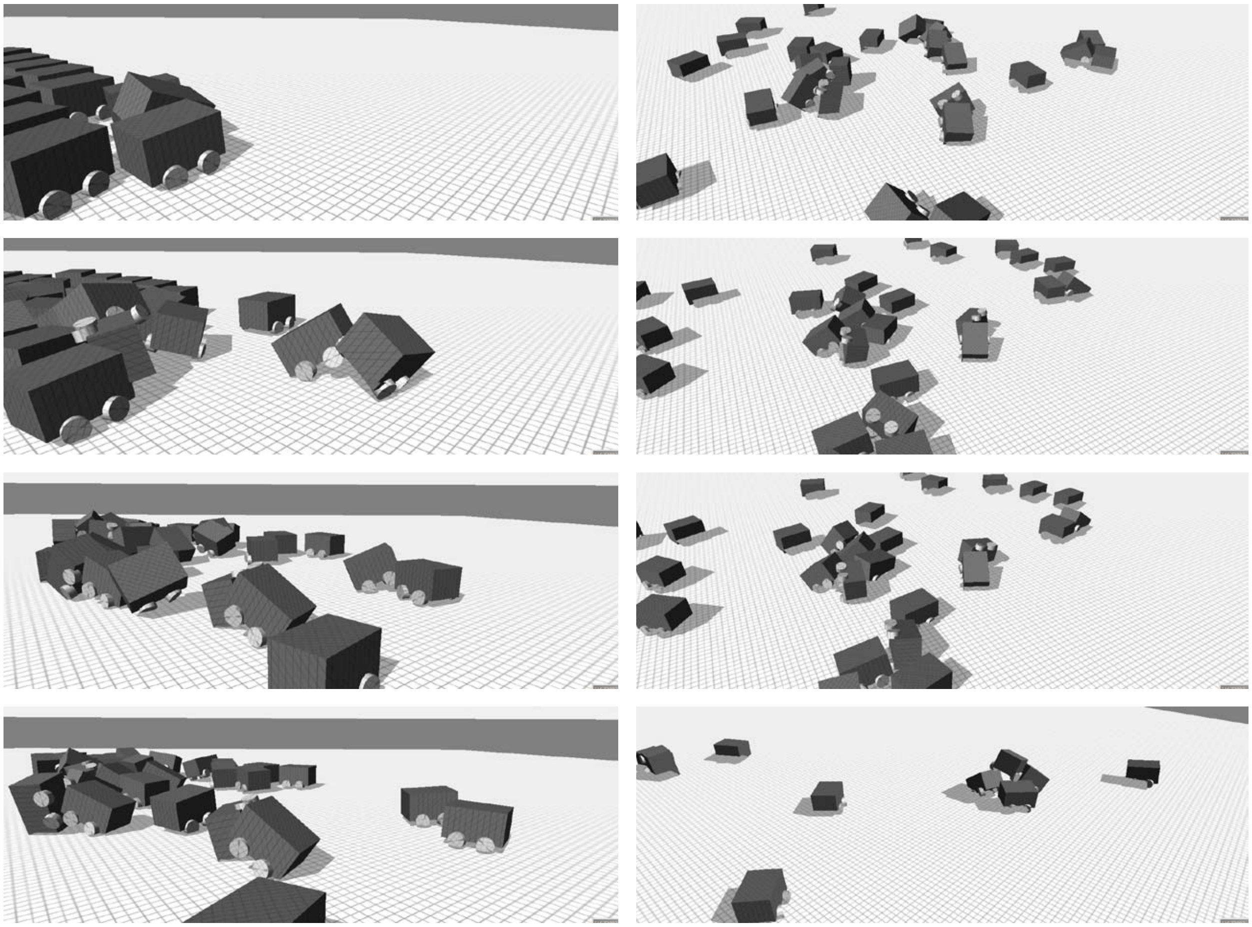




to furniture and challenge its design as an object. The split of back rest and seat bucket led to a new interior design. The seat bucket is attached to the door frames and swings out with it, aiding in egress, and the backrest stays connected above to the roof forming part of the safety cell.

Although this design has many disadvantages, it is successful in challenging the concept of car seats as furniture within a living room like interior. The design contrasts the conventional seat design with a more integral, composite design. Through its new sub-components, it responds to several other previously unrelated functional constraints as well.

A design evaluation of the performance of a tricycle design by the author. This is programmed in C++ in the open dynamics engine, an open source rigid body dynamics simulator. The approximate physics based behavior allows a quick assertion about stability of a vehicle sketch before prototyping. The first study is the one wheel athlete, discussed later in this section. The second study is the full sized athlete.



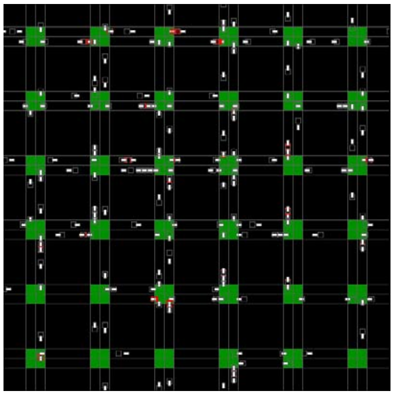
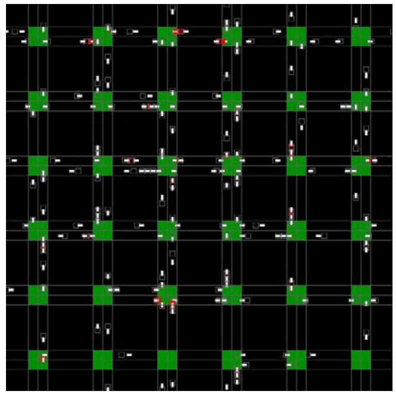
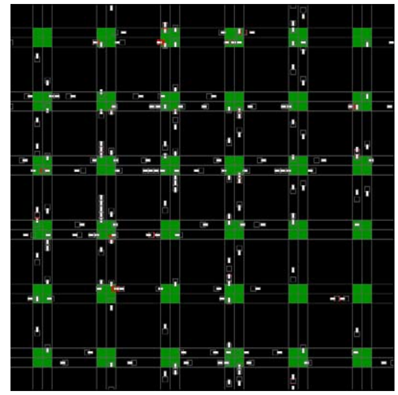
A vehicle interaction study based on the question: What happens if vehicle can bump up into each other? This study built upon the idea of "gentle congestion" by Mitchell Joachim imagining soft cars.
Design study by the author in C++ using the ODE environment.

5.2.2.8 Experiment 8: Switching the frame orientation from horizontal to vertical

Another such move was the switching of the chassis from its dominantly horizontal orientation to a vertical orientation that frees the floor for novel egress and ingress designs. The structural shift frees the floor of structural functions and allows for door designs that withdraw the floor in order to create comfortable ingress and egress solutions for instance for the elderly.

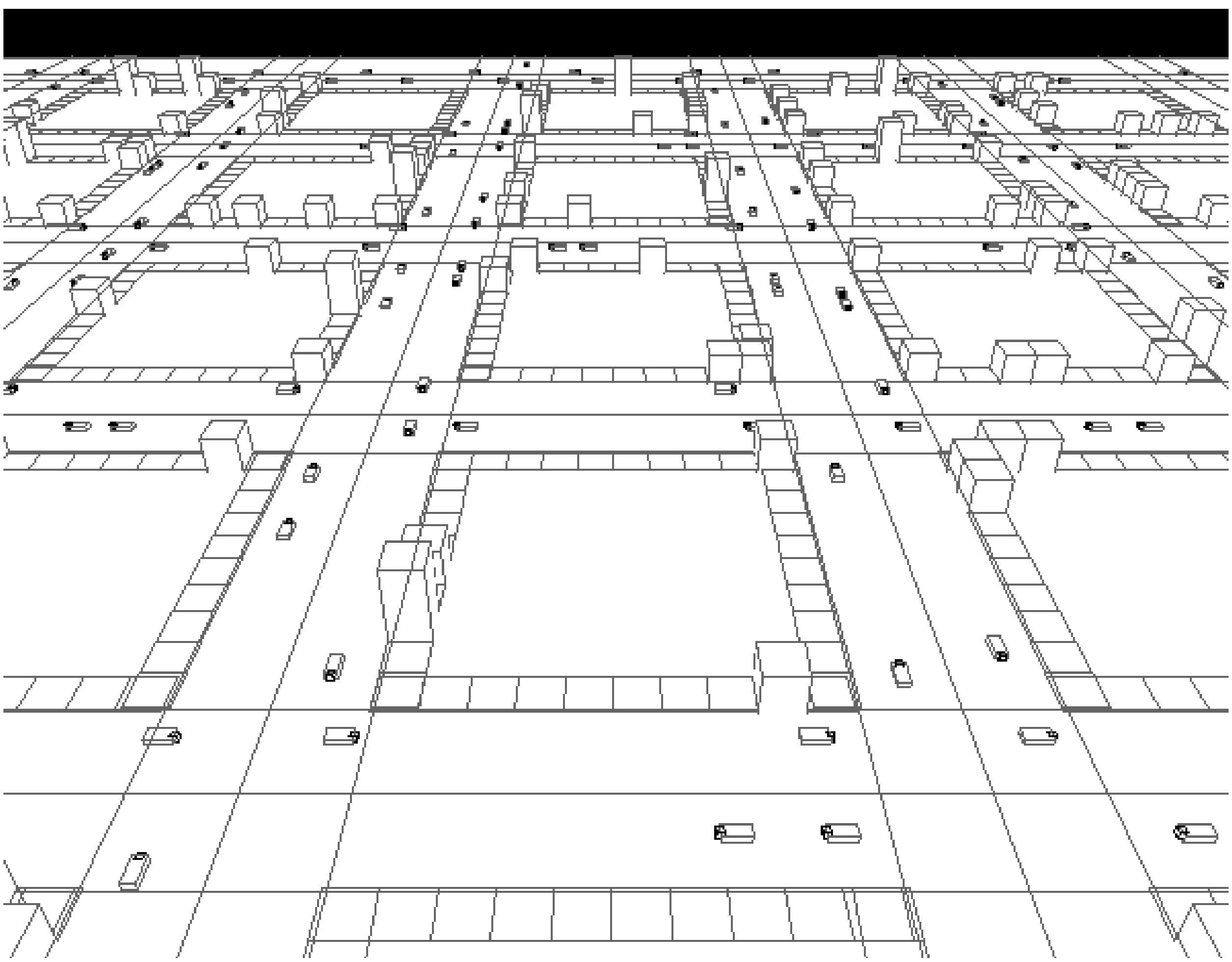
5.2.2.9 Experiment 9: Performance studies

A number of the vehicle concepts were modeled in a physical simulation engine used for computer games that approximate rigid body dynamics and physics in a simulated three dimensional environment. The boxes represent the generalized masses for the vehicle and constraint relationships position the blocks in space and allows for controlled articulation of the joints.

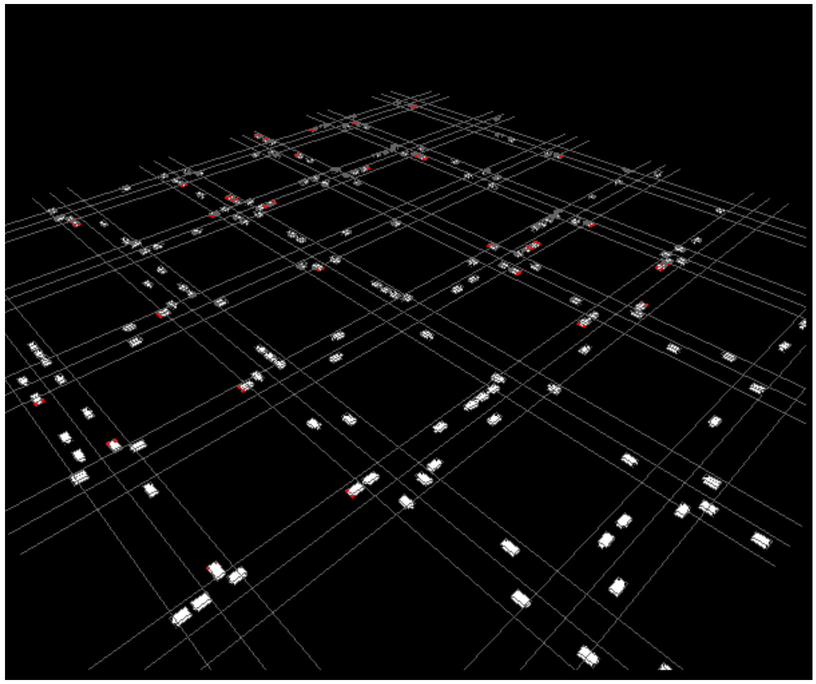




Behavioral studies of simple vehicles within a street grid. The study was implemented to track traffic effects from the interaction of vehicles. Implemented in processing.
Below, the grid is populated with buildings. Their volume represent

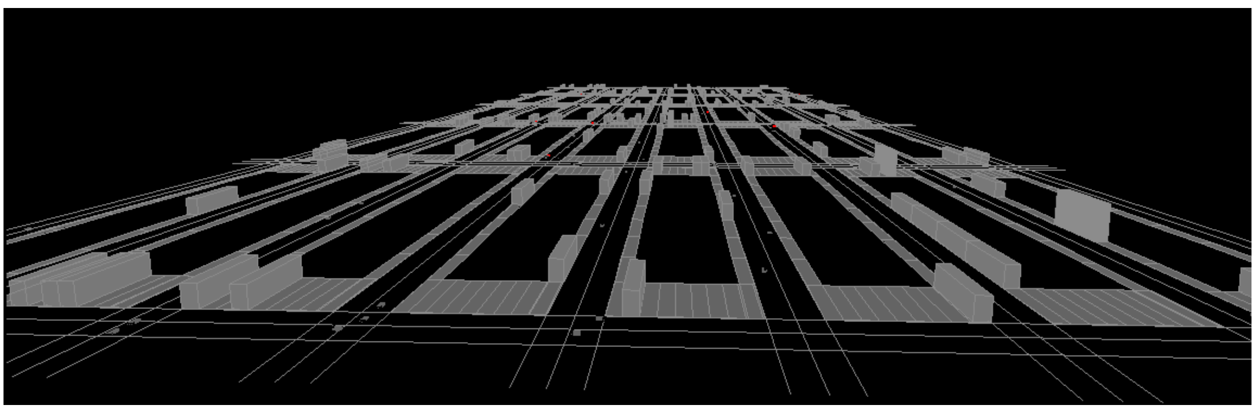
the number of cars heading towards them. Once the cars reach their destinations they keep circling the block. The goal was to study the interaction of density and traffic. Project by the author in processing.

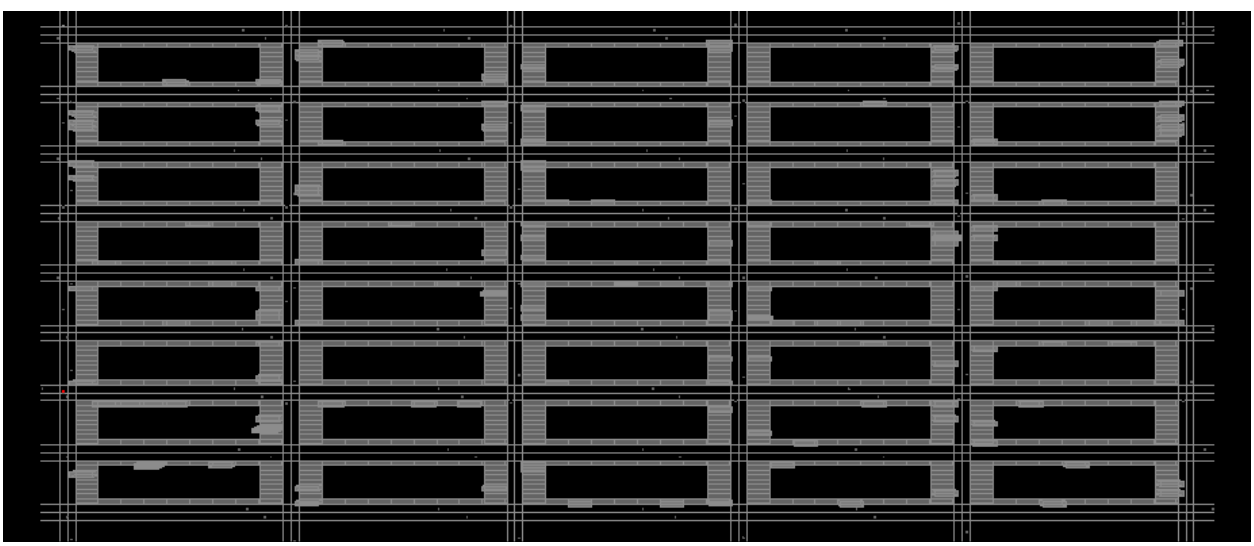


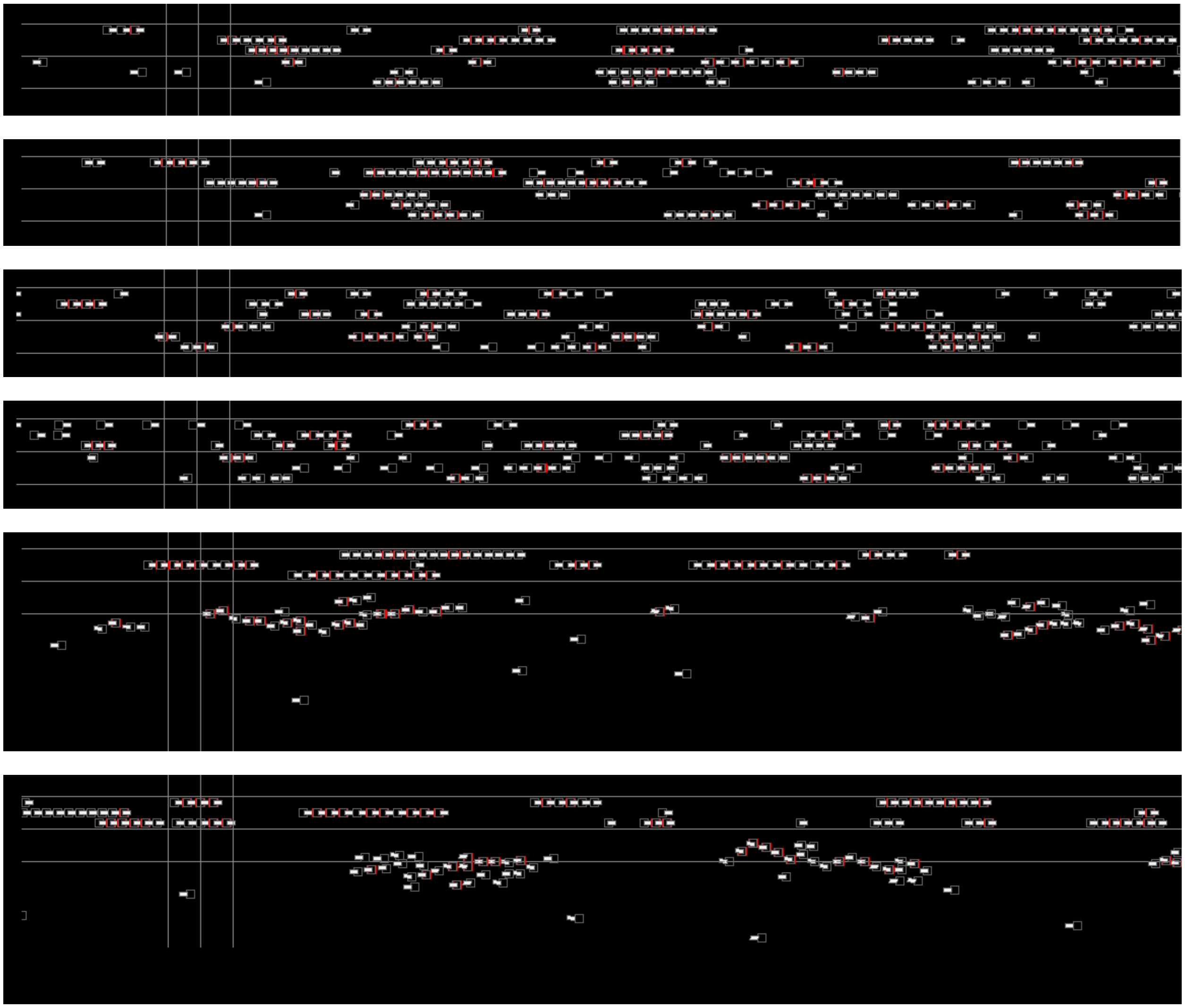
Three dimensional view with the breaking behavior of the cars at intersections.



A larger destination grid model.

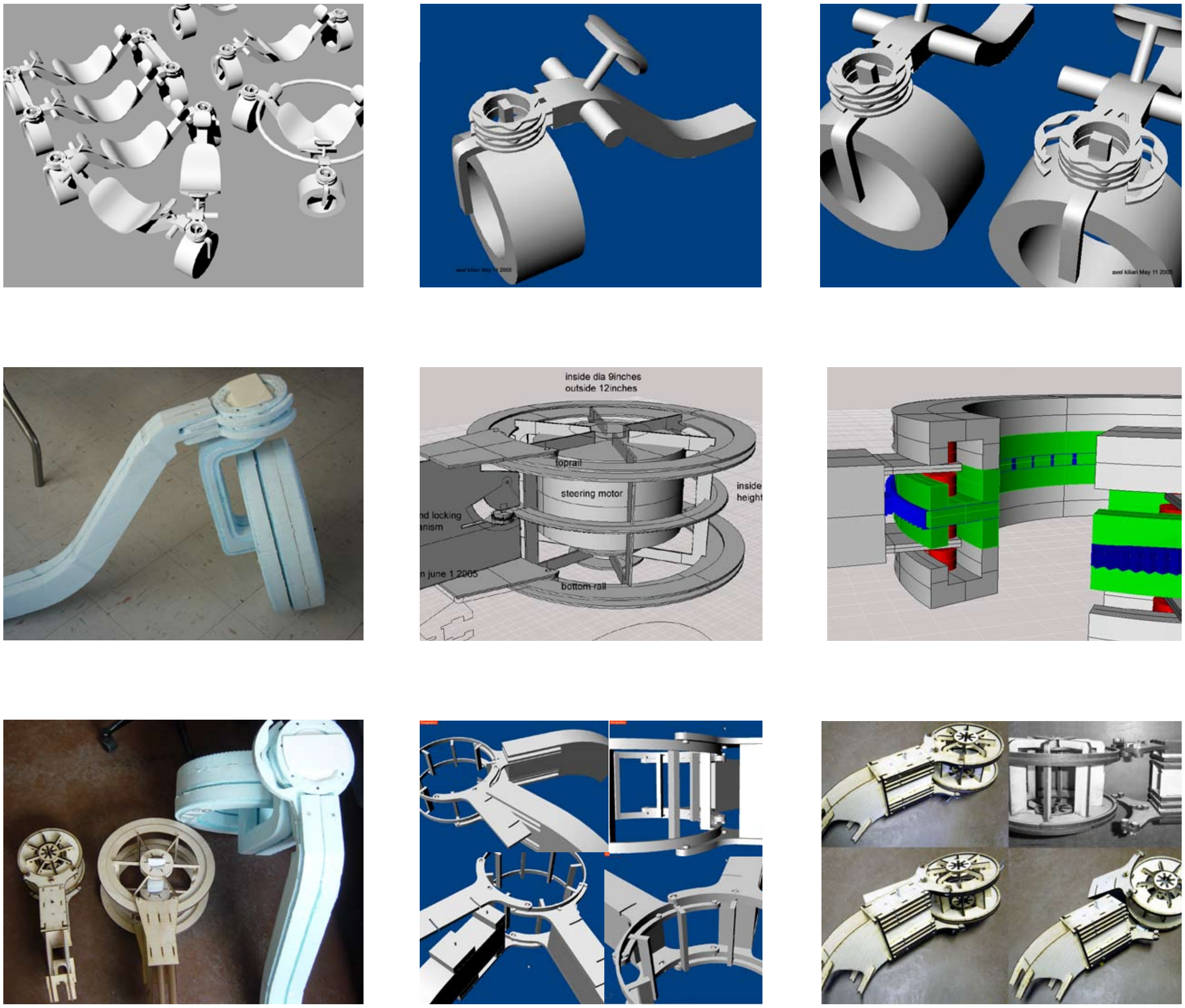






The advantage of this approach is that it is possible to quickly test the stability and performance of the vehicle designs without physical prototyping. This evaluation step can become part of the generative cycle of exploration and give input to the design direction. The performance studies were programmed in C++ in the open dynamics engine (ODE), an open source rigid body dynamics package. (ODE 2006)

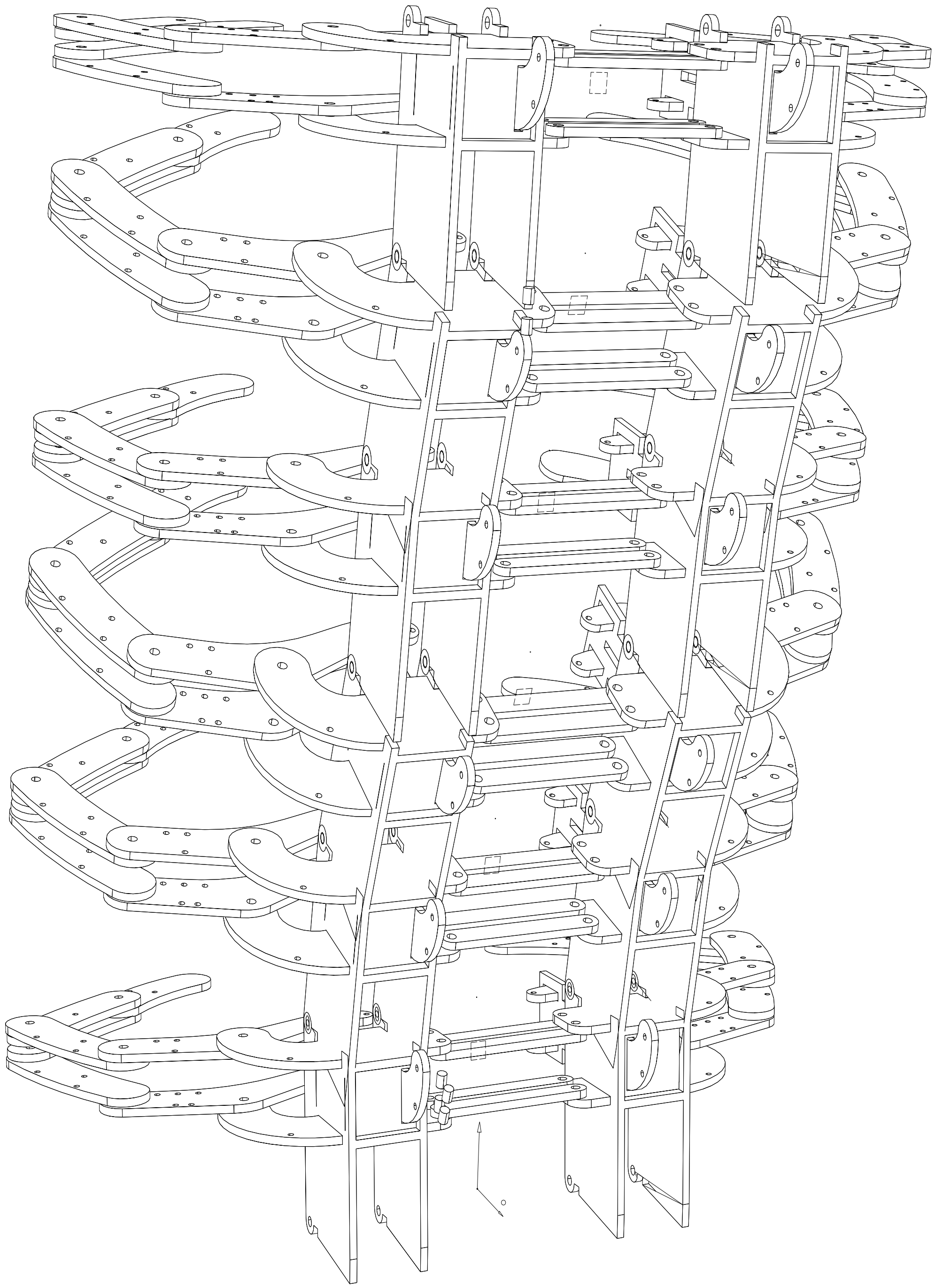
Exploration of different emergent traffic patterns under changing constraints.
Most constrained on top, least constrained on bottom.
Programmed in processing..



A small design exploration of a vehicle detail for the "zero car" developed by the Smart Cities Group.
The goal was to develop a detachable and angular adjustable wheel frame connection. Through several design iterations and prototypes a rail track solution emerged that allows for fast attachment and detachment of the arm. These small design explorations were not formally modeled for this thesis but exhibit many of the same traits of branching explorations as the main vehicle design exploration.
Connection studies by the author

5.2.2.10 Experiment 10: Behavioral studies

In parallel to the physical performance studies the author programmed behavioral studies that look at emerging traffic patterns from rules in the interaction simulated vehicles. The initial studies were based on grided street patterns and vehicle that would keep a certain speed dependant minimum distance from the cars in front of them. This safety envelope and the interaction of traffic patterns with the intersections create emergent traffic patterns. Similar studies have been done for a long time with the turtles programming concept developed at the media lab.
In an extension to the simple traffic patterns destinations were

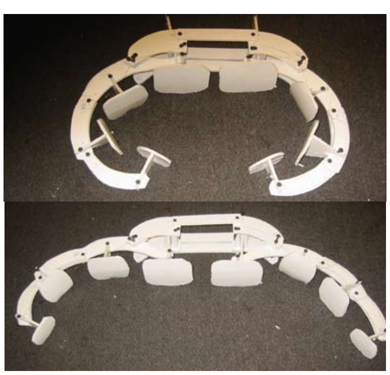


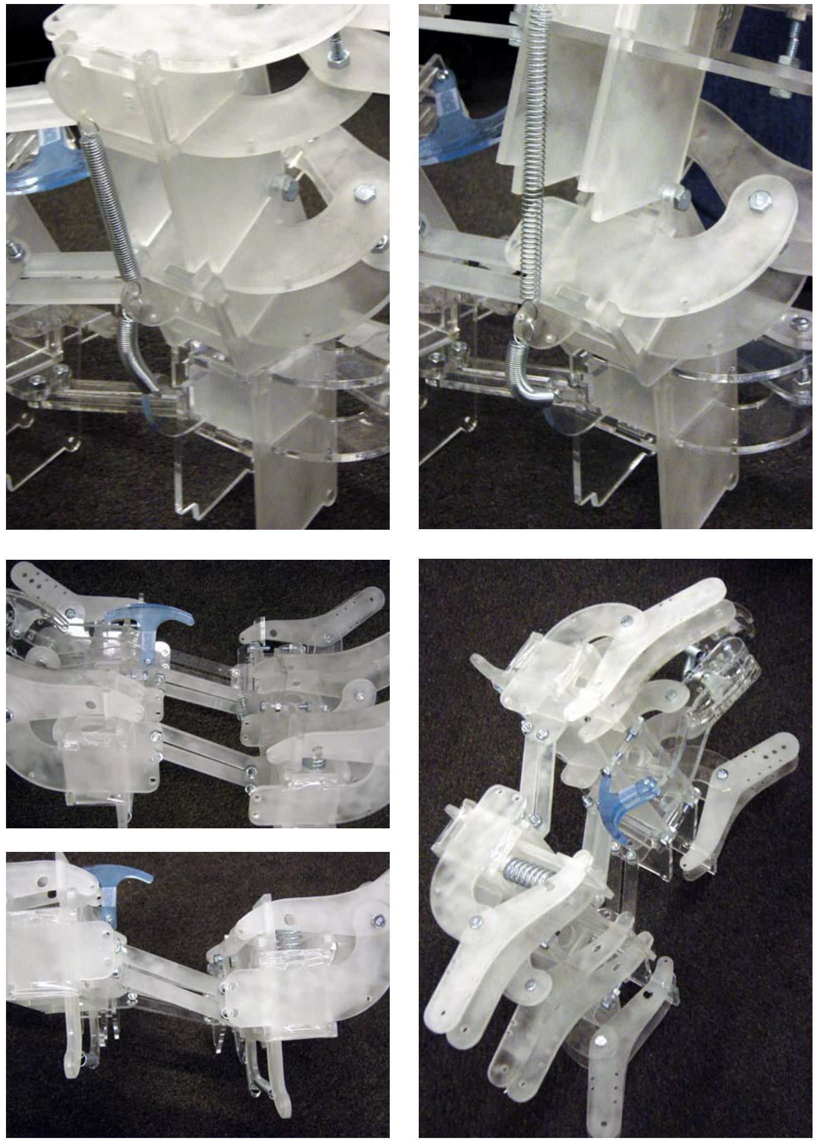
added for each vehicle and the turning behavior was based on minimizing the distance to the target in the city. Each target would generate a floor of a block like structure in the three dimensional model to simulate a very simplistic reversal simulation of traffic triggers building. This was developed to study traffic behavior with increasing congestions and dependency on urban densities. The

An articulated seat back study developed together with Patrik Künzler from an initial idea of a wearable seat that protects the passenger. This variation here tests a mechanical solution through laser cutting and plexi.

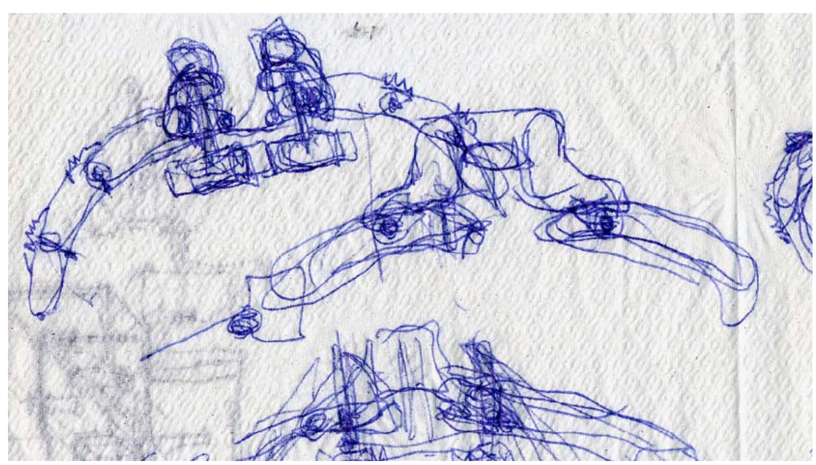
Opening chair back in CATIA.

Opening back in foam core sketch model

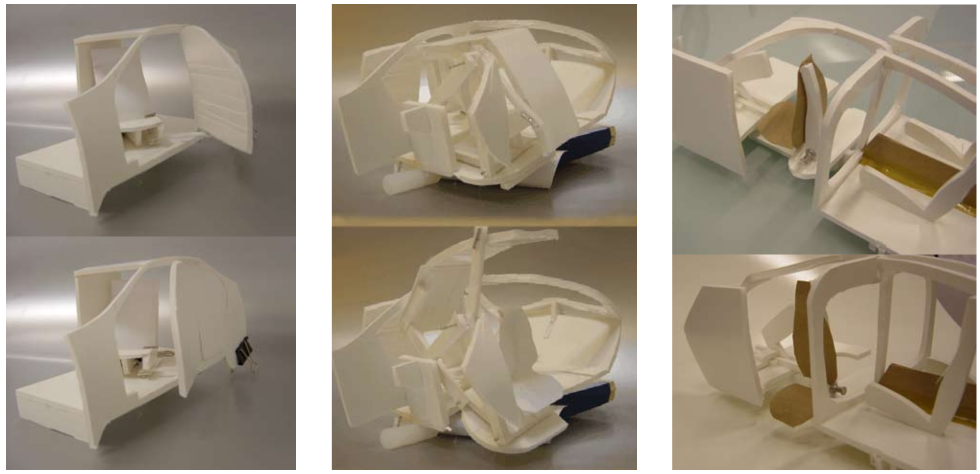


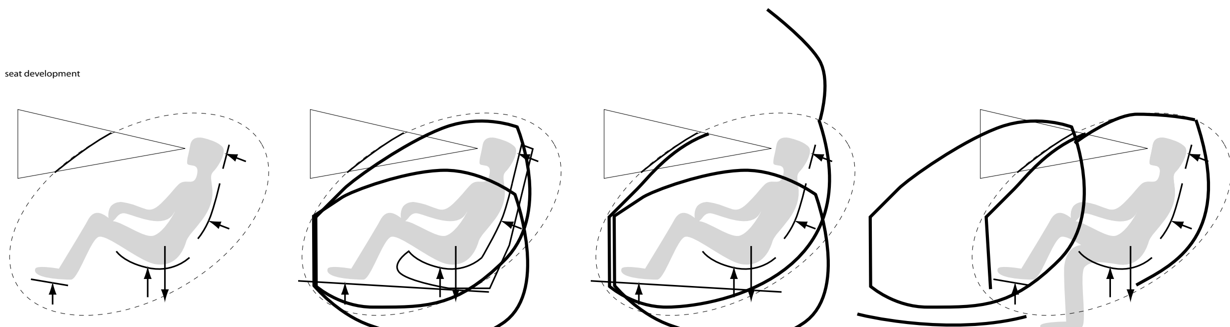


Articulated plexi assembly. Model development with Patrik Künzler.



Sketch of initial design.





behavioral studies wee programmed in the processing environment (Fry and Reas 2006) in JAVA through the creation of vehicle objects with built in behavior based on street color patterns.

Ingress egress diagram studies. Diagram as design driver for exploration.

Study one: Seat as furniture. Door closing the opening.

Study two: In addition to the side opening the roof opens as well in one kinetic movement. The seat swivels around with the floor and lifts the passenger to the outside.

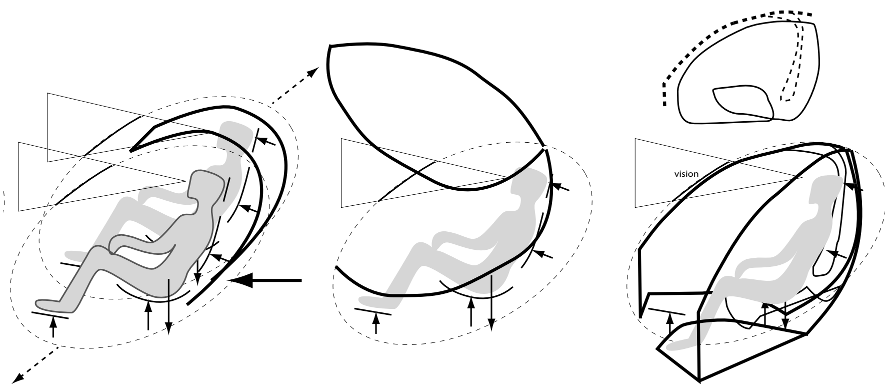
Study three: The floor as the door. The opening extends into the floor providing room for feet to drop down.

5.2.3 Selective Prototypes

The egress ingress design studies were the first instance referred to the thesis as *selective prototypes*. A selective prototype addresses a design problem mostly isolated of neighboring constraints in order to more easily come up with a novel solution that is not passed on conventions enforced by the context of a local intervention. For instance, it is very hard to come up with a new door solution if one starts from the chassis of a conventional car. Most of the context such as seat position, height and structural openings is fixed and the design becomes a problem-solving exercise rather than a rethinking of egress and ingress.

The following design series by the author studies the problem of ingress and egress focusing on one design driver a time, such as vertical freedom of movement below the seat, above the seat, and



Study four: An architectural solution for accessing the seats. The car splits in two along the middle and the parts slide apart, revealing a corri-door to walk to one's seats. The sides stay closed.

Study five: The clam shell design for one of the softcar studies. The hard part of the windshield lifts and allows access to the seats from the front.

Study six: Breaking apart the seat-as-furniture idea. The seat bucket supporting the driver is attached at the door and swings out with it. The backrest stays and is attached to the roof acting as part of the safety cell structure.

All designs and images: Axel Kilian

so on.

The door studies series conjugate the possibilities of openings in the outer envelope without respecting other constraints. Simple physical, kinetic mockups were used to so, animated through stop-frame motion to illustrate their functionality.

The door studies developed were:

5.2.3.1 Experiment 11: Drawer door

A simple drawer-like door replaces the hinge with a sliding track below the seat to allow for the horizontal movement of the entire seat ensemble out from the car. The advantage would be easier ingress and egress with combined door and seat movement. The disadvantage is the space requirement beside the car, as well as large structural levers.

5.2.3.2 Experiment 12: Rotation platform doors

The rotation of the door floor combination offers the possibility to access the seat directly and link door opening and seat orientation.

The platform provides room for the feet to be moved together with the seat. The large footprint required inside the car cabin to move the full assembly is a disadvantage.

5.2.3.3 Experiment 13: Seat door integration

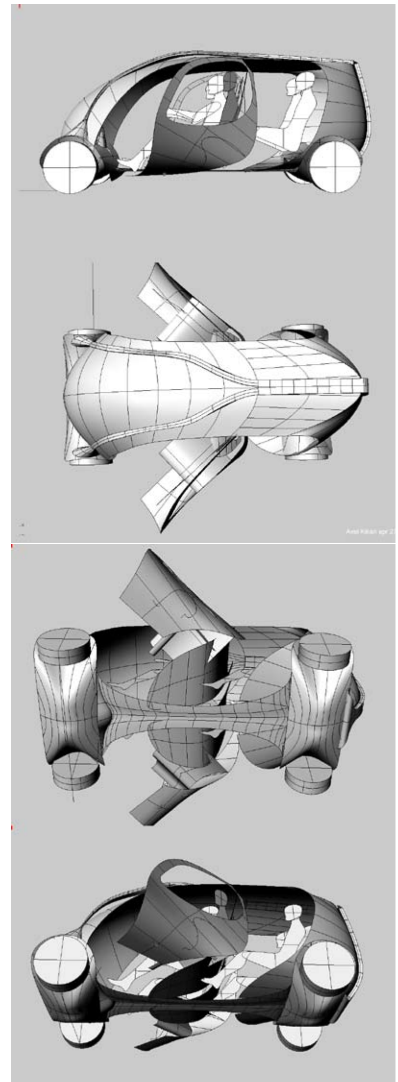
The seat-door link is developed into an integrated structural and functional unit that absorbs the force of side impact. The design splits the seat into a seat bucket combined with the door and the seat back rest, which is attached to the roof of the car protecting the occupant from above.

5.2.3.4 Experiment 14: Kinetic door

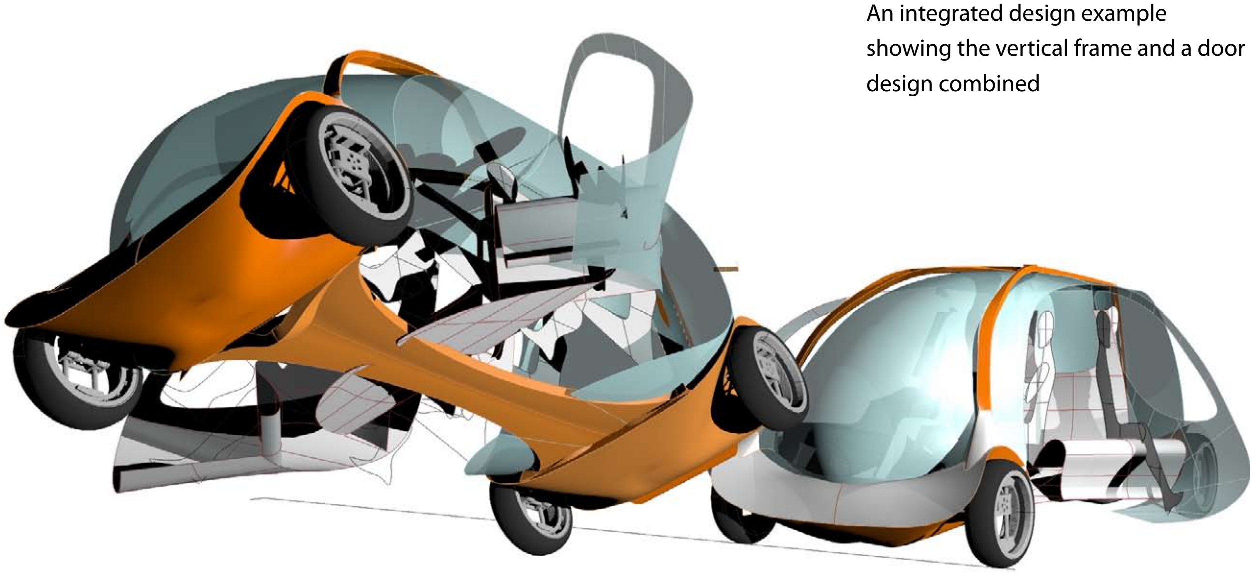
The kinetic door studies the possibility of a more complex kinetic construct that opens door latch, floor, and a roof panel simultaneously in one rotational move. The complex mechanism offers the possibility to resolve almost the entire cabin in favor of a kinetic envelope that responds to the driver.

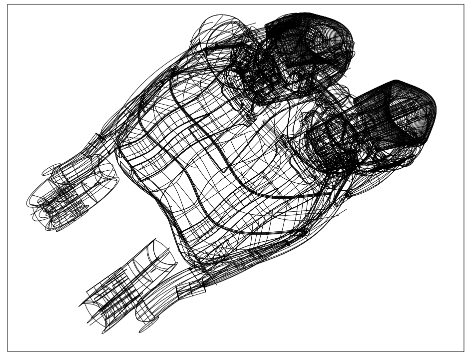
5.2.3.5 Experiment 15: Floo-door

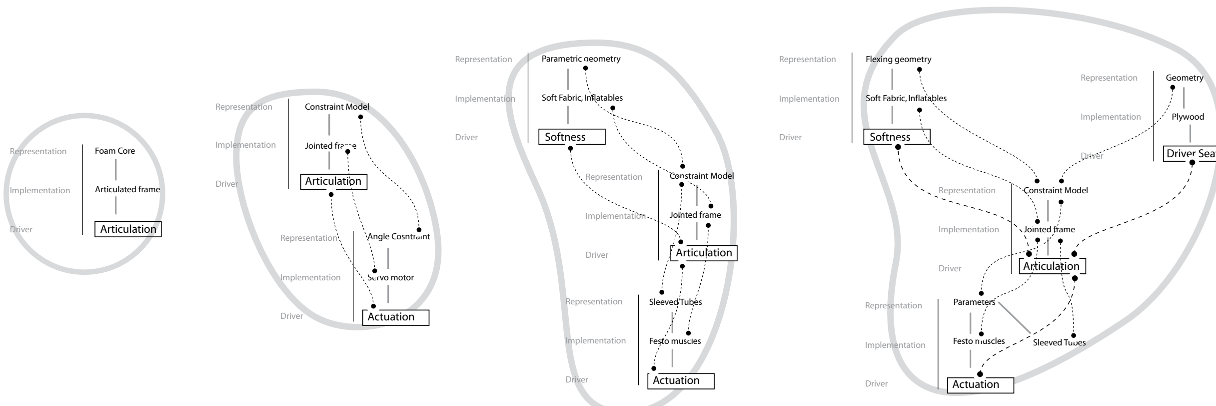
This door study investigates the use of the vehicle floor as part of the door. Opening the floor panel for access rather than the roof or the side panel only offers some advantages in terms of comfort. Once the floor is removed, the distance to the ground is that of an average height chair. This arrangement also has other implications for the overall chassis structure of the car. The implications for the car structure are as follows. Either the frame structure has to



An integrated design example showing the vertical frame and a door design combined







The branching exploration of the concept car design. Exploration for the establishment of a design problem from initially undefined constraints.

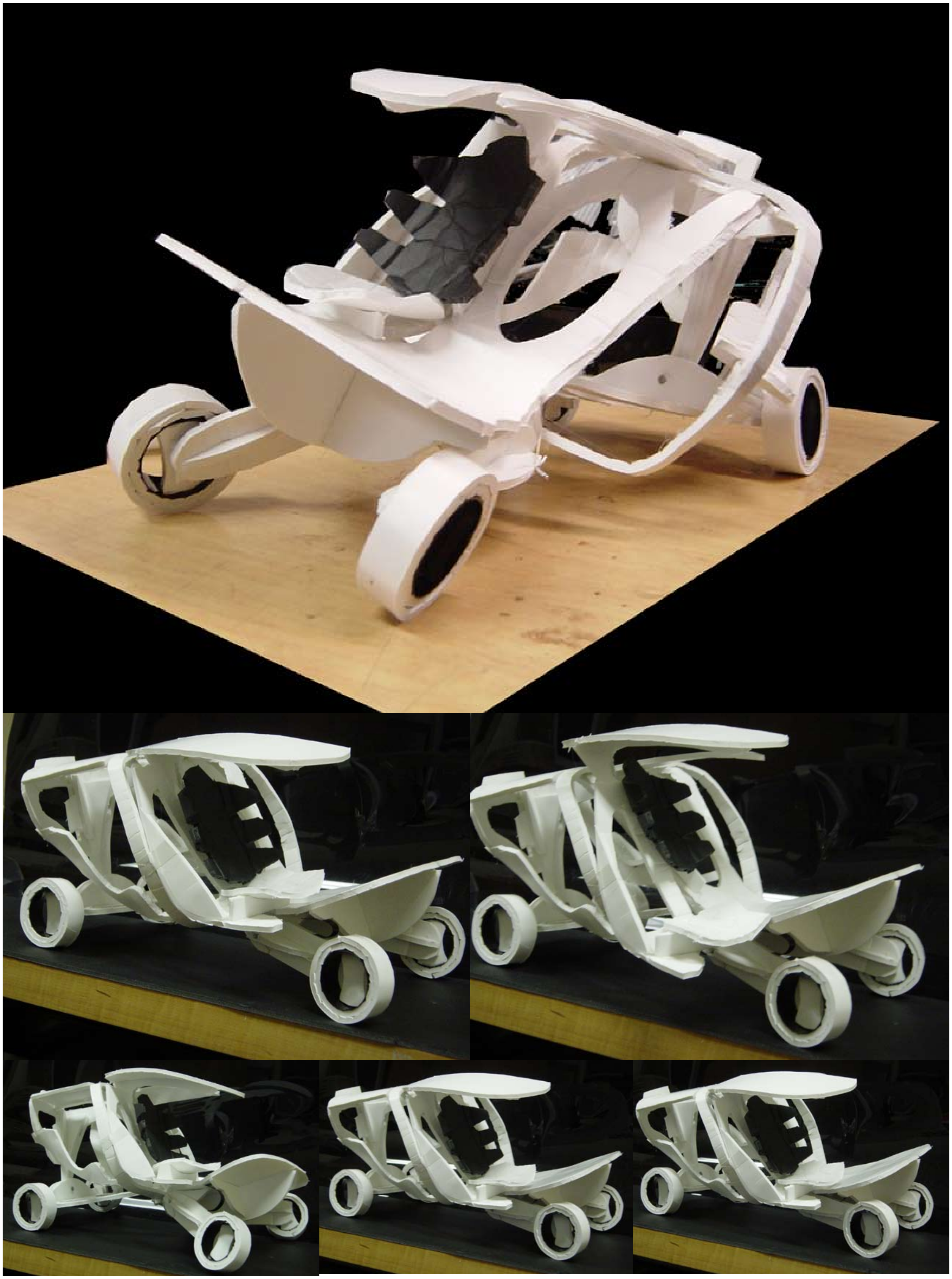
be tilted vertically or it shifts into the roof all together. This poses its own set off problems but also opportunities in constructing a different safety cell.

5.2.3.6 Experiment 16: Corri-door

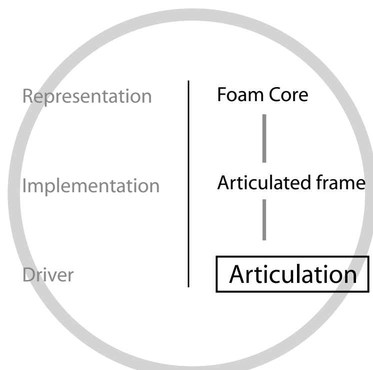
The corri-door is related to an architectural circulation scheme. Rather than providing access to each individual seat from the side of the vehicle, the design suggests a central access corridor at its center. This approach would allow for completely closing the sides of the vehicle, increasing crash worthiness for side impacts and also substantially affecting styling.

5.2.3.7 Experiment 17: Integrating selective door mockups

The door studies led to a series of holistic car designs that integrated



the door design features. At the center of the further development was the integration of the floor door and the vertical chassis frame. These designs would not have been achieved following a top down approach, but are a result of the selective prototyping approach. The design studies for a complete car documented in the following section take the selective prototyping approach a level higher by questioning the established car architecture even further through selective prototyping of the frame structure.



The articulated first athlete design by the author. Establishment of the main design driver for all following explorations: Articulation.

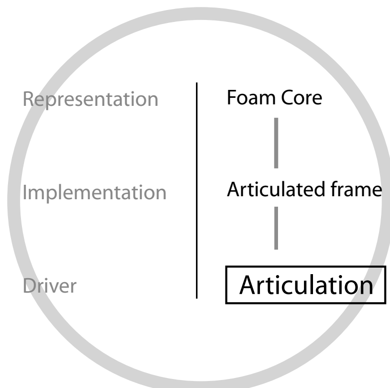
The foam core study shows the banking movement and the integration of egress and ingress into the frame design.

5.2.4 Main Experiment: The Athlete

The Athlete concept car design is a design experiment in which established vehicle components are replaced with alternative designs. This did not directly lead to better solutions than those that emerge from the established four-wheel, fixed chassis architecture, refined in thousands of engineering iterations. But it did help to review, question and partially reinvent some of the core elements of a vehicle architecture that in some ways has reached its limits. There is, of course, plenty of room for innovation in car design in terms of energy usage, active and passive safety, and the environmentally friendly use of resources, both in production and reuse. Even more important is the room for innovation in terms of the integration of the vehicle in the larger urban context or the transportation context in general. But at the same time a lot of design proposals have been made for those design goals and they tend to be somewhat repetitive in many of the core design

aspects.

The Athlete studies exemplify a different kind of design approach: one that does not try to answer the obvious questions, at least not in obvious ways, but rather suggests the study of an alternative way of motion and locomotion. This was done by looking at approaches from various areas of study, drawing more on robotics than on vehicle design. In addition, the brainstorming sessions took place in an interdisciplinary setting, with participants including architects, engineers, artists, and a doctor of medicine.

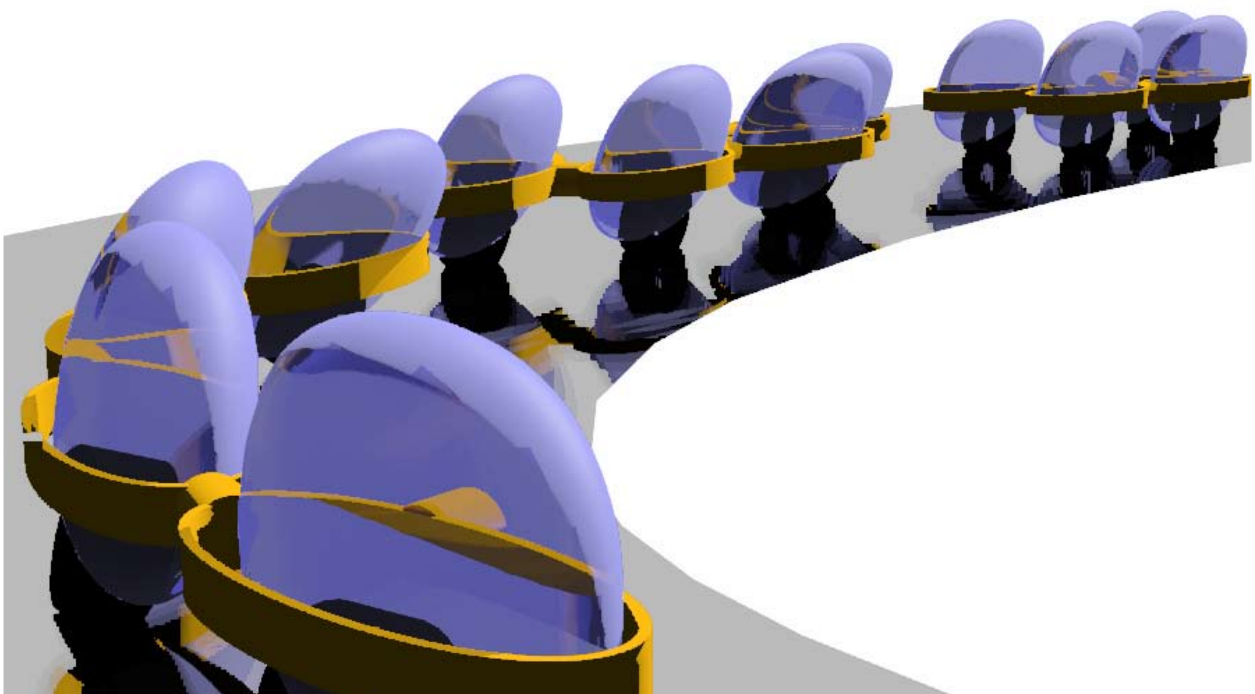


A simple four hinge kinetic study testing the concept of joined pods with a wasp like hip.

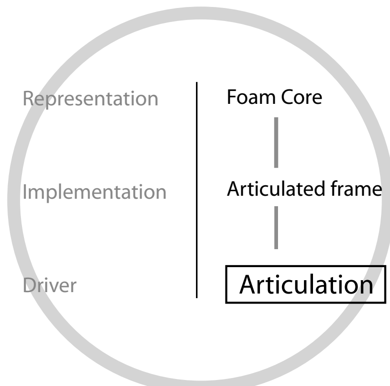
The beginning of the team work on the athlete design through brainstorming sessions. All work shown here is by the author.

In the beginning of the experiments there was the motivation to explore and a number of design features were chosen as starting points... Articulation of the chassis and an artificial muscle based actuation system were the basis of the initial exploration. An inspiration for articulation was the articulated BMW© skateboard. The design was translated into a first articulated vehicle. The structural skeleton explored the mixture of soft and hard elements, inspired by the use of plastics and rubber in athletic shoes as well as by pneumatic skins found in air mattresses and kites. One can refer to these features as the design drivers of the unfolding exploration. The softness for the skin was a response to the articulation of the skeleton and the athletic shoe comparison a precedent.

The main motivation of the Athlete project was to overcome the brick-like chassis of conventional vehicles. Even the most sophisticated high-end racing cars exhibit, almost without exception, a stiff, light frame connected to the wheel through suspension elements. Wheels are articulated to a certain degree to allow for steering.



Performance is defined by ever increasing horsepower of the engine, and the increase of the tire contact patch in combination with more sophisticated suspension systems. But why not give up the stiff chassis and look at the way a skier negotiates a downhill slope, to take one example from sports, shifting his or her weight, changing stance based on cornering speed and terrain conditions. Why not think about performance literally as performing through movement and graceful motion? Why cast driving as an individual activity and not as a shared activity similar to pair dancing?

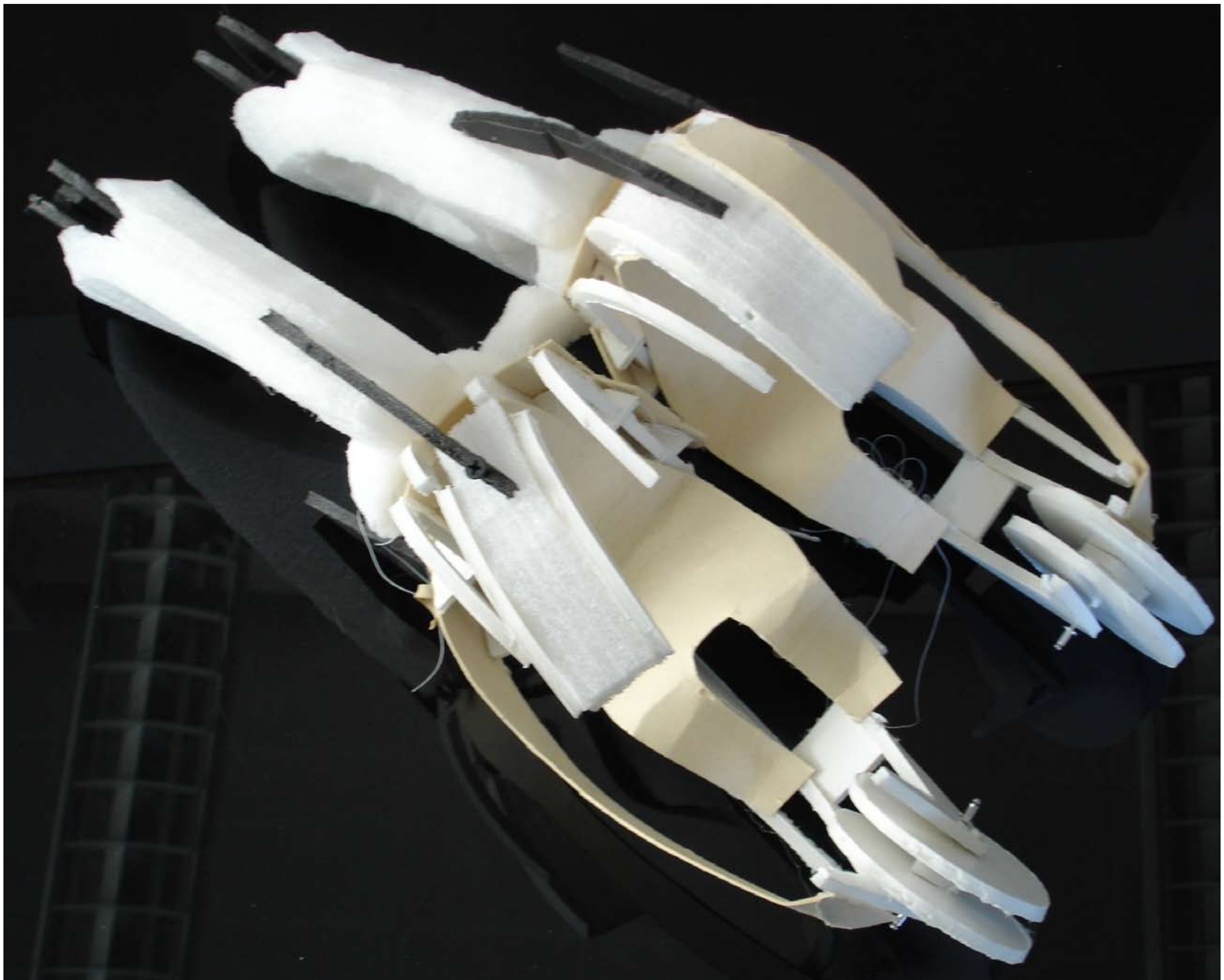


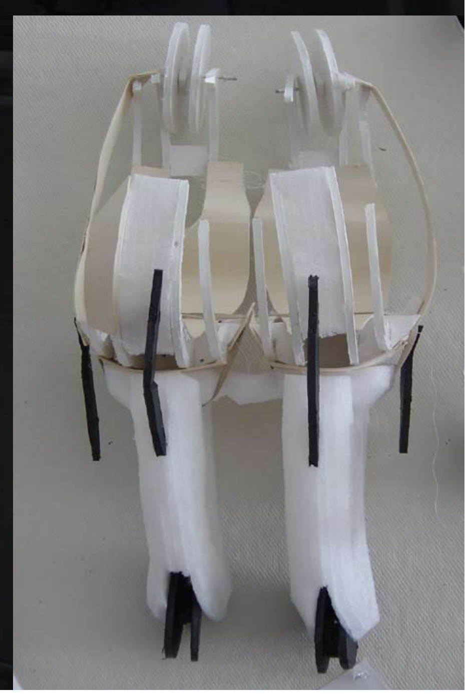
A variation of the jointed pods. The four pods can reconfigure themselves into a line for going through a corner and bundle up again afterwards. This is more of a swarm behavior than a question of chassis design. But it introduces the shifting of the center of gravity and the notion of coordinated, shared driving.

The idea evolved from a brain storming session with the group members: Axel Kilian, Patrik Künzler, Mitchell Joachim, Peter Schmitt, Kate Tan, Luis Berrios, Lorraine Gates-Spears, and Timocin Pervane.

These were some of the questions that emerged out of the design proposals for the first articulated Athlete car in the fall of 2004 by Axel Kilian. They provided the basis for the H-series group design by Axel Kilian, Patrik Künzler, Mitchell Joachim, Peter Schmitt, Kate Tan, Luis Berrios, Lorraine Gates-Spears, and Timocin Pervane. Finally, a partial proof of concept prototype of a three-wheeled Athlete called the Mini Athlete was built. It was based on a design concept by Axel Kilian, which incorporated the robotic wheel version by Peter Schmitt and the human seat by Patrik Künzler, Enrique L. Garcia and the author.

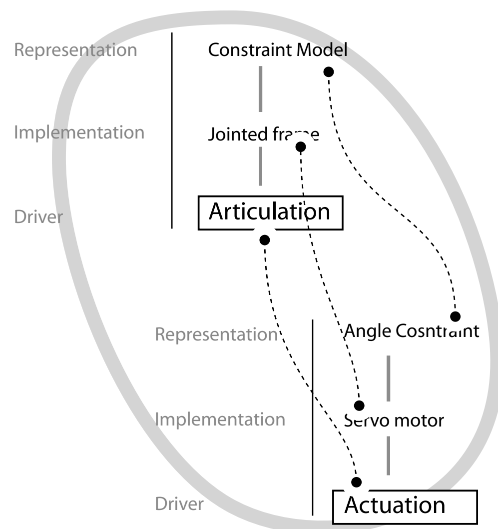
The design explorations by the author shown here were conducted not in pursuit of a specific design, but to formulate a design problem. The Athlete H-series is the most detailed of theses studies. The exploration focuses on a number of design representations adequate for different areas of the design. The term introduced for this process in this thesis is “selective prototyping”, a process that





iteratively focuses on chosen sub-domains of a design problem in order to develop prototypes that possibly could be fed back into the overall design. Eventually, the sum of such selective prototypes can accumulate in an overall design. In order to combine the novel design in the sub domains a new overall architecture may be necessary as well.

The Athlete design study approached the design of a car by questioning and redesigning parts of the established car architecture. These core components included the frame, the



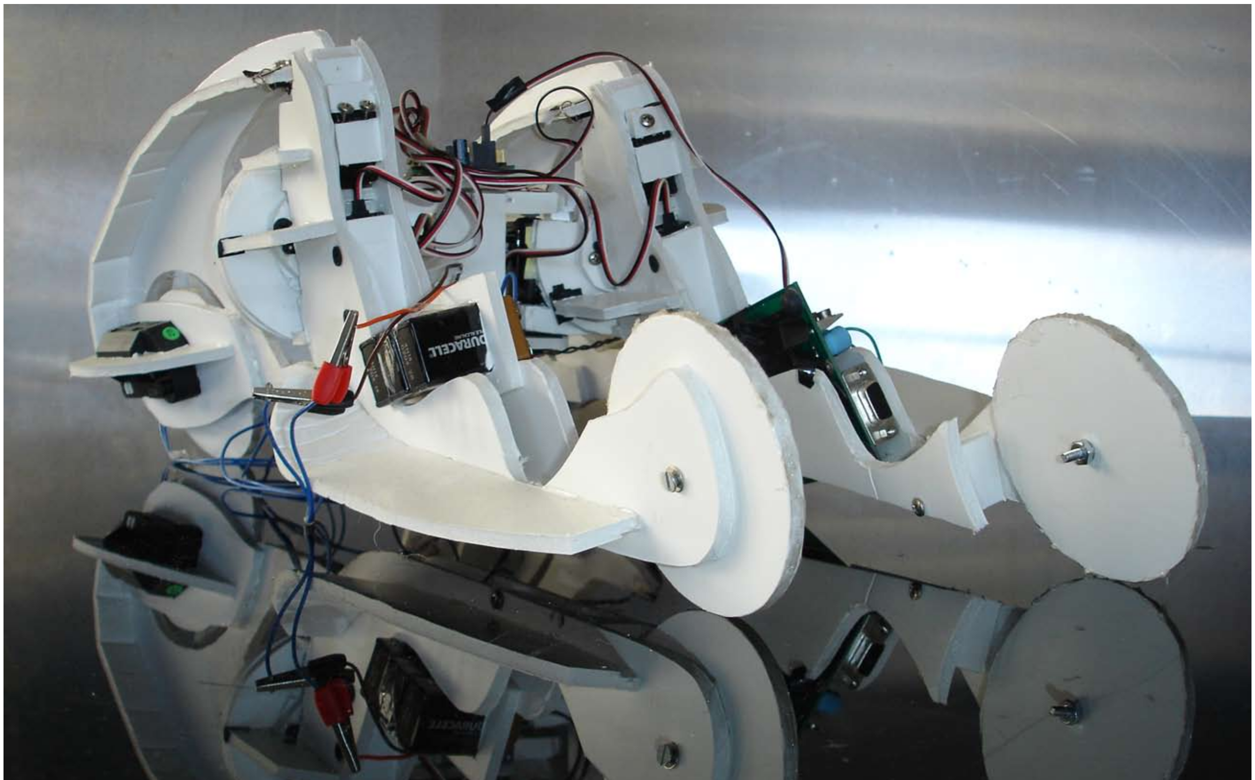
The previous design gets integrated into the initial articulation study and the hip joint is created from it. The design also integrates the consideration of actuation for controlling articulation.

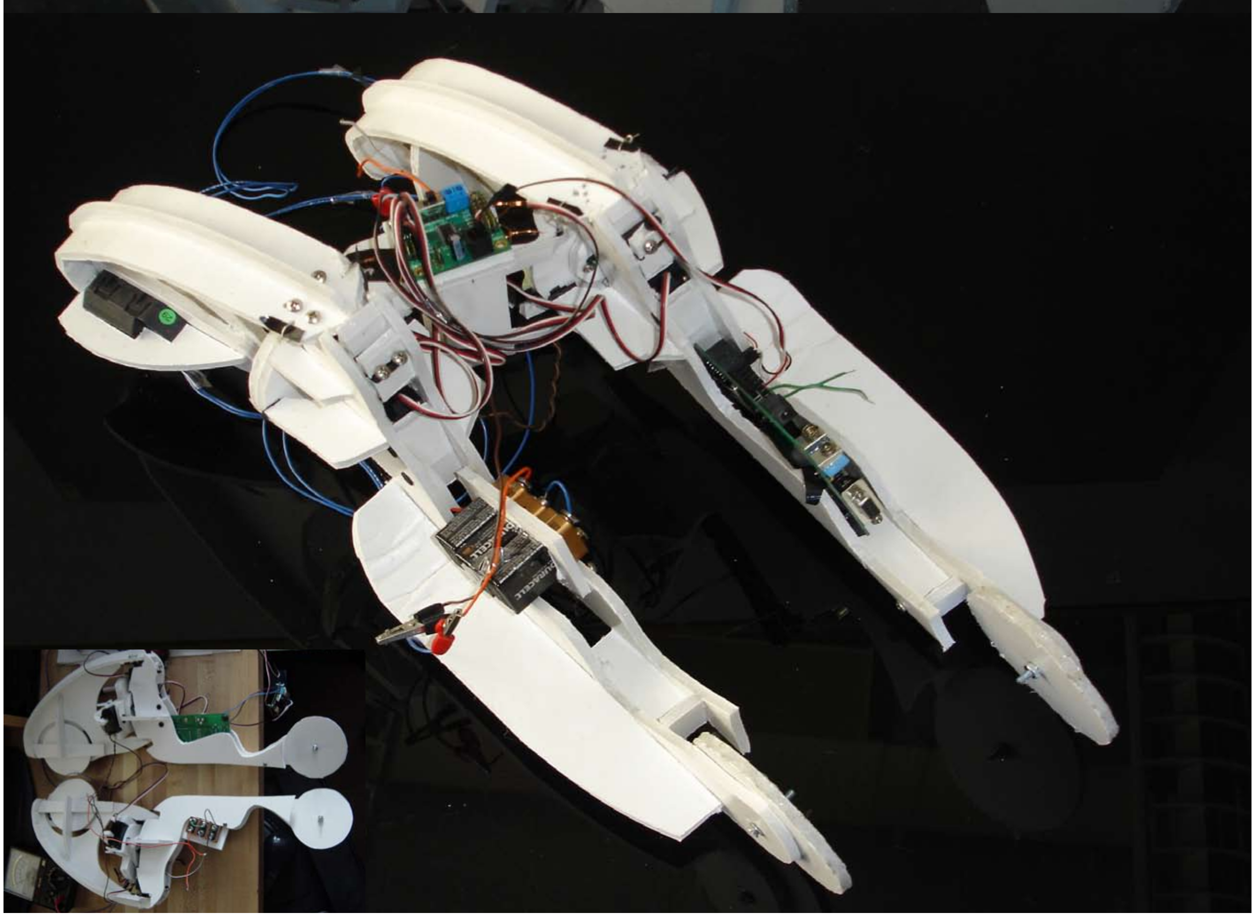
envelope, the steering and the seat-driver relationship. Prototyping the subparts of the design allowed for the emergence of novel aesthetics, because the subparts do not have to comply with the immediate pressure of the design conventions of a complete car.

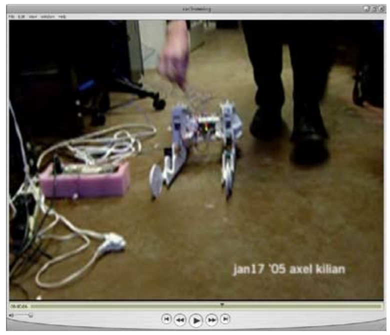
The following sketch designs go through *selective prototyping* studies for the kinetics of the articulated frame, the form factor of the envelope, and the pneumatic muscles.

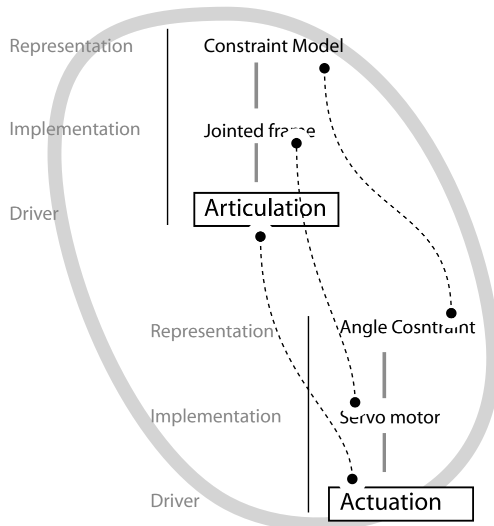
5.2.4.1 Kinetic constraints

Kinetic constraints were tested using a foam core sketch, externalizing the first rendition of what was later called the H-series. The mockup model served as a fast, physical constraint solver to capture and test an idea during brain storming. It also served as a


DURACELL





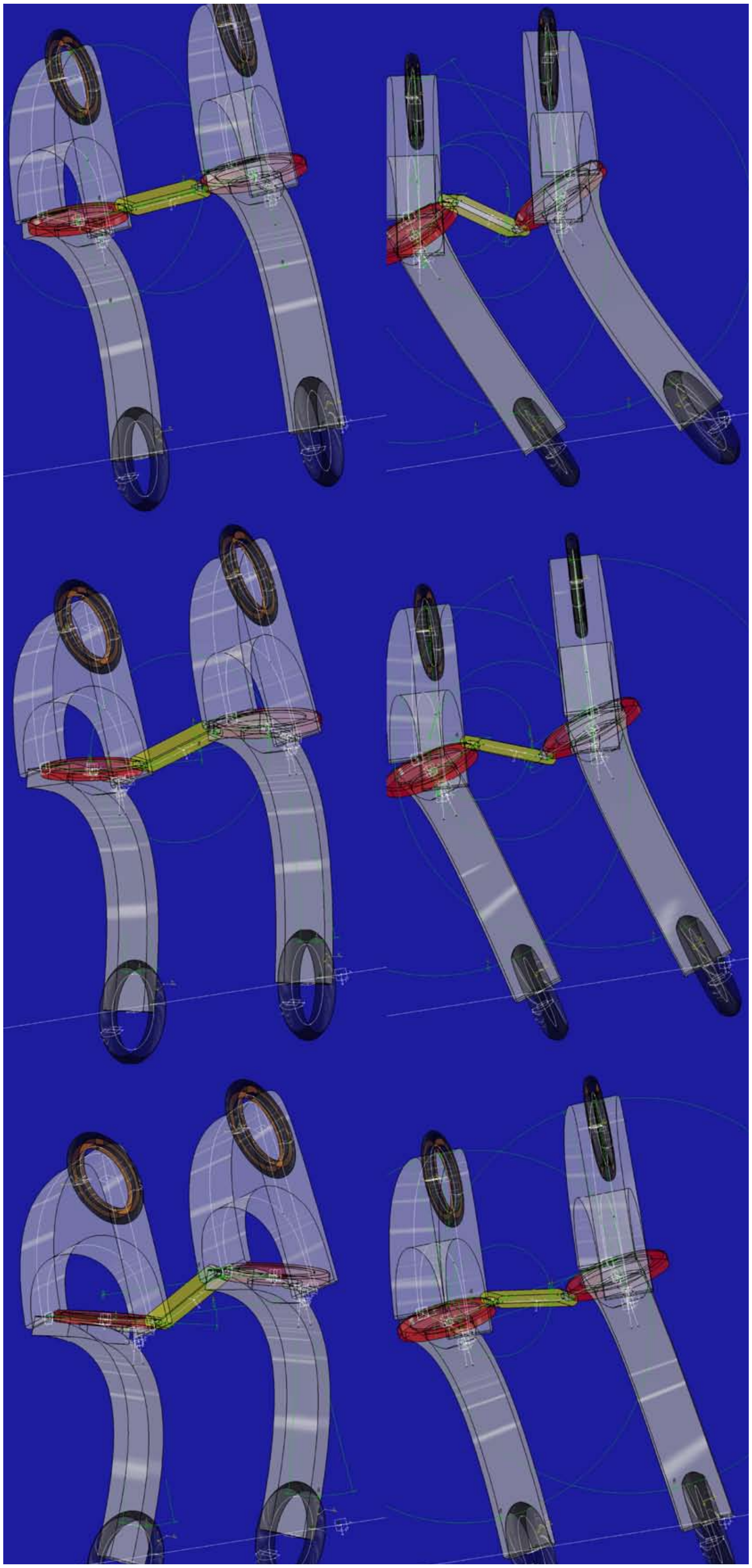


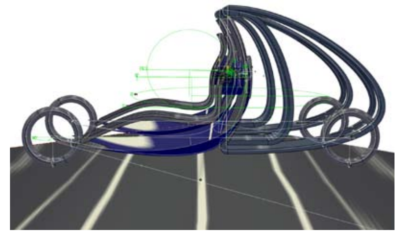
To further understand the relationship between articulation and actuation for the overall behavior, a servo driven fully actuated model is built by the author. It uses a stamp micro controller to drive the servos.
(top) Movie stills from a test run by the author.

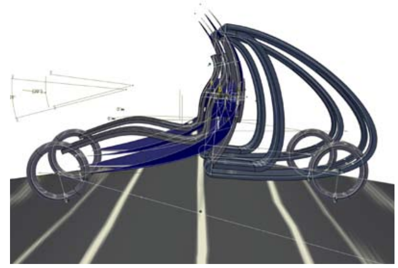
tangible model to explore the degrees of freedom of the wasp-like joint.

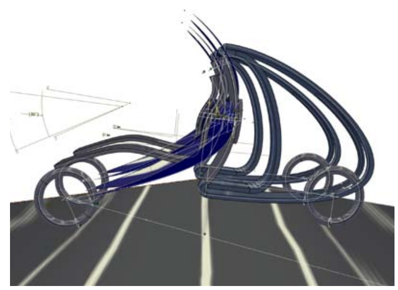
5.2.4.2 Formal constraints

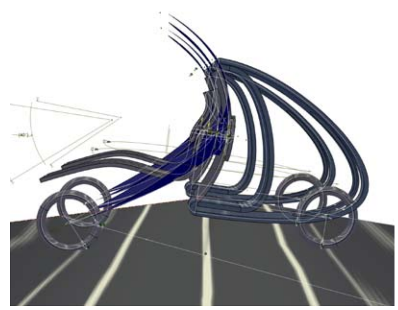
A series of foam core models were developed in parallel to explore formal constraints. These models started to develop a formal vocabulary for the frame typology. A series of foam core models were developed in parallel to the kinetic studies for the articulated

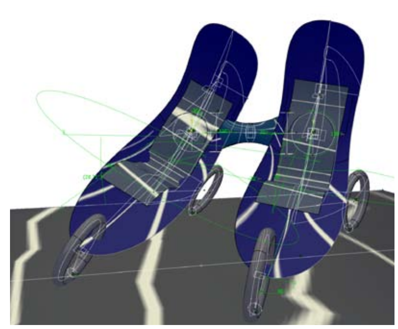


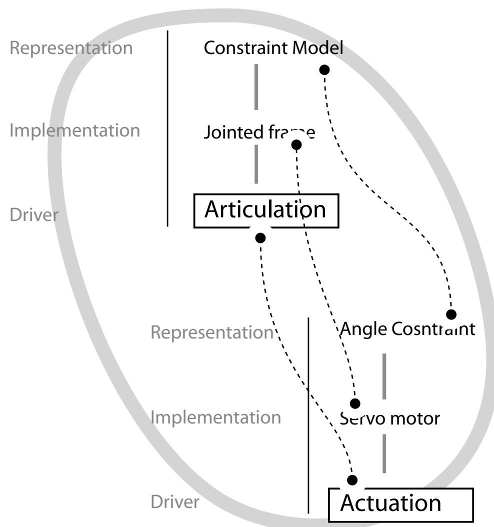












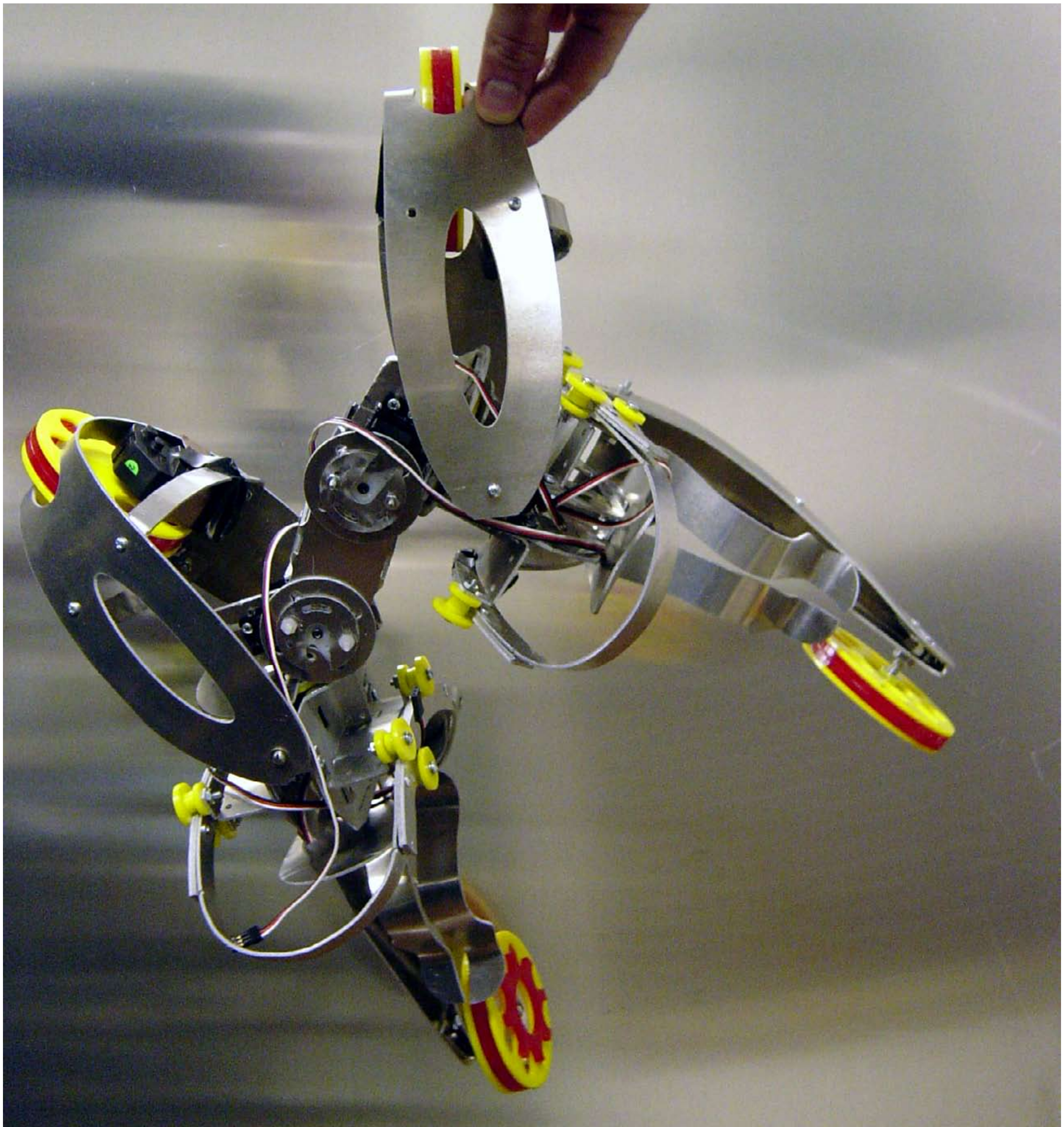
Two iterations of constraint solver models. These models are not built to develop the formal aspects of the vehicle but to selectively test the interaction of the degrees of freedom (DOF) in the assembly. The entire model is assembled out of 13 parts that are chained together through geometric constraints. The relationships amongst the constraints are defined through equations or direct mappings.

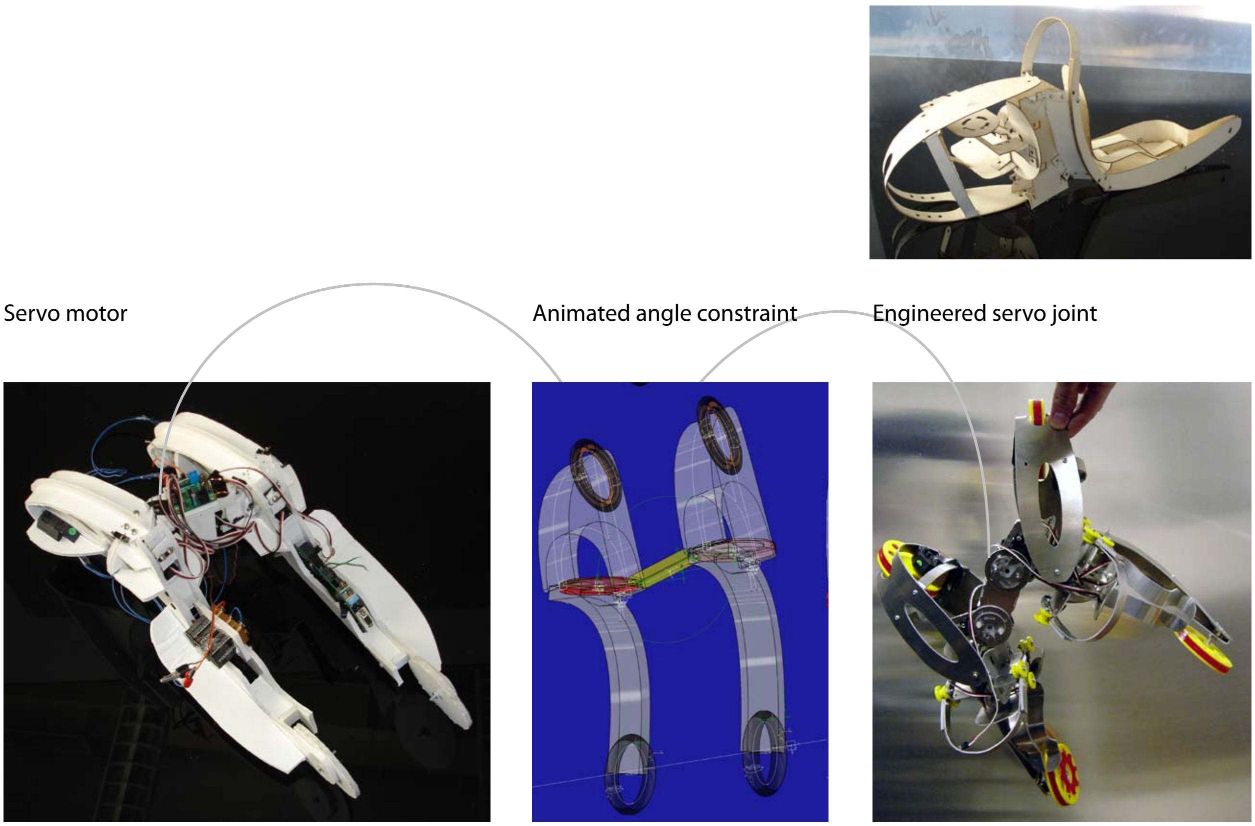
The left set is a six degrees of freedom model. The right an eight degrees of freedom version. In addition to the previous one it can raise itself up and down for parking and curves.

frame. In particular the hip section connection the four wheel arms was studied. The hip section posed the biggest challenge as it is the most removed from the conventional car topology. Through several iterations the volumetric and proportional parameters of the design were refined and integrated into the visual rendering studies.

5.2.4.3 Actuation constraints

The articulation of the under-constrained articulated skeleton is crucial to both its aesthetic and function and was explored with a physical actuated 1:10 scale foam model using servos to animate the joint constraints and controlled overall by a stamp parallax© micro controller. The micro controller was programmed in Java using the javelin© Java based chip, to go through a cycle of coordinated rotations for all six degrees of freedom. In addition





Six degrees of freedom (DOF) physical, servo controlled model.
Physical constraint solver

Six DOF digital constraint solver model.
Translation of representation.
Leads to redesign with two additional DOF.

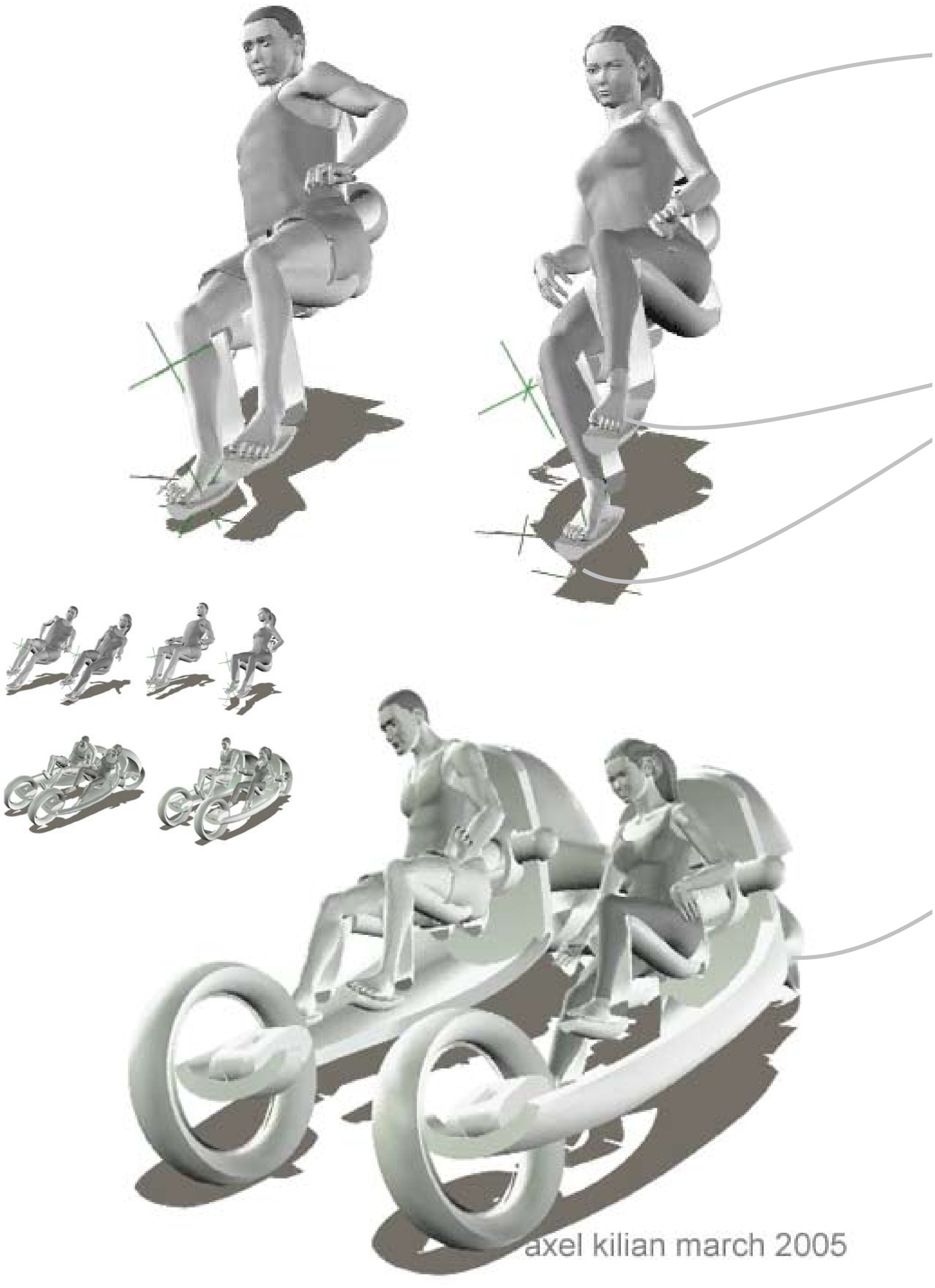
Eight DOF redesign physical mockup incorporating the changes.
Change of fabrication triggers further design changes.

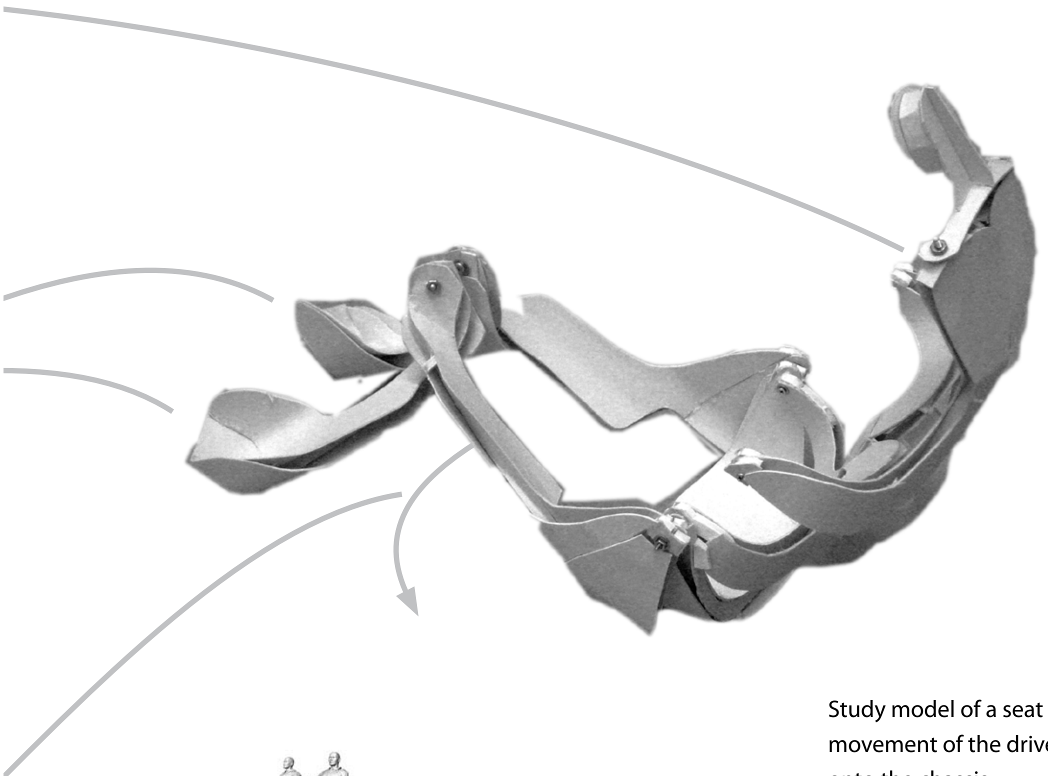
The complexity of the physical aluminum model has increased substantially from the foam model due to the need to fabricate each part from a digital file and the increased number of engineering details. The model has eight degrees of freedom. The two new ones are to lift and lower the chassis during cornering and parking.

the model was equipped with two motors for propulsion. The model was test driven and performed well in many aspects of the articulation but also exhibited some problems with overall stability due to the imprecise nature of the foam skeleton. The model was a design development sketch study built in less than 5 hours with only marginal digital modeling done up front. It set the precedent for the later aluminum model for the placement and functioning of the joints.

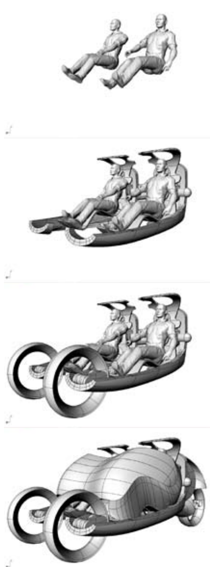
5.2.4.4 Three dimensional constraint solving for six degree of freedom frame

The testing of constraint models for the six degrees of freedom articulated frame was moved into a digital constraint solver


axel kilian march 2005



Study model of a seat that senses the movement of the drivers and maps them onto the chassis.
Later this design was further developed by Patrik Künzler, Enrique L. Garcia and Axel Kilian
Chair model and Image: Axel Kilian

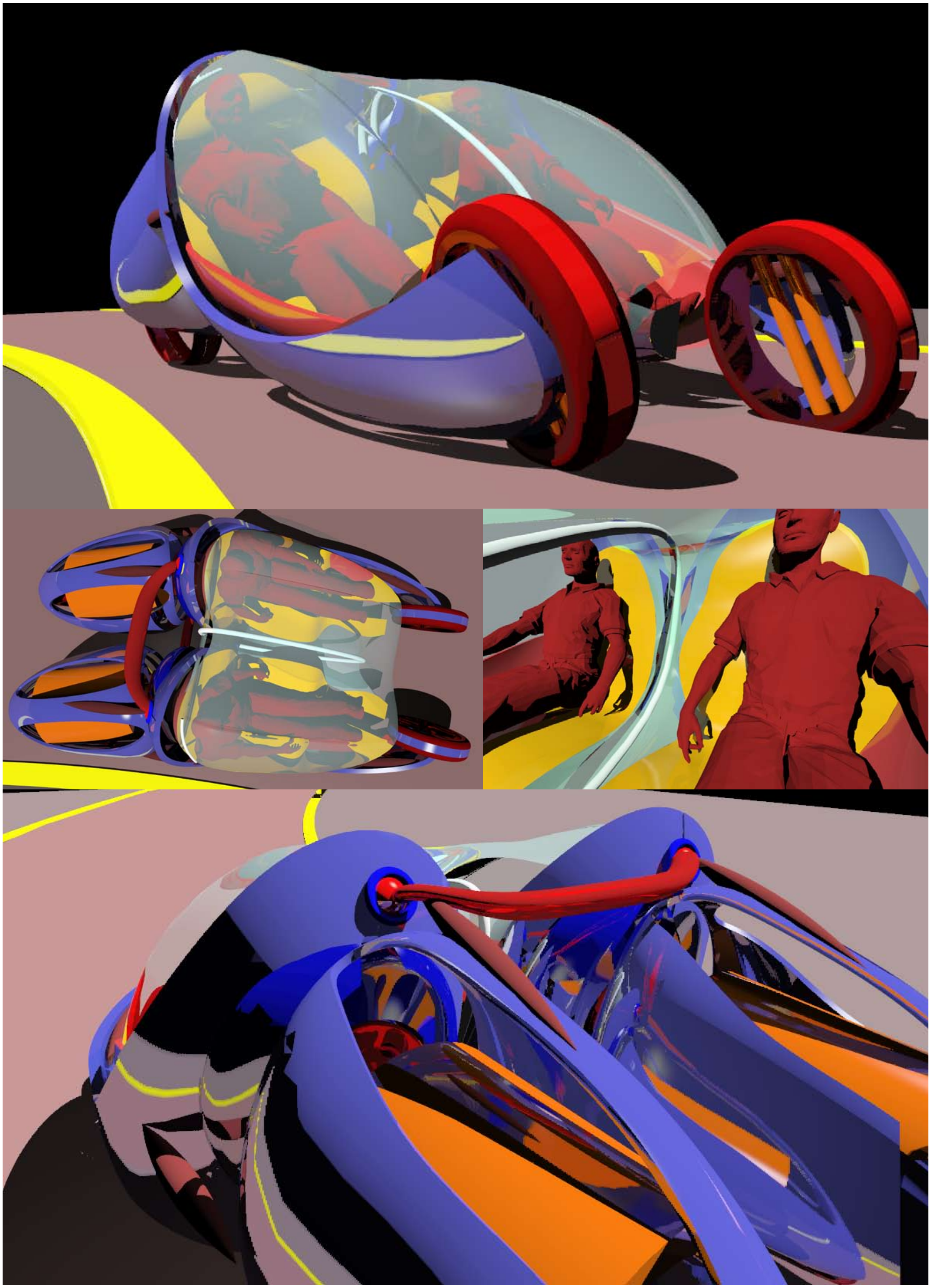


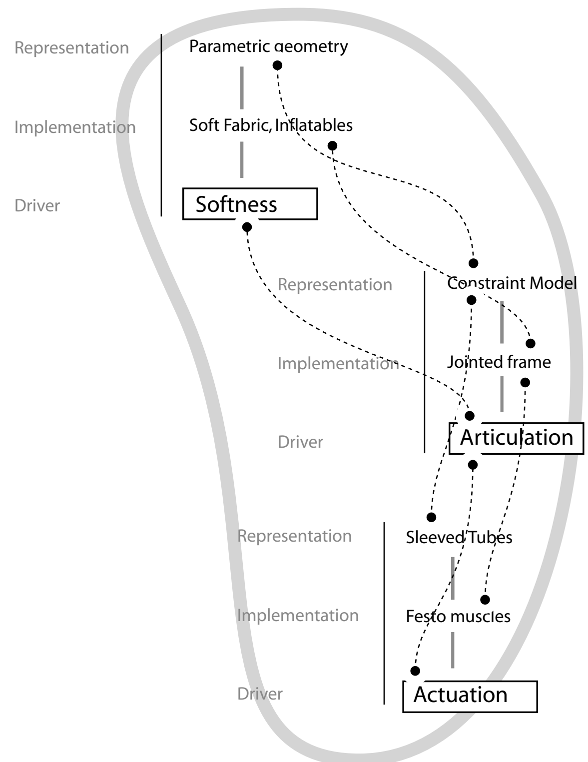
The study of human motion and how it can be mapped onto the degrees of freedom of the car. Building the car around the drivers is the goal of the experiment.

environment inside CATIA© to better test the dependencies without the inaccuracies of physical implementations of the joints. The study let to a remodeling of the joint system to eight degrees of freedom in order to compensate for the non planarity of the four wheels in turns in the six degrees of freedom version. The second CATIA model shows this iteration. The design progressed substantially through the translation of the physical constraint study into the digital constraint study based on the comparison between the configurations of the two models.

5.2.4.5 Fabrication constraints

The next design representation translation was back from the digital constraint model to a physical implementation with servo



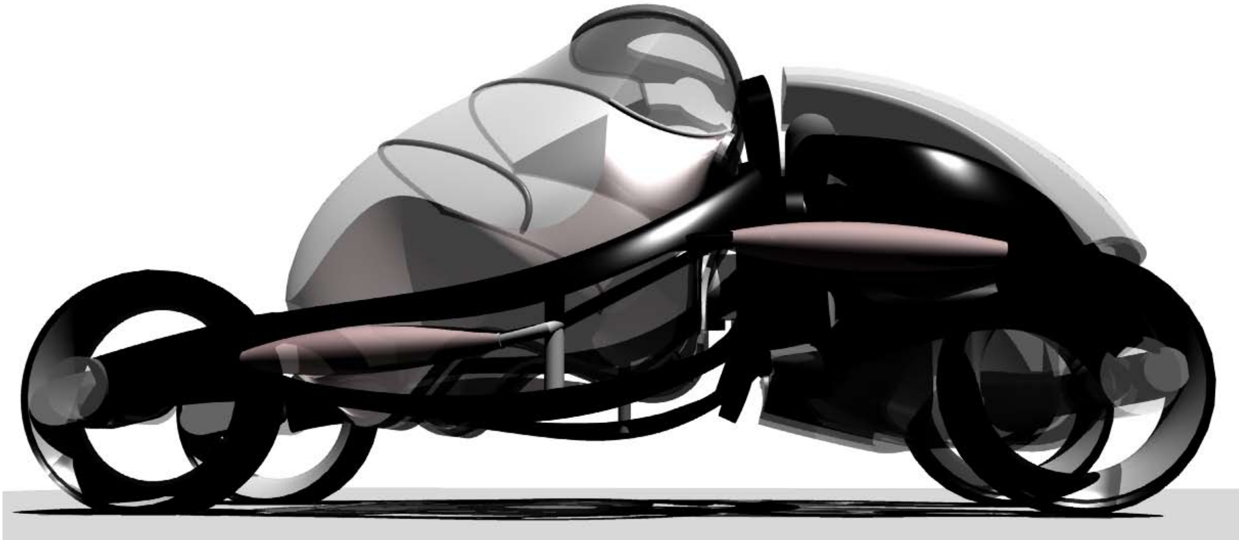


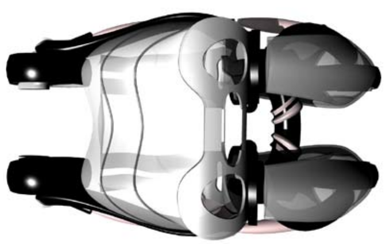
In the next iteration, softness is integrated into the design. An integrated design study of the design responses so far is shown on the left. The soft shared cockpit follows the movement of the occupants and the movement of the articulated H-frame.

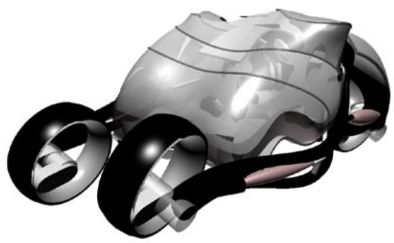
actuation to test the additional two degrees of freedom. The design was translated to a CNC machinable prototype that offers more precision in actuation and engineering of the parts also introduces new formal elements that pose possible avenues for further development. The aluminum model design introduced the design solution for the two additional degrees of freedom by separating the front beam supporting the front wheel from the main seat support. This allows the car to arch between front and back wheel to compensate for any non planar wheel alignment.

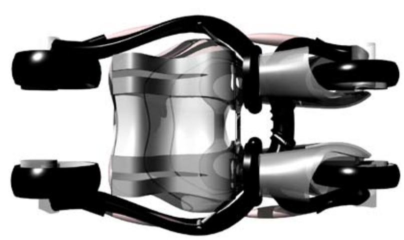
5.2.4.6 Mapping movement of the body onto the frame

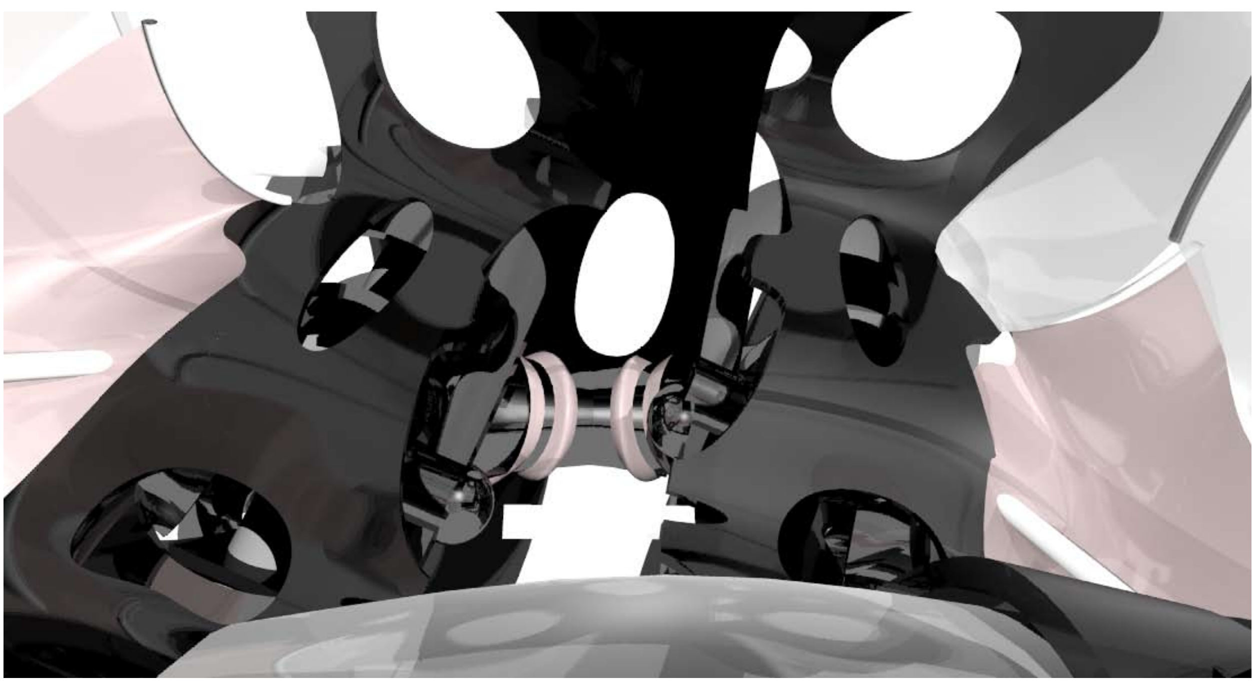
An inverse kinematics study was developed to study the mapping of two person's movement onto the articulated frame of the

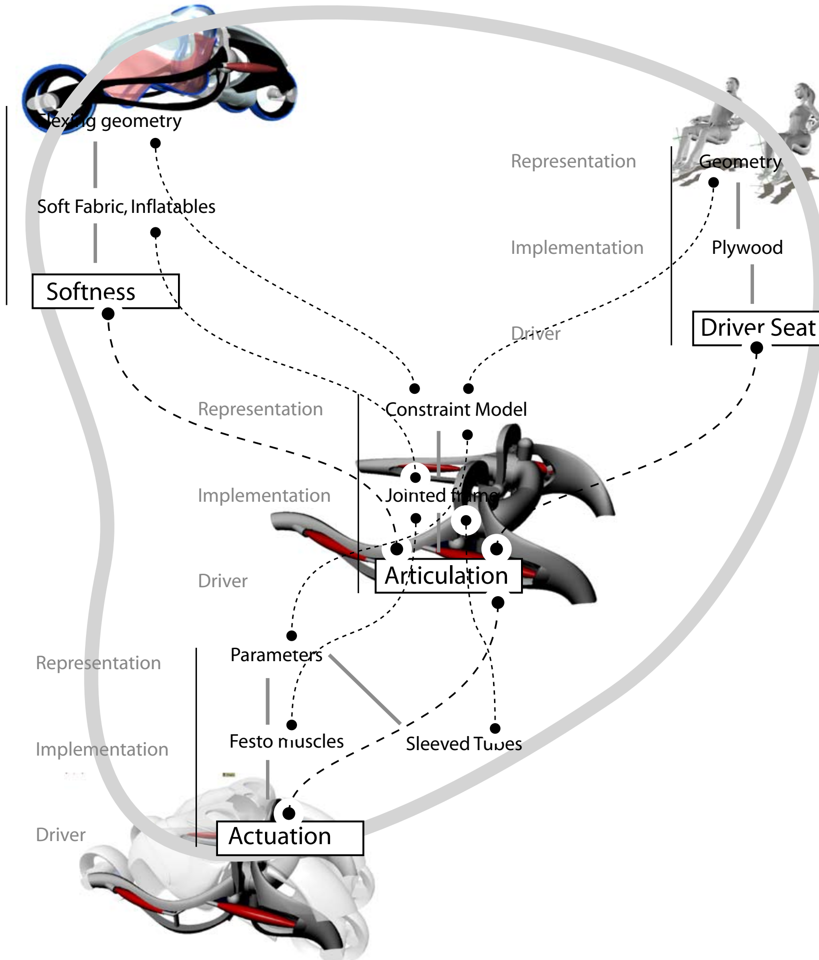










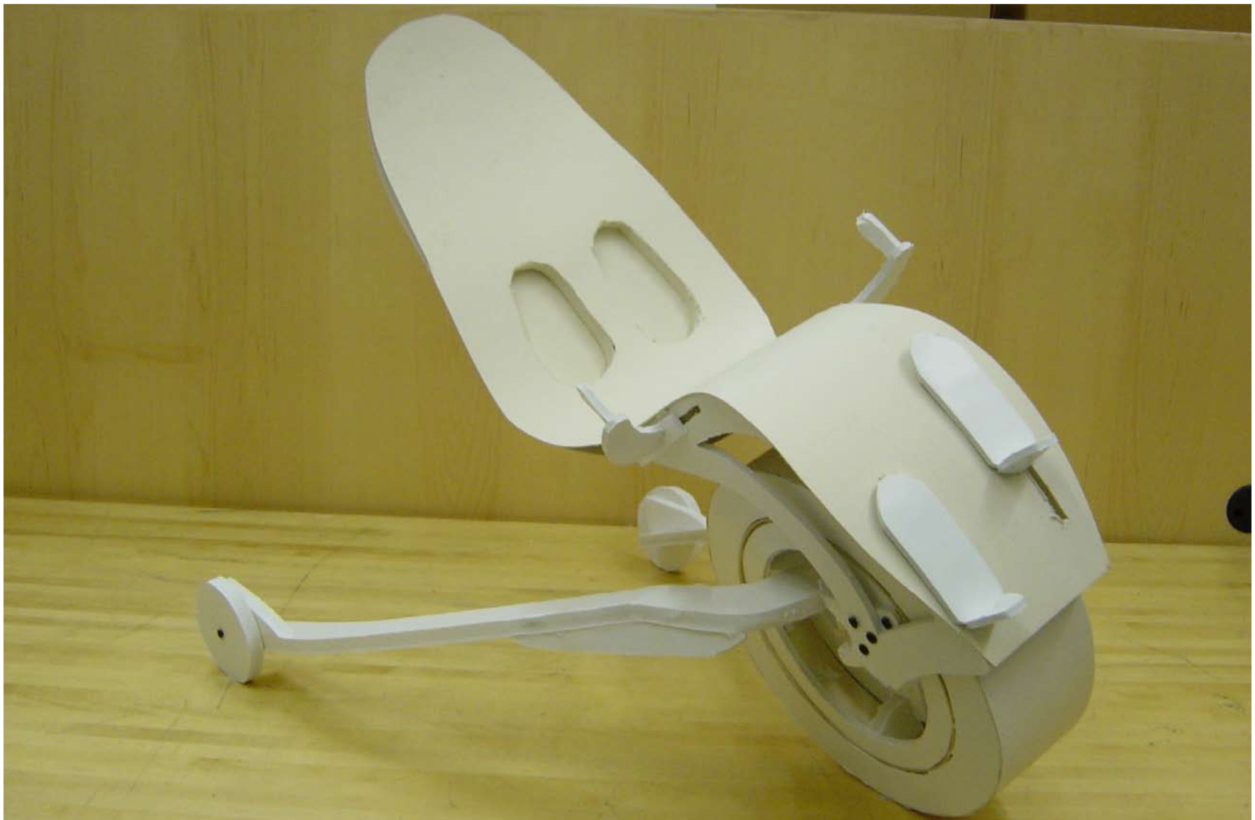


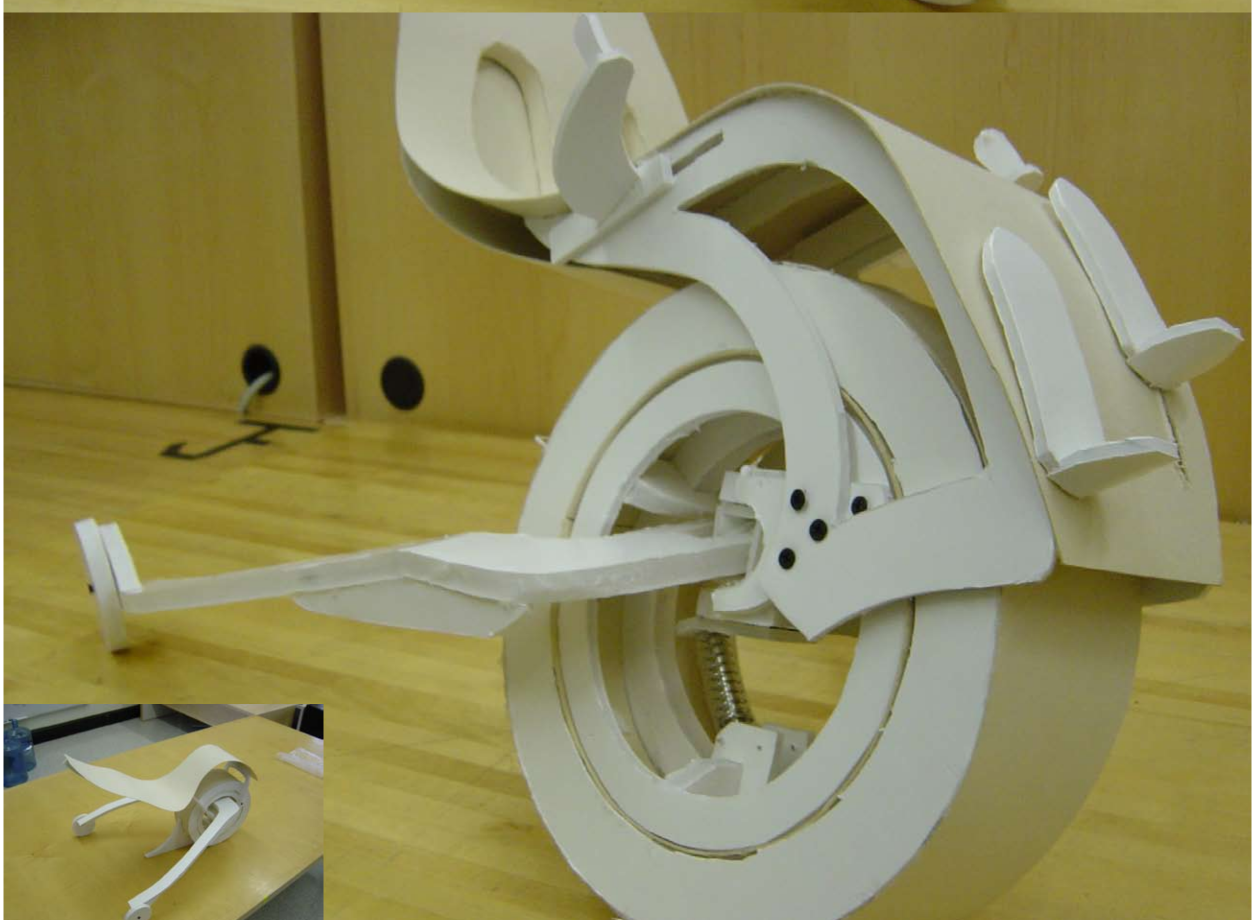
Another iteration, this time with more progress in the skeleton and articulation areas of the design.
In addition the movement constraint of the drivers is introduced from the previous selective prototype.
Chassis and muscle development by the author with Peter Schmitt.

vehicle. The animations was accomplished through animating the movement of the people's skeletons first and than subsequently build the vehicle around those motions in an extension of the body layer by layer, starting with the seat and ending with the wheels touching the ground. The study was done to test the timing of human movement as a trigger for steering the vehicle and to understand the potential of mapping gesture onto vehicle articulation.

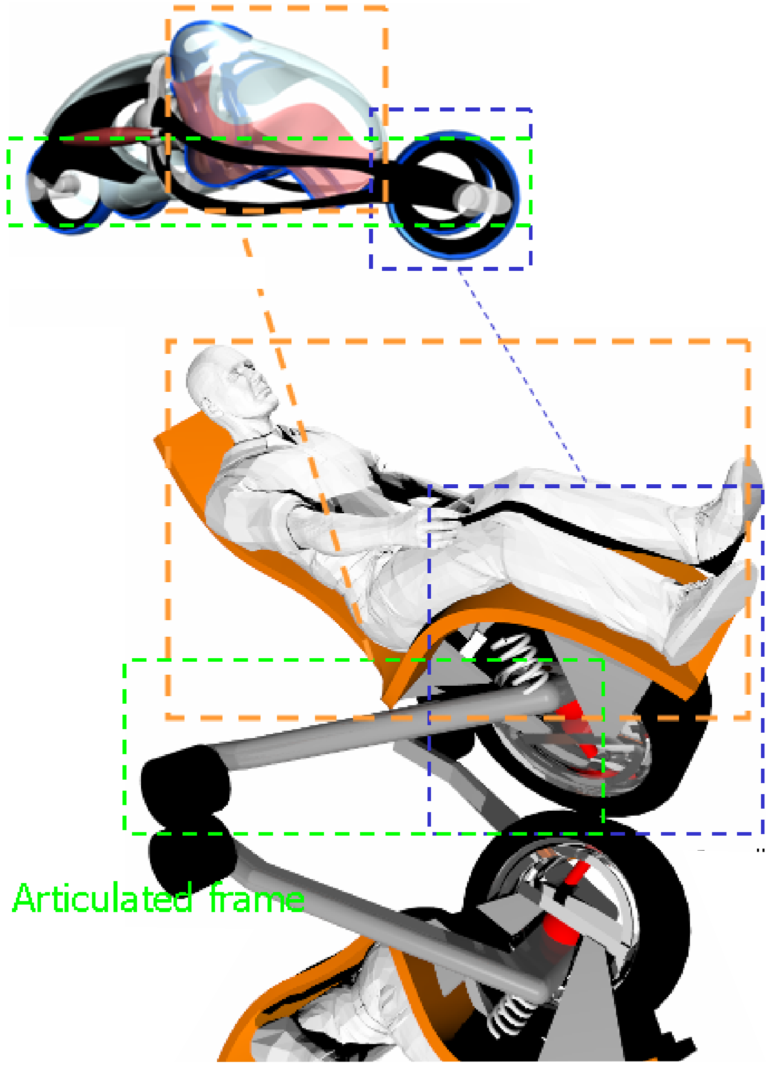
5.2.4.7 Aesthetic development and evaluation

A series of renderings and aesthetic explorations, based on the previous selective prototypes, pushed the holistic design idea of the Athlete car idea forward. The integration of several novel concepts





On the right: the relationship between original and reduced athlete.



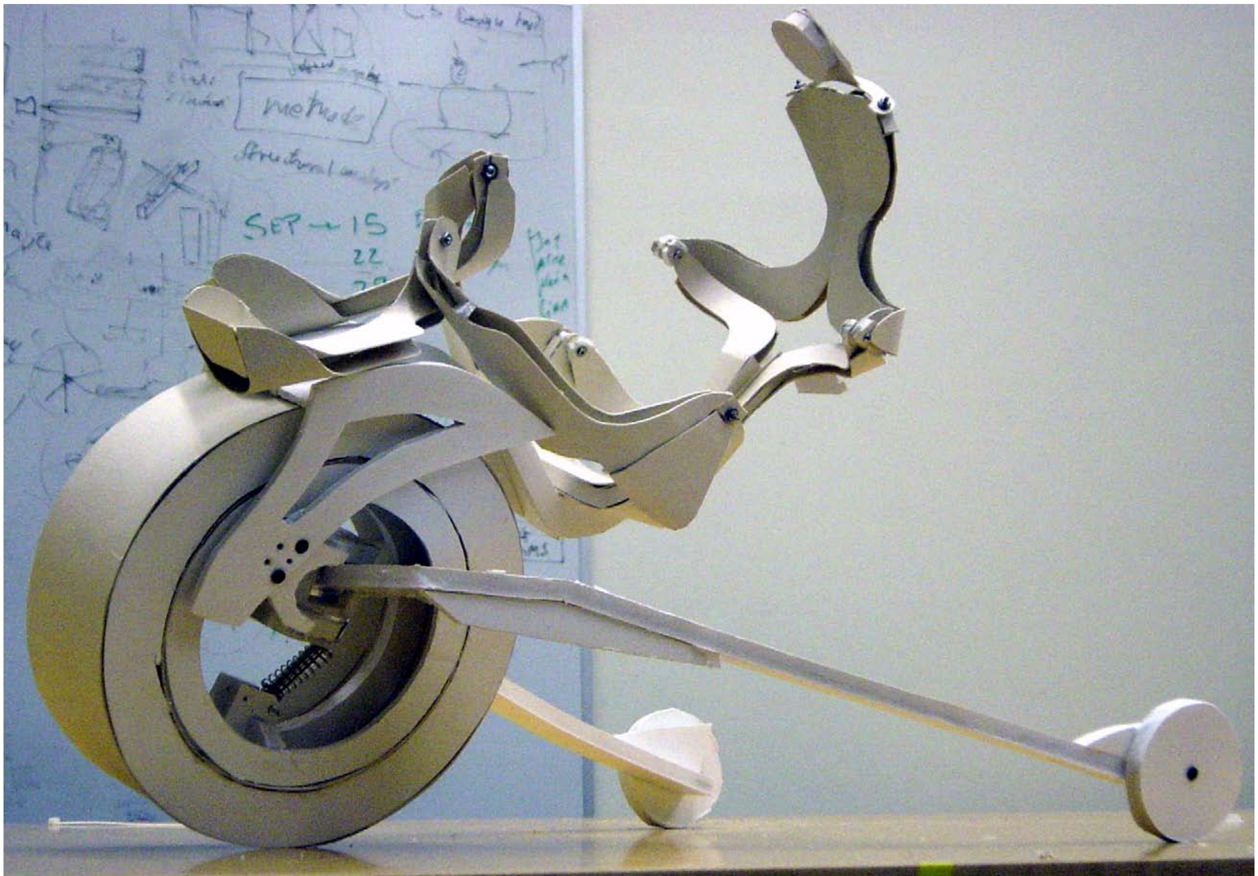
Due to the complexity of the athlete design the author decided to continue with another selective prototype design that reduces the design to one robotic wheel, one seat and a two degrees of freedom frame. The initial cardboard model sketch. The wheel articulates around a universal joint pivoting point centered in the hubless wheel.

like the frame, the door studies, and the skin required many design innovations at the level of details to resolve the concept as a whole. This was an ongoing process and did not end with the studies presented here. In addition, the design studies triggered a number of small changes in the underlying architecture.

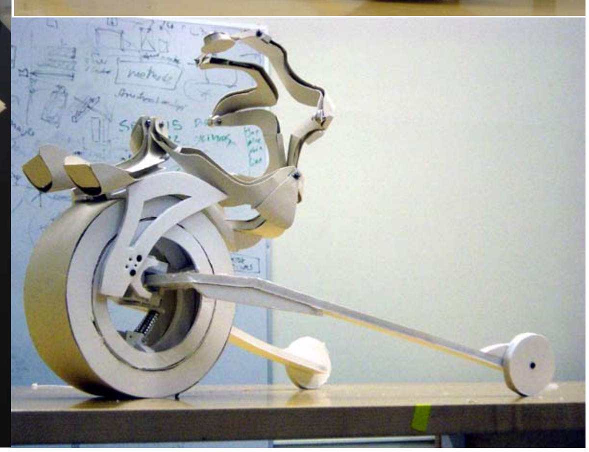
The design developed through a step by step expansion of design features and design constraints which articulated themselves through the translation of the design into different design representations. The adding of additional design branches slowly defined the design space of the athlete vehicle.

The design representations were:

- Physical mockups using foam core without digital input
- Digital geometry models for image rendering.

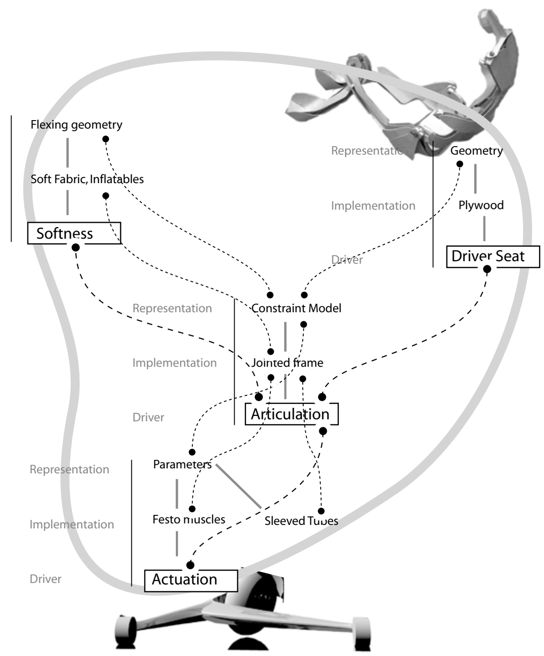



axel kilian july28 2005



- Servo actuated kinetic models
- Constraint solver model in a three dimensional assembly
- Inverse kinematics studies

Once the design had reached the level shown here the author decided to switch the scale and complexity level for further prototyping to first selectively prototype a small but complete functional entity of the athlete at full scale to test it. The following studies illustrate the scale model development.



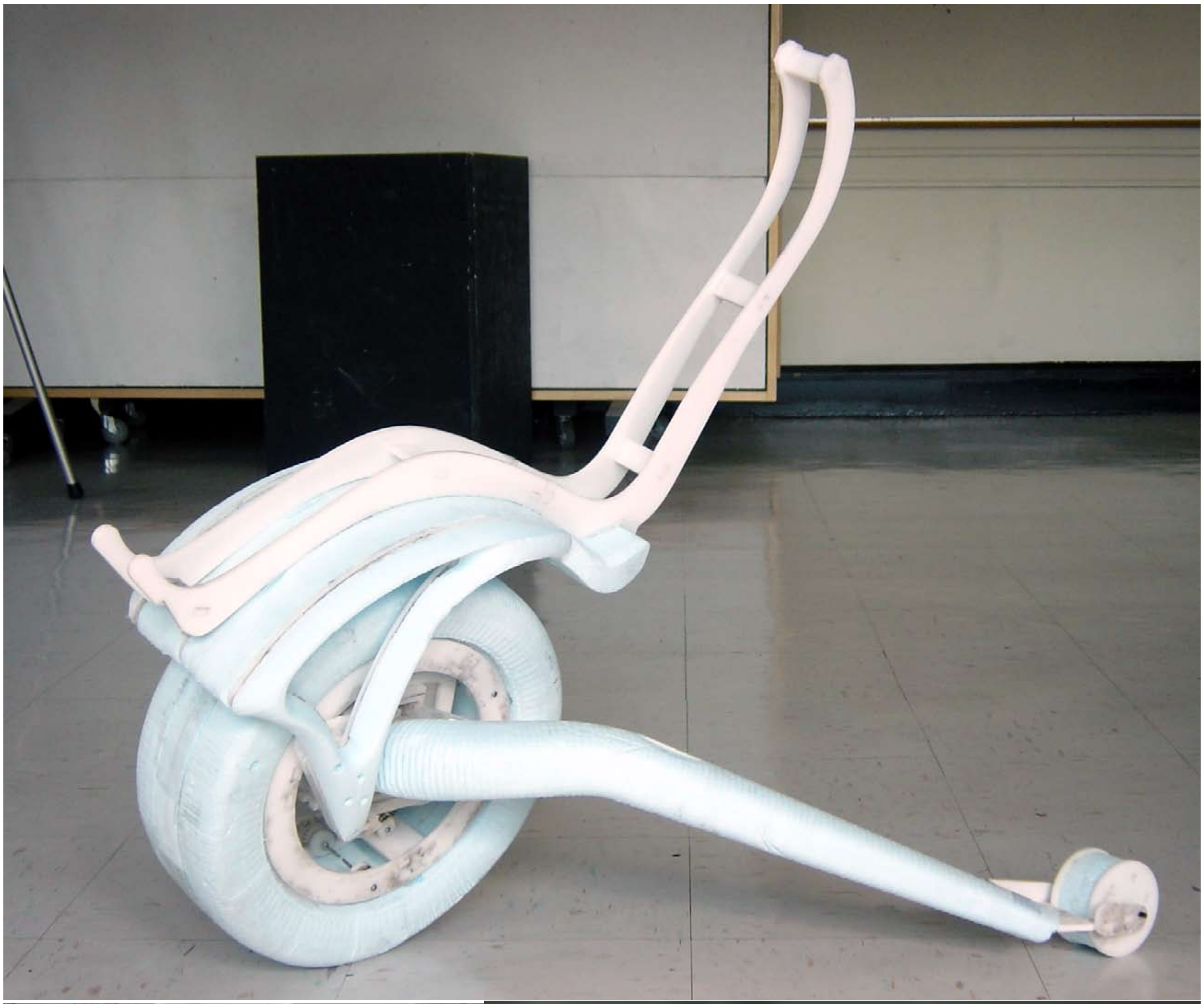
Further design iterations around the same constraints as the full athlete. Now the exploration is shifting more towards a circular exploration as all the constraints have been established and the revisions focus on improving the dependencies of the parts in detail.

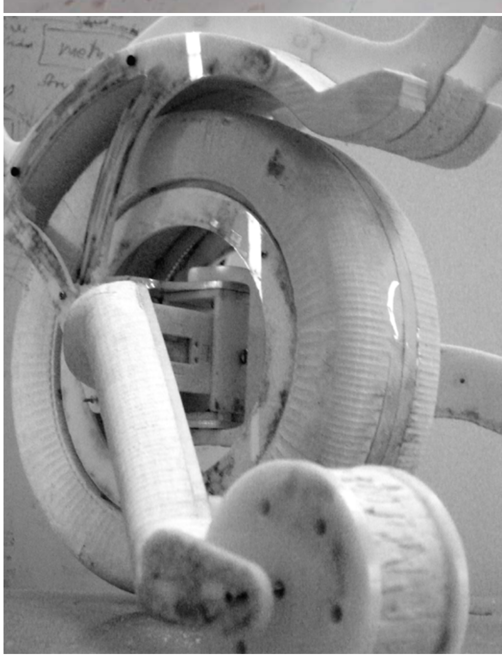
5.2.5 Implementation: The Mini-Athlete

It was too early to consider a prototype at full scale was too early for the novel h-series architecture and with scaling come too many mass- and force-related problems.

5.2.5.1 Selective prototype versioning to support decision making process – scale and complexity

At the end of the full athlete design process it was too early to consider a full scale prototype for the novel H-Series architecture, and scaling would have meant too many mass- and force-related problems. Therefore, the author went ahead with a series of partial prototypes, full-scale prototype designs, aimed at capturing the essential parts of the Athlete design and translating these into a





buildable, functional Mini Athlete. One of the inspirations for this design reduction was a three-wheeled toy with a big front wheel and back wheel steering. In its final rendition, the Athlete mini version was implemented as a group building project headed by the author, together with Peter Schmitt, Patrik Künzler, and Enrique L. Garcia and fabrication help by Phil Liang and Raul-David Poblano, and Brad Schiller.

5.2.5.2 The One Wheel Athlete development

The One Wheel Athlete contains all major components of the full scale prototype: the robotic wheel, the seat-based driving control, and the articulation of the frame using pneumatic muscles. The driver banks and steers with the robotic wheel. The back part of the frame acts as a stand-in for the remainder of the frame of the full scale Athlete.

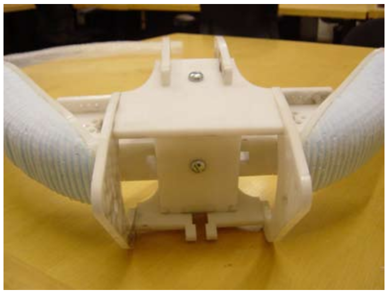
5.2.5.3 Selective prototyping

The selective prototyping continued at a faster and more limited scope for the Mini Athlete. Major progress was achieved through a series of prototypes of the universal joint connecting the initial double arm to the wheel robot. This was later turned into an asymmetric one-sided arm design for the sake of being able to use a hubbed wheel.

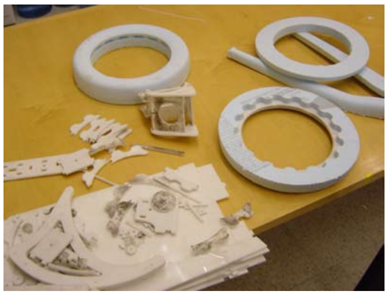
5.2.5.4 Scaled model design development

Following a rendering, an initial sketch model in cardboard was produced at a scale of 1:3. Based on the cardboard model, a number of design refinements were made to the seat and the universal joint connecting arm and hubless wheel. Those findings were then integrated into a CNC-fabricated model of the Mini Athlete at half scale, designed and built in about 3 weeks by the author. It involved a sandwich assembly of laser cut parts of Plexiglas© with milled foam form pieces glued together for stability.

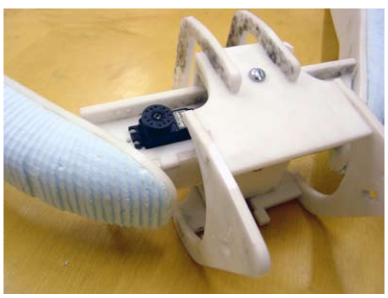
The externalization of the design idea in form of physical models helped to identify problems and conceptual shortcomings. The rapid revision cycles in physical form moved the design along much more effectively than purely digital modeling could have. The structural and balance feedback was crucial for developing the design.



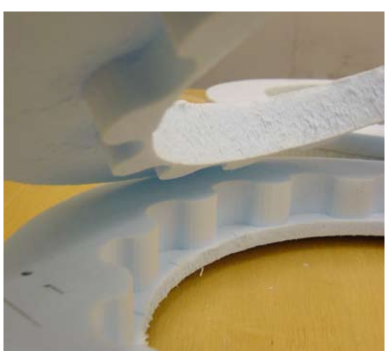
Symmetric joint assembly



Assembly overall



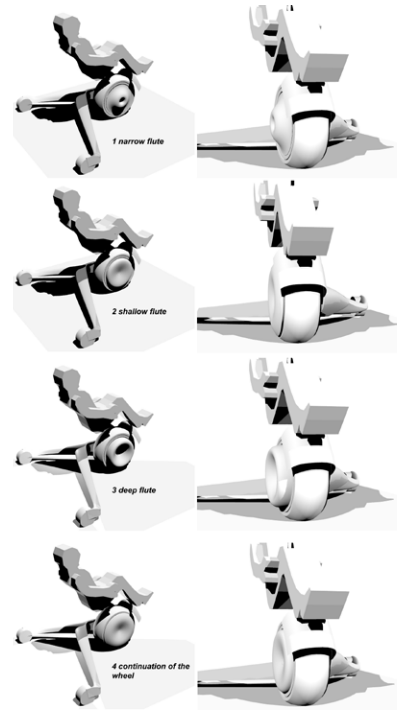
The last iteration of the universal connection joint



Connection detail between parts

The 1:2 scale digitally fabricated athlete study in foam and plexi by the author.

Final full scale design and built prototype:
Axel Kilian, Peter Schmitt, Patrik Künzler, Enrique L. Garcia.
Fabrication assistance:
Phil Liang, Raul-David Poblano, Brad Schiller.



Exploration of wheel designs.

Finished arm surface

Arm with channels for muscles

Carbon applied

Fabrication of the full scale design in carbon fiber based on molds in mdf and soft foam plugs. All molds and plugs were CNC milled.
Rim mold shown by Peter Schmitt.

5.2.5.5 Articulated arm

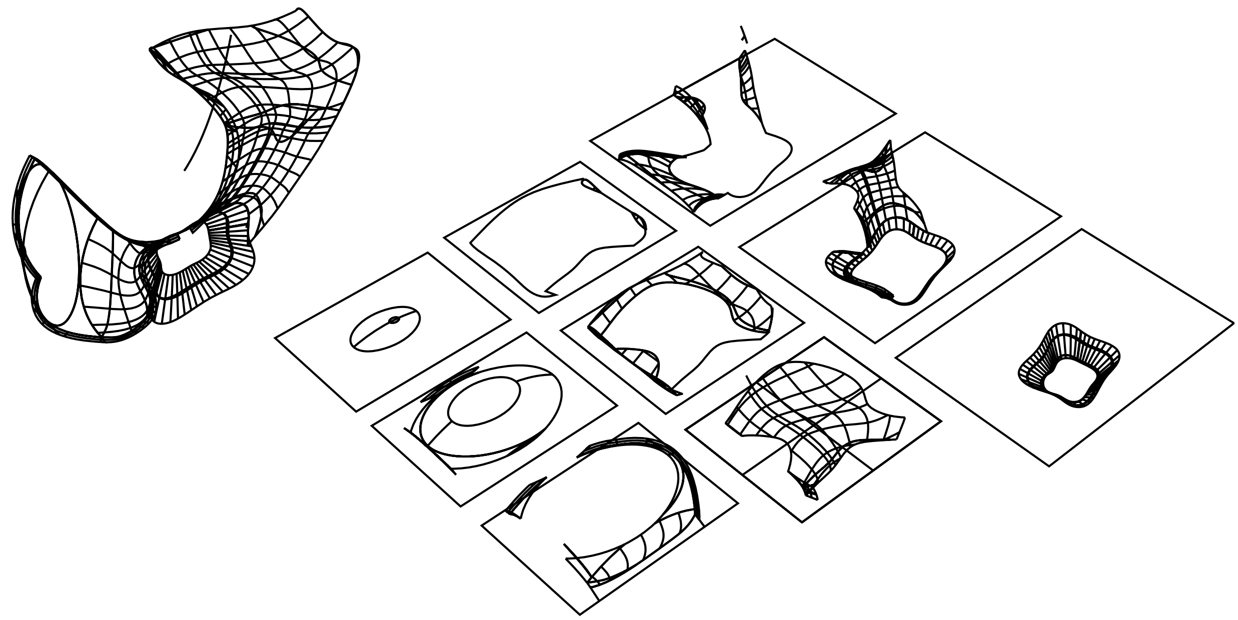
For the partial athlete one wheel prototype, the articulated skeleton was reduced from its eight degrees of freedom to the minimum of two degrees of freedom that still allow steering and banking of the robotic wheel. The arm in the full-scale prototype was mostly a test bed for carbon fiber molding and the combination of muscles with a universal joint. The muscles are positioned along side the arm and then guided through embedded channels to the four exit points which connect to the attachment points on the main support structure of the seat.

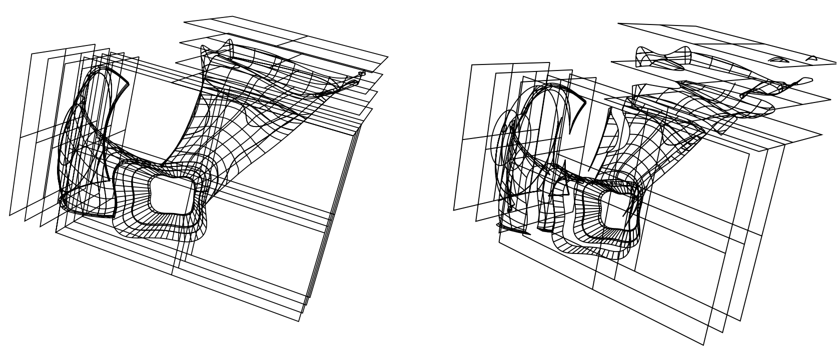
5.2.5.6 Pneumatic muscle-based actuation

Several muscle studies were built around the air-based pneumatic muscles similar to FESTO© air muscles. For the prototype, meshed cable sleeving was used in connection with airtight tubing and solenoid valves to produce an electronically controlled muscle. It was developed by Peter Schmitt of the Smart Cities Group. The air muscles are extra ordinarily fast in their response time and comparatively light compared to their piston counterparts. The main disadvantage is their relative flexibility and elasticity, which makes high precision positioning challenging. The mini Athlete will test the use of 3 feet long air-based muscles for the controlling of steering and banking.

5.2.5.7 Robotic wheel

At the core of the mini-Athlete is a version of the robotic wheel designed by Peter Schmitt. Earlier versions of the robotic wheel were developed by Patrik Künzler. The latest version in the mini-Athlete consists of an integrated suspension and power train using a hubbed wheel design to reduce the cost for large bearings needed for earlier hubless designs. The robotic wheel has been used in many of the other design studies developed besides the Athlete. In this case, it is combined with the articulated frame universal joint to allow for banking and steering of the wheel. The choice for the tire reflects the need for the tire to be able to bank into corners.
The potential of the robotic wheel goes far beyond this prototype study. The study works well in showcasing its possibilities in a minimal package.





The fabrication of a larger part within the constraints of a limited depth mill. Several translations of the geometry are necessary to prepare it for fabrication.

5.2.5.8 The human seat

The seat was developed by Patrik Künzler, Enrique Garcia and Axel Kilian as a lightweight bent wood exoskeleton-type seating support. It supports the driver's weight in a suspended fashion while allowing for the movement of the legs and the back within the range of motion designed into the joints. After initial design studies of the joints as hinged connections, the choice was made to use flexures instead, as they provide the combination of hinging and force resistance, as well as variability in the degrees of freedom allowed by the connection. This is important as the human joint does not follow simple mechanical joint patterns. The knee joint, for instance, follows multi-pivoting point schemes. To work against those joints is uncomfortable to the wearer and may cause long

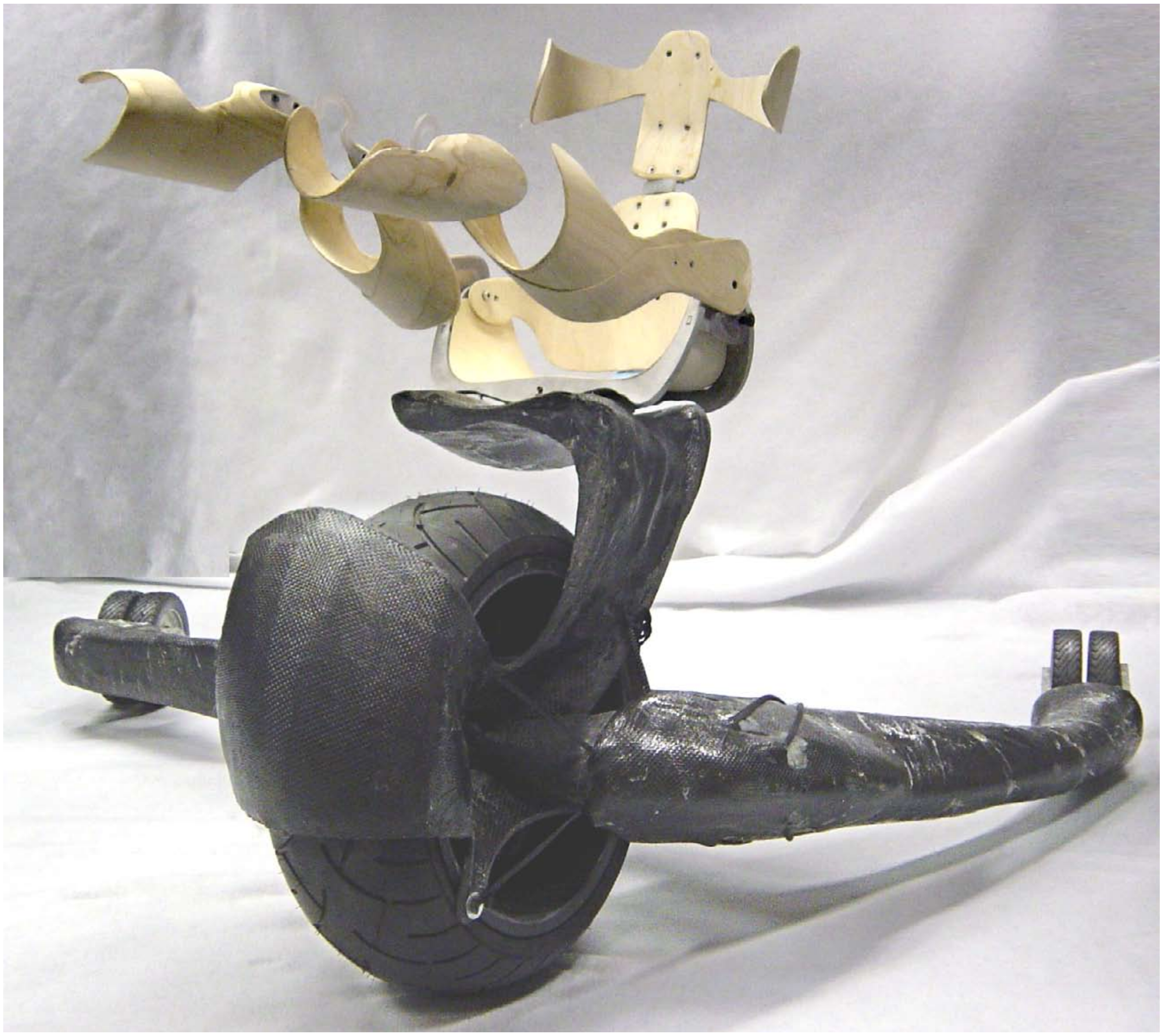
The molding process of the seat support showing the combination three dimensional printing for the interior wheel hub (design Peter Schmitt) and the exterior seat support structure in foam (design and implementation Axel Kilian with Enrique L. Garcia and Peter Schmitt).

term damage to the joints.

The main finding in the chair design was how to construct a supportive structure that was chair like but driven by the unique requirements of supporting the bodies weight while allowing free movement of legs and back.

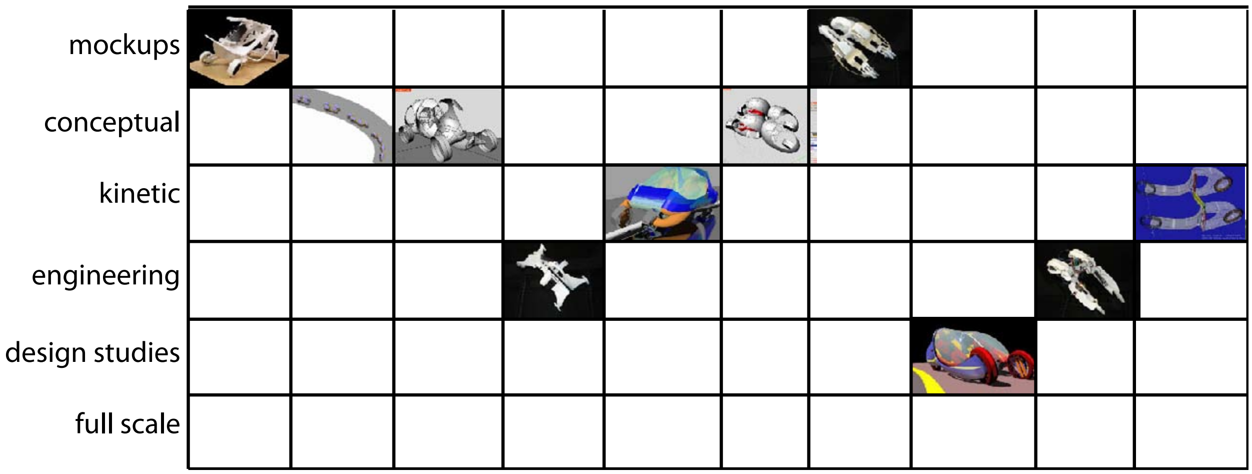
5.2.6 Conclusion

The type of exploration in the experiments overall is referred to as branching. Branching stands for the development of a design space from the linking of design constraints into a design explorer. The experiments developed several exploration techniques for establishing a solution space or to expand it for novel design problems in the vehicle design domain. They were:



Finalized model. Carbon frame and wheel, plywood seat, muscle tendons piercing through the arm to control the wheel. Design overall: Axel Kilian. Robotic wheel and arm: Peter Schmitt, Axel Kilian Humane Seat: Patrik Künzler, Enrique L. Garcia, Axel Kilian.

The author riding the wheel



Matrix of the different design explorations by the author over the course of the project of approximately one year..

Functional chains

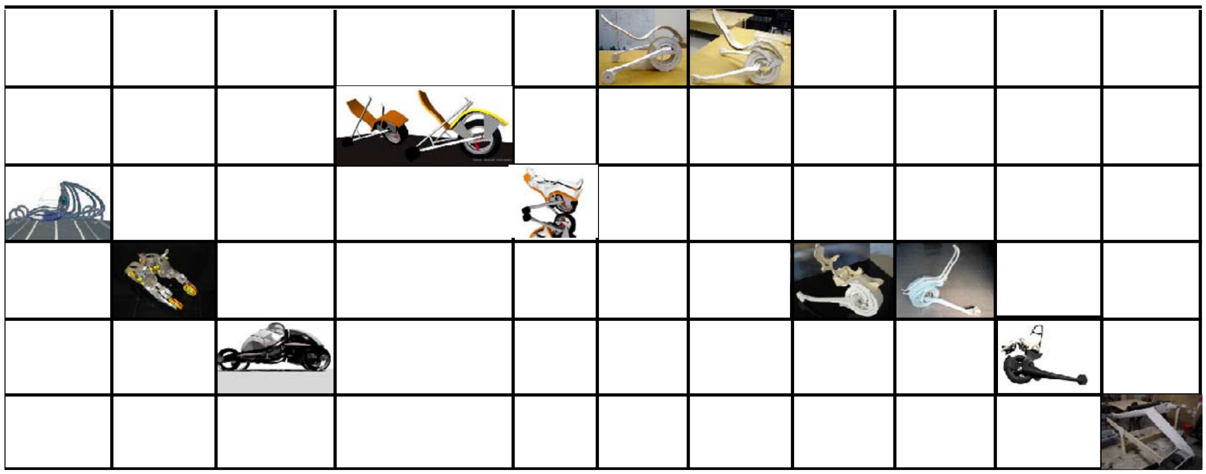
The formulation of a design goal from the chaining of functional demands extracted functional design triggers from a set of existing designs. The method proved to be a high level light weight design description at the diagrammatic stage of design development. As the complexity rises of the mapped functions rise it becomes increasingly challenging to identify general component free function relationships. As a visual diagram study to sort and review the analysis of complex function dependencies it is a very powerful technique.

Design exploration based on selective prototyping.

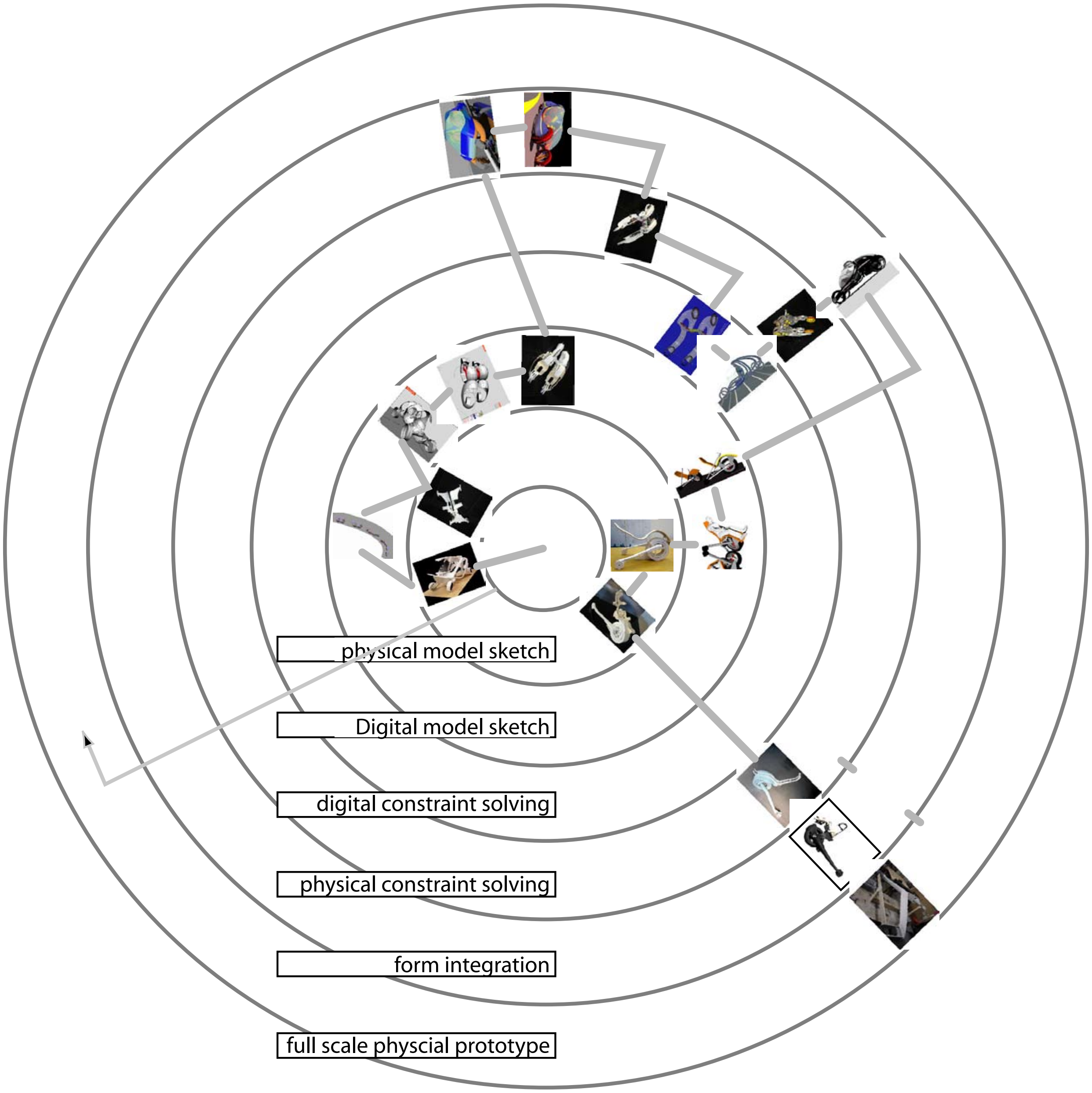
This process helps to iteratively develop subcomponents of a novel idea and feed them back into a larger evolving concept form the bottom up. Novel architectures can emerge without having to break away from conventions but by reframing the problem descriptions.

Translations between design representations

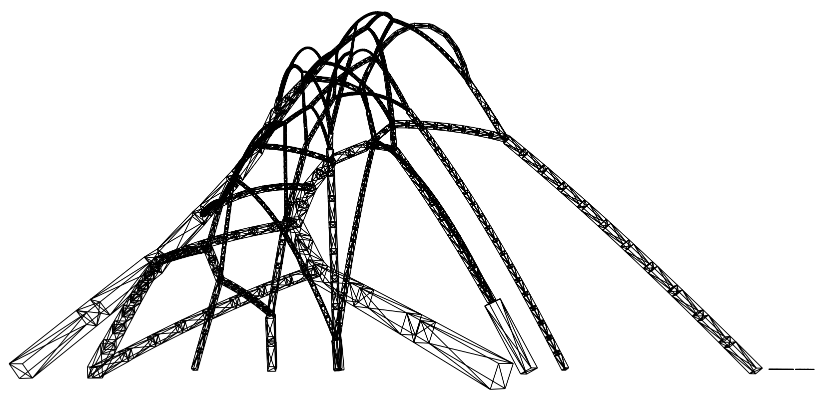
The athlete study was mainly designed through translation between physical, digital and kinetic mockups that informed the



design variations. The translation from one representation into another is never a complete mapping. The gaps in the translation trigger a partial redesign based on the essential properties of the design. The repeated redesign forces the constant reassessment of the design based on what is continuous across the design representation and what is a medium specific artifact. Scale plays another important role in the translation as does force. The articulated servo based studies would not look like they do even using the same component at different scale as forces and motor



Radial version of the design steps.



A digital hanging model. An example of a parallel exploration of constraints.

torques do not scale linearly with the sizes of components. Similar scale dependent factors play into proportions of designs and detail development of connections.

Overall the design exploration of the athlete can be divided into two parts. One part was the establishing of the design concept through a series of selective prototypes that established the constraints for the design problem. And in the second part, once the design constraints were established an exploration of refinement was done for the one wheel reduced athlete concept. Here the translation between design representations mostly served the problem solving of the established design idea unlike in the first study where it had to be created first. In terms of the thesis types of exploration it is a case where a branching exploration segway into a circular exploration.

The combination of these three methods illustrated in the concept car experiment forms the basis for the concept of branching explorations, a way to create a design problem from the linking of design constraints and representations.

5.3 Parallel Exploration: Exercising the Constraints

In his paper "Intelligence Without Representation", Brooks lays out the state of the field of artificial intelligence (AI) in the late 1980s. He addresses how to define an appropriate AI problem, if any problem that can be solved by an algorithm is not deemed a real AI problem by other fields. (Brooks 1991).

The field of computation and design faces a similar dilemma. There is a lot of overlap between the challenges of artificial intelligence and reasoning in design. In fact, the two rely on each other. Computational design relies on computer-human interaction for steering and implementing design generation, despite attempts to define autonomous design machines (March and Stiny 1981)

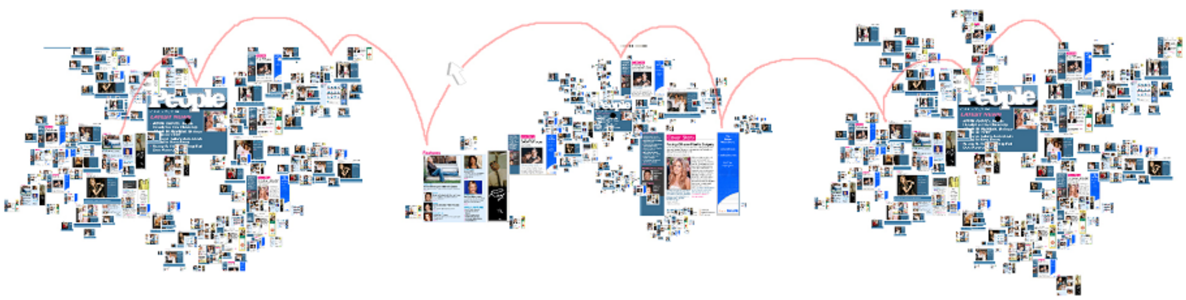
Design is a process that balances an arbitrary number of factors and constraints from the environment to make decisions about how to change things in it. With increasing sophistication of the design process, it does so with more and more abstract constructs that act as intermediate representations between the idea of change and the actual act of changing. These layers introduce the possibility of playing out scenarios without actually implementing them, essentially the concept of design exploration. The challenge of design exploration is to abstract and define the constraints from the real world into an exploration construct as if one were to create a game like chess. In AI, games like chess were initially considered the hard problems to solve in comparison to everyday problems like locating an object in the world. But it turned out creating an understanding of the world and what surrounds us is far more challenging than playing a game within a defined rule set (Brooks 1991). In addition in design the rules may change constantly even during the game. One can argue that design through a design explorer is a two stage process. First, creating an understanding of the world and the rules that play into the design and turning them into an exploration "game"; second, exercising this game according to the rule and constraints. In the exploration process one follows all the constraints or rules in parallel and the design evolves from the intersection of design input and system-game response.

The comparison to a complex game may only be applicable in a well-understood and well-defined exploration, where design becomes playing a game of exploration within a set of defined constraints and rules.

The main example outlined in this experiment section, the hanging model, comes closest to this scenario. Based on the introductory reference to Brooks, one could ask whether this design process is still design since it can be implemented through an algorithm. In balancing between interdependent constraints, as is the case in the hanging model example, the degrees of freedom allow for design exploration: the emergent states are unpredictable and unique in their response to the constraints. However, it is probably appropriate to include the definition and implementation of the constraint explorer itself as an integral part of the process, if one is to refer to it as design.

What follows is the sub chapter of the third type of exploration in this thesis. So far we have seen in this thesis the circular exploration that refines the constraint dependencies in a problem, the branching exploration that helps to establish the constraints for a design problem. This section illustrates the third type, the parallel exploration that exercises the design problem. The experiments are based on a final project in Rodney Brook's and Una-May O'Reilly's course embodied intelligence, and the hanging model was initially created together with Megan Galbraighth and Dan Chak for a final project in Computer Graphics course taught by Seth Teller and Fredo Durand. It later was further developed for two workshops on digital hanging models initiated by John Ochsendorf and co-taught by the author, Barbara Cutler, Eric Demaine, Marty Demaine and Simon Greenwold in 2004 at MIT.

The parallel exploration experiments test the parallel integration of multiple constraints through Genetic Algorithms (GA), parametric geometry and finally through the use of Particle Spring Models in the form finding example. The GA is used for exploring high-rise structures based on building volume and for developable strips on double curved surfaces. Although the use of a fitness function



does not technically constitute a constraint, the selection process eventually enforces the constraint indirectly.

The particle spring model uses a solver architecture to process multiple constraints in parallel. The technical models of GA and solvers are not limited to the example shown but can also be used in different types of experiments. The modularity of these computational models is an important aspect for the implementation of design specific design explorers.

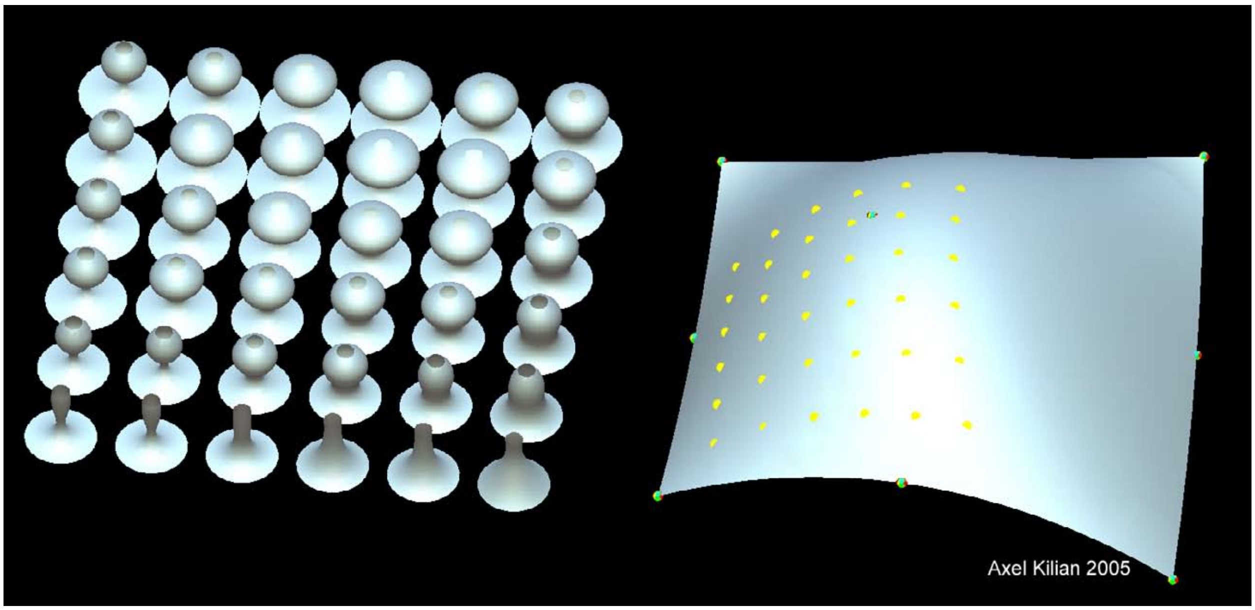
An attention based exploration device for visual data is an example of a dynamic design explorer. It keeps track of the history and displays it simultaneously through attention based scaling. Example implemented in Java, originally for the author's SMArchS thesis in 2000..

5.3.1 Search Strategies for satisfying Multiple Constraints in Parallel

The question of how to navigate through the solutions in a given design space is a major issue. The fact that constraints are present does not provide definitive answers in terms of design solutions. Any design proposal that satisfies the given constraints may still not fit the design intentions on other accounts not included in the initial description of the exploration. This is very common in design and in fact crucial to evolving the design target through the investments made in the exploration process.

5.3.1.1 Experiment 1: Visual tracks of design exploration

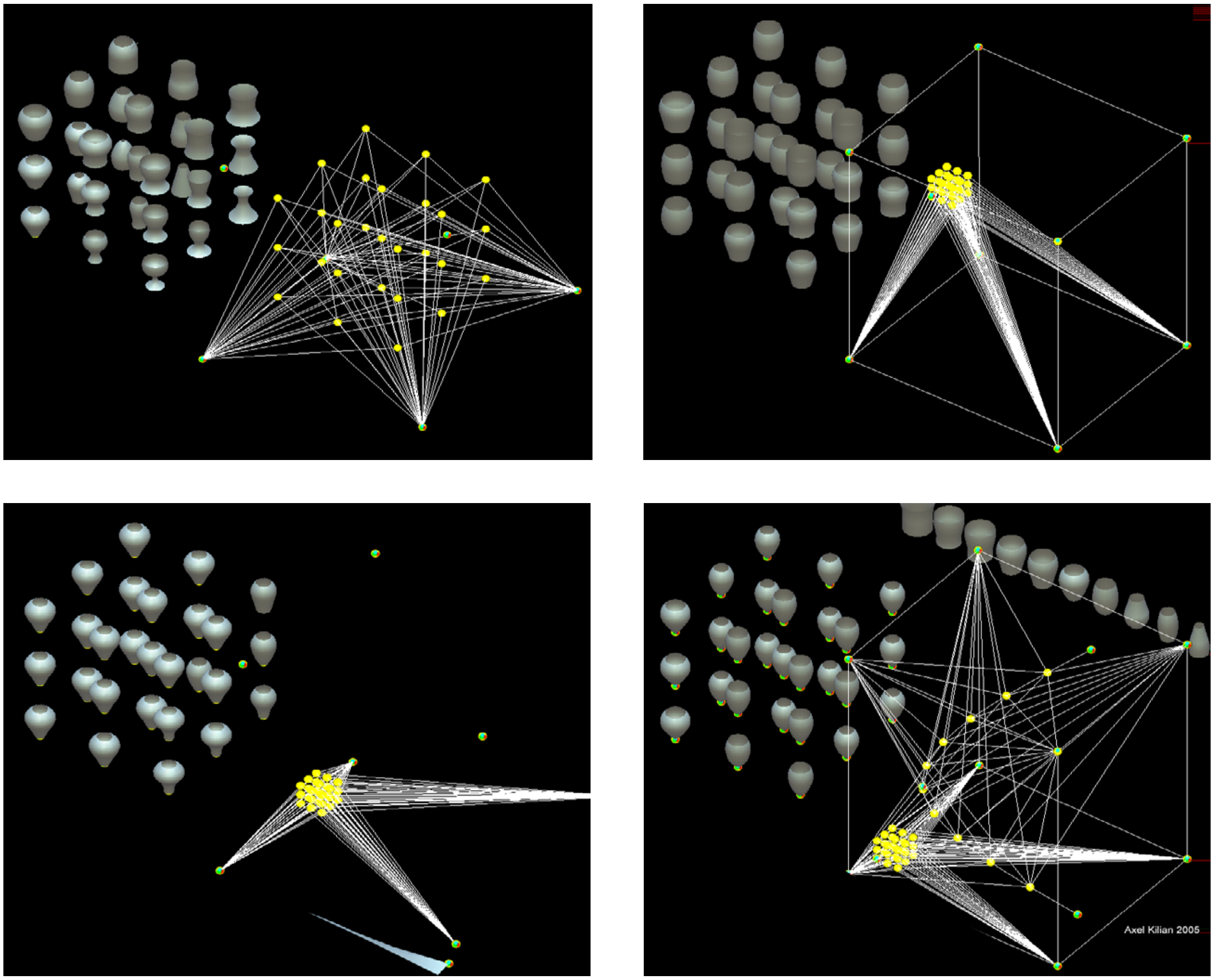
A first exploration into the tracking of design history was the



Storing parametric states of models in secondary control objects. In this case the three parameters for each instance of the parametric object are derived from the xyz coordinates of one of the surface based points. If the surface is changed all designs will update according to the new values.

multi-branched browser history experiment shown here. Nodes are created on demand as the exploration expands in a chosen direction. The amount of time spent on a node is recorded as a weight for the scaling factor in displaying the nodes. The entire history of exploration is visible at all times as a tree, with each node visually scaled according to the time spent on it This process allows one to quickly to locate the nodes that were visited more often. In the example, the nodes are screen shots of design variations of the chair experiment. The approach of tracing steps of an exploration dynamically including different decision branches promises a comprehensive overview of the design history and better support for exploration. The example shown here is an interactive sketch of a possible implementation programmed in Java.

5.3.1.2 Experiment 2: Recording parametric states through

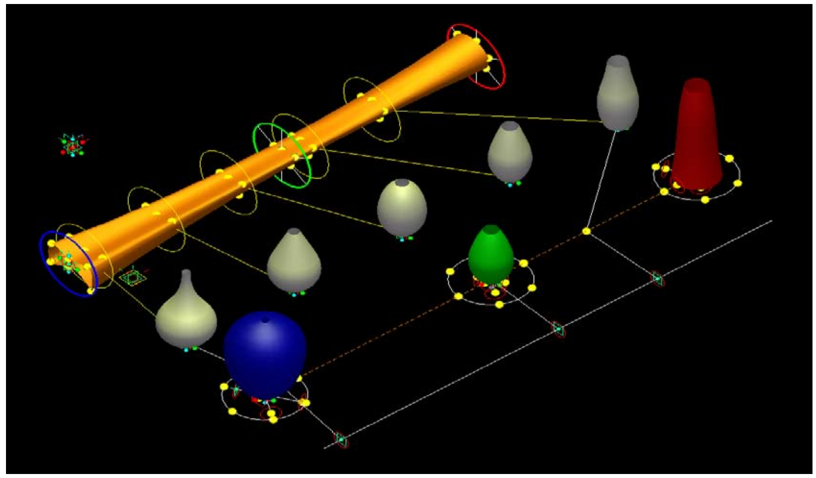


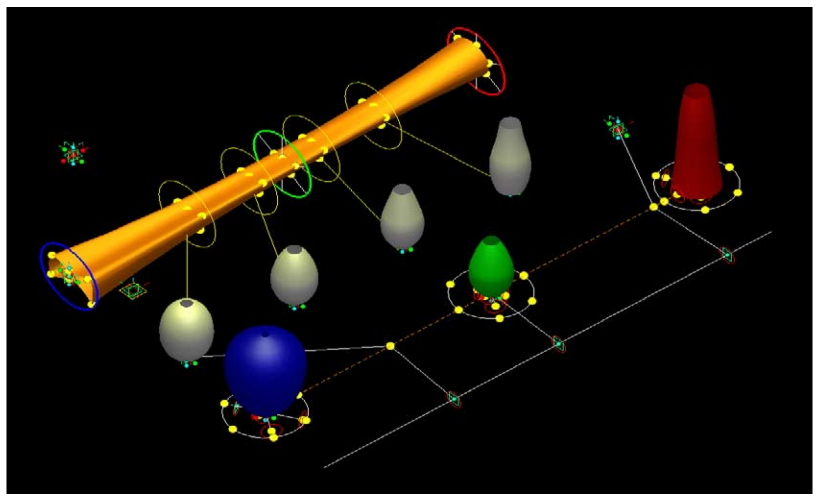
A three dimensional version of tracking multi dimensional parametric settings in visually memorable ways. the broad sampling of the first image gives a first impression of the range of possible design. The other images show more focused sampling of one of the designs. Trails of favorable designs can be laid down through the solution space.

UV coordinates on a surface

Designing with parametric objects poses the challenge of evaluating the design range of possible outcomes presented by a parametric construct. Higher numbers of parameters make it less intuitive to interact with the design construct. Parametric settings that are satisfactory are easily lost in manual manipulation. Design tables only offer the recording of series of instances of the state of a model but not a continuous interpolation between values. Often the interpolation between known states reveals the best designs as compromises between several optimal settings, especially with

Another variant of a secondary geometry object to store and manipulate instances of a design set. In this case three designs act as the input for an interpolation and selective sampling of the design ranges between the starting designs.





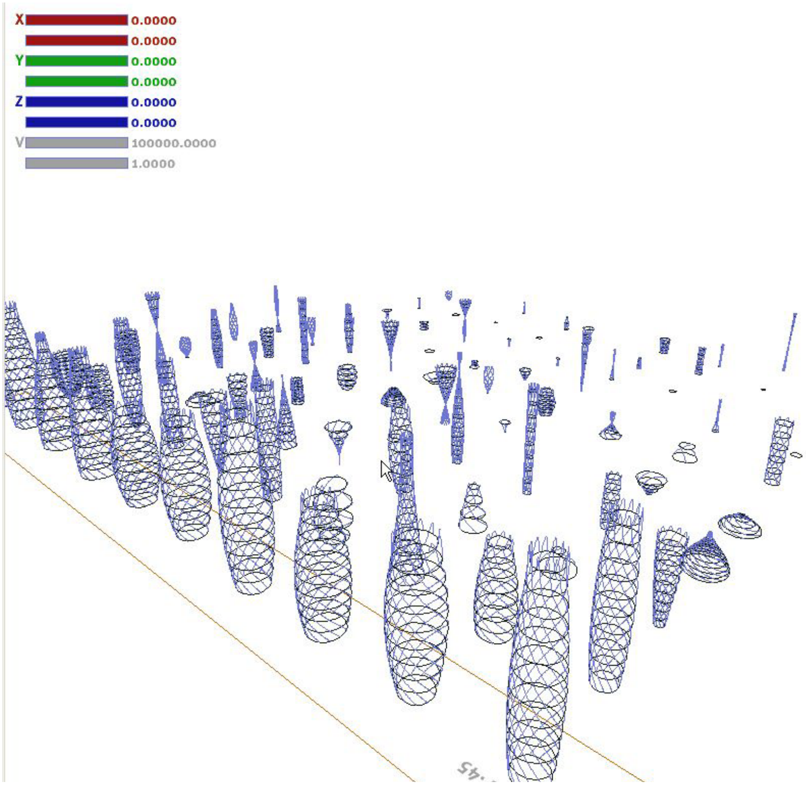
multiple constraints in parallel.

The example shown here demonstrates a geometric control object that provides a way to interpolate numerical settings for parametric objects and the possibility to record and memorize states of the parametric settings through a secondary geometry.

The first example shows a family of objects whose parameters are mapped to a grid of points that sample a surface based on regular UV spacing. Moving the UV grid adjusts the parameters and regenerates the object family. By increasing or decreasing the sampling rate around the points of interest, one can explore parametric variations in more detail where needed.

5.3.1.3 Experiment 3: Recording parameter settings in solution space

A variation of the surface-based recording is to place points in



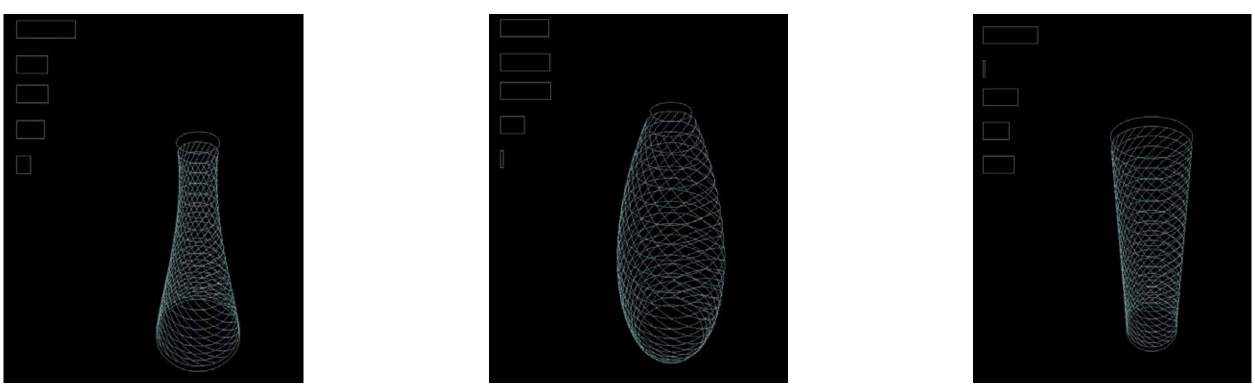
A simplistic example of alternative mappings of the genotype into a phenotype The information of the genotype is used to drive six parameters of the phenotype - in this case a function generating a tower. The towers are ranked based on their overall volume. The range of possible parameters is limited. On the left a snapshot of the towers as they are being generated and ranked sequentially according to volume. The top 10% of the genotypes are reinserted into the next generation.

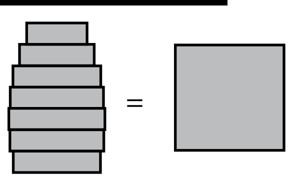
genome | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 0 | ....

genotype | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 0 |

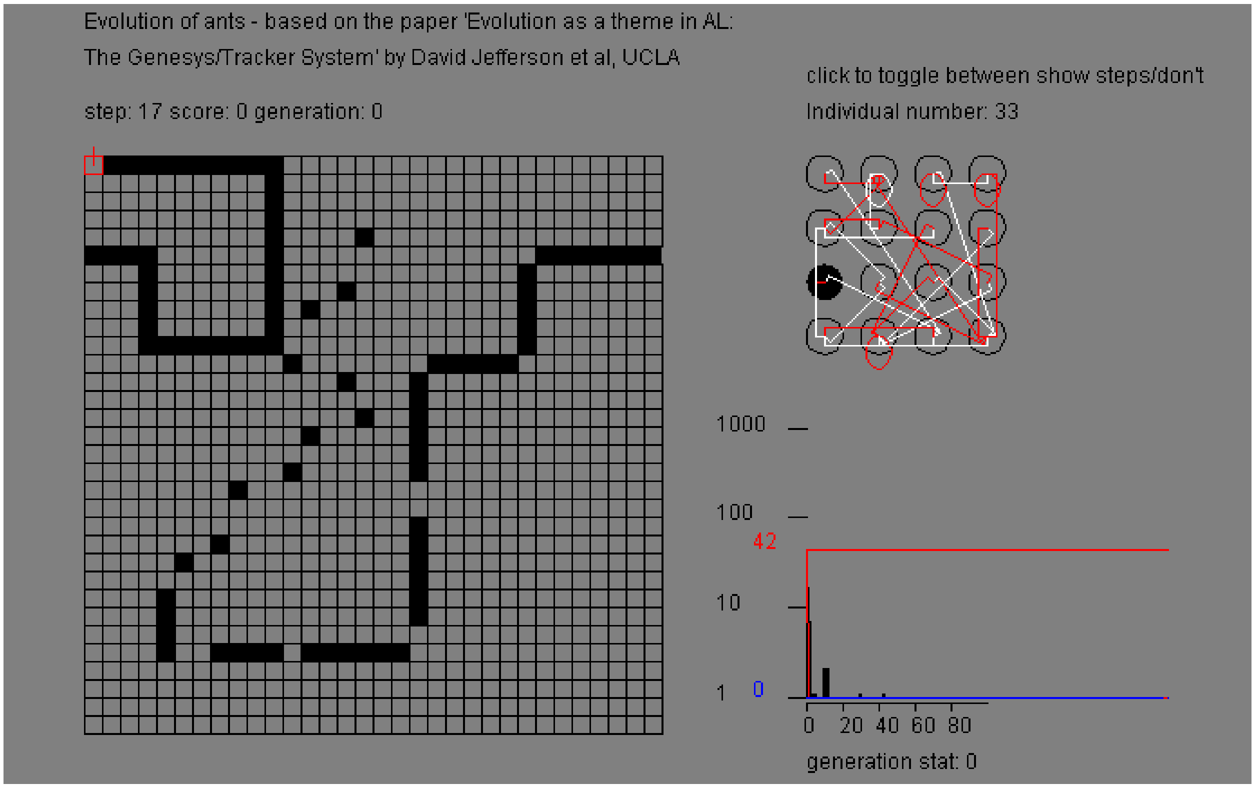
Interpreted as instructions to generate the phenotype through six parameters

3,3,2,2,2





Total building volume as fitness criteria



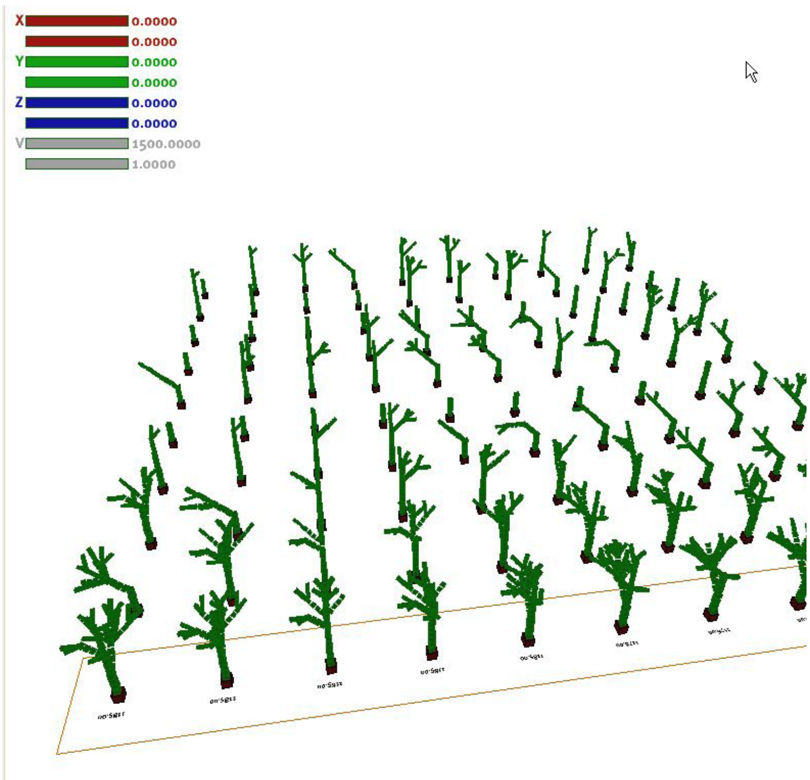
Java based implementation by the author of the original genesys/tracker by Jefferson et al. It shows the current "ant" interacting with its environment, the finite state machine belonging to the current ant and the overall statistics of the current run. The black trail is the "sugar trail" the population of ants is evolving to follow.

space and use those point coordinates to record parameter values of the parametric design object. Spatial point grids in a solution volume can give a good sample of the range of solution in that value range. Focusing and widening the range of the samples can provide a more detailed sampling of any sub-area of the solution space.

5.3.1.4 Experiment 4: Mapping parameters onto secondary control objects for higher parameter counts

A variation of surface-based recording of parameters is the use of spatial objects to track and drive parameters of higher dimensional parametric objects. In this example a vase with six parameters is

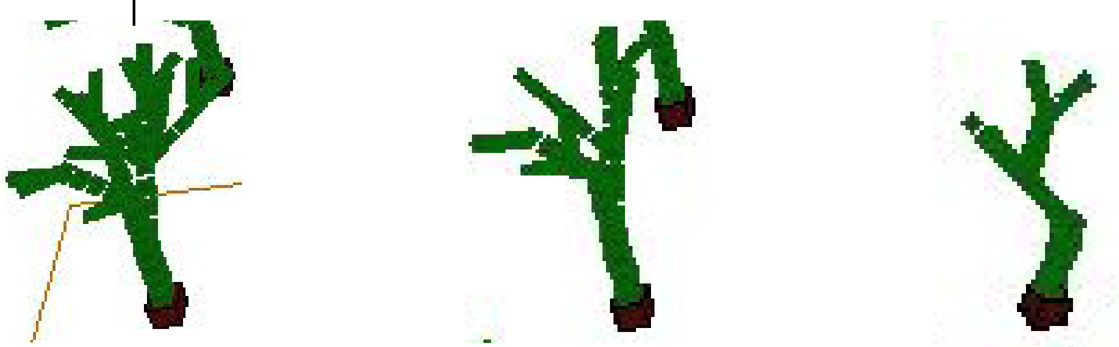
A simplistic example of alternative mappings of the genotype into a phenotype. Use of genome information to determine branching frequency.

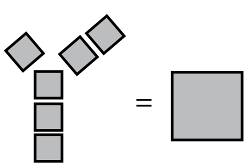


genome 1 1 1 1 1 0 1 1 ....

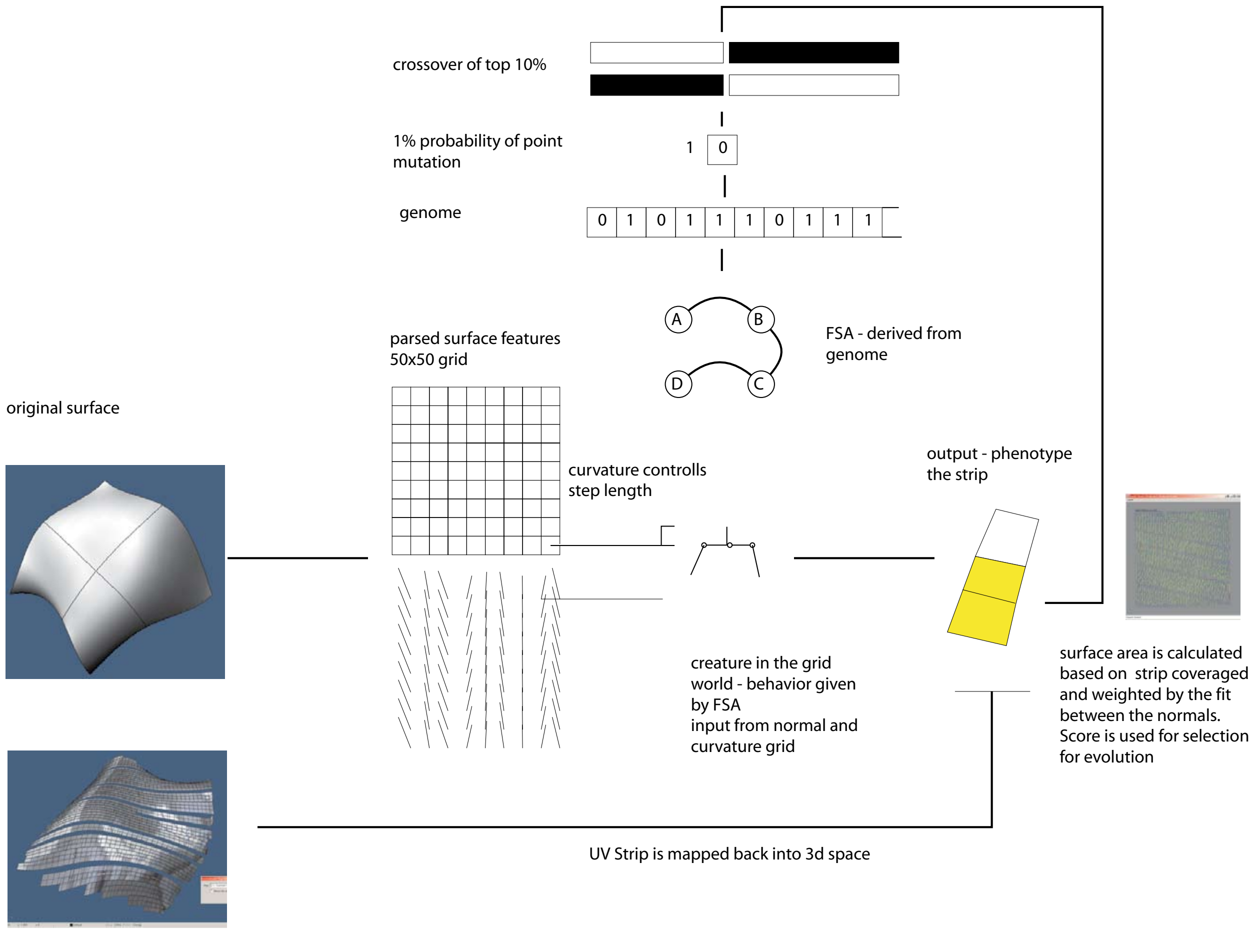
genotype 1 1 1 1 1 0

Interpreted as instructions to generate the phenotype following a simple growth algorithm. 3,3,2





Total branch volume as fitness criteria

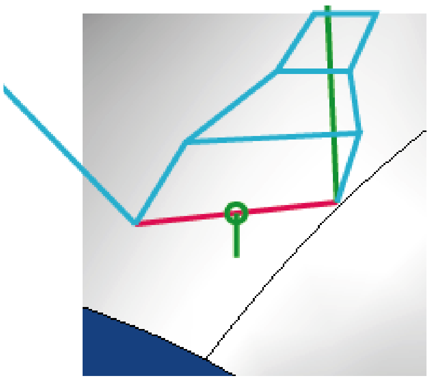


The overall GA system for searching developable strips in double curvature geometry.

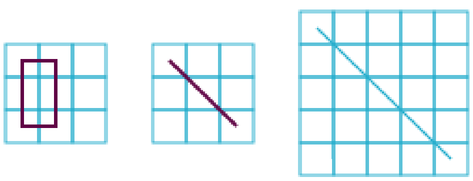
used. There are three instances of the vase to act as design drivers. Then there is a variable number of sample designs that are the sample parametric objects but with parametric settings based on interpolations between the three driver designs. The sampling density and range can be controlled as well. This is accomplished by constructing a control surface from the three driving designs through an interpolated surfaces passing through the radially offset points. The sample parameters are obtained by intersecting the control surface at the sampling intervals.

5.3.2 Genetic Algorithm Approach

This section addresses the use of genetic algorithms for searching a design solution within the framework of a parametric model. It is a good example of a modular design explorer where a search engine, the GA, can be combined with different entities it drives. One is the interpretation of the GA's genome as input for a parametric model, another is the definition of a finite state automaton (FSA) that is used to steer a strip on a double curved surface. The interface between the driver (GA) and the driven (the models) allows for a wide range of mappings making the GA method an ideal candidate for modular



Strip of variable width. The surface normals are sampled at the tip of the strip. The degree of how well the tip of the strip conforms to the developability constraint determines the behavior of the strip creator. In addition this value serves as a measure to rank the success of the strip in producing a



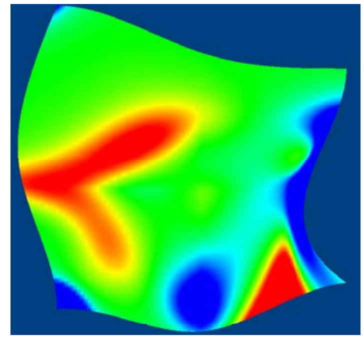
use in design explorers.

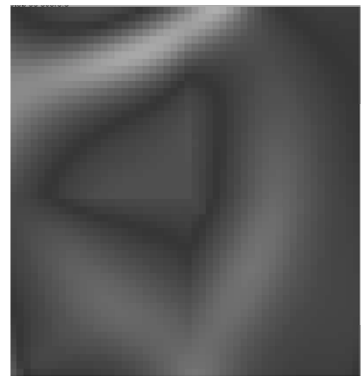
5.3.2.1 Experiment 5: GA based search for developable strips

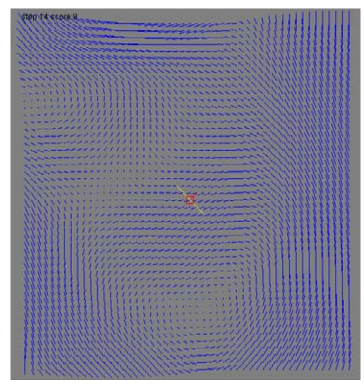
Rather than modeling a geometry directly, alternative models can be developed that construct geometry following a set of given constraints. Many such models exist based on rules or procedural methods for the task of determining developable strips on double curved surfaces.

The approach is based on a finite state automaton which is evolved to construct strips on the target surface. Based on the genotype, a phenotype is constructed. The implementation is based on a final project in the course "Embodied Intelligence" taught by Rodney Brooks and Una-May O'Reilly in the spring of 2002

The implementation follows the paper "Evolution as a theme in Artificial Life - The genesys/Tracker System" by David Jefferson et al, UCLA (Jefferson et al. 1992). A genetic algorithm and finite state automata combine to evolve virtual ants to follow trails of food in







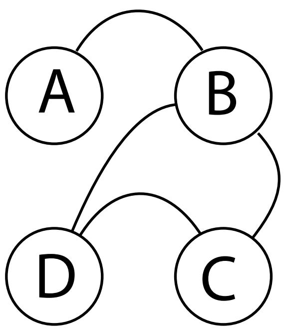
The input layers derived from the NURBS surface.

Gaussian curvature, on the NURBS surface.

Gaussian curvature mapped to UV space.

Normal direction in UV space.

a changing environment. The ant was replaced with a strip laying "creature" for finding developable strips on double curved surfaces. The GA algorithm is not novel but the extension to a search method for developable strips is developed by the author. It stands as an example of enforcing material constraints indirectly through search techniques.



A diagram of a four-state finite state automata (FSA). The automata stores its state and based on external input can switch states along the transition lines. This principle allows for complex behavior from a limited number of inputs from the environment.
For instance a human could be described as a four state finite state automata with the states:
hungry
sleepy
tired
refreshed
with food and sleep being possible inputs from the outside. The same external input has a different effect if the automata is in the state sleepy compared to hungry. Both sleep and food can shift its state to refreshed.
In the example the input is the curvature of the surface.

The strip creature

The GA for the "strip-creature" moves through the UV space of the surface and has two sensors placed at the tips of its track that record the normal vector at the current position on the surface. Using the behavioral specification in the genome, the creature moves, turns and widens and narrows its strip based on the FSA and the input of the environment. The goal of the creature is to travel along the surface leaving a strip trail where each normal vector pair at the line of ruling share a plane, ensuring the developability of the created strip. If this condition is consistently met (within some tolerance margin), the strip that results is developable. The creature is defined through a binary string sequence extracted from the genome. The length of the string is dependent on the number of states in the FSA. The higher the number of states, the more complex the FSA network of behaviors can possibly be. However, a higher state count is not a guarantee for a more successful creator because the probability for a non-functional FSA also increases.

The FSA

The behavior of the creature is based on a FSA representation. The possible actions within the FSA are move, turn left, turn right, do nothing, widen the strip, narrow the strip. With the chosen binary representation in the genome a three bit representation for the action chosen, which results in three neutral actions in order to fill all possible states of the three bits?
The number of possible states varies between 2^3 and 2^4 states, based on the experiment. Hand designed test FSA did seldom go beyond five states, but in order to give the system enough flexibility to evolve, the number of possible states is set intentionally higher. Based on the Jefferson paper, the FSA was designed to execute one of the three behaviors based on environmental input. In this case the input is the Gaussian curvature of the surface.

The genome

The bit representation of the genome places the least significant bit on the right and the most significant bit on the left. It is divided into segments where each segment represents either a state or an action related to that state. The genome is a binary string representation of the following table based on the Jefferson paper.

The genome is represented as a string of bits

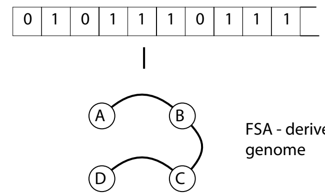
`0 0 0 0 1 0 0 1 1 0 0 1 0 0 1 ......`

Where the bits are assigned the following meanings

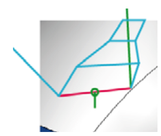
Start state	new state	action	new state	action
000	010	011	001	001

This can be expanded to a transition table

Start state	old state	input	new state	action
000	001	0	010	011
	001	1	111	010
	010	0	011	010
	010	1	111	010
	011	0	011	010
	011	1	111	100



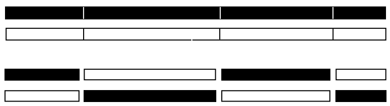
FSA - derived from genome



A genotype is extracted from the genome. A program can be written to interpret the information found there as a finite state machine. In this example the FSA instructs the movements of the strip creature based on input from the world.

A mixed model

The genome describes a transition table that defines the FSA. When the FSA is presented with input from the environment (in this case the surface), a state change is initiated and the action associated with that state change is executed. The action controls the movement of the strip creature on the surface. The creature writes a trail of step coordinates across the UV surface for a fixed number of cycles. Its method of taking inputs from the environment resembles the way Braitenberg vehicles (Braitenberg 1984) sense their environment, using two sensors with a space between them. The model incorporates several different models previously explored and combines them into a design explorer to generate developable strip solutions. In order to be able to compare the different strips that are being generated, an evaluation function is used to compare them. The length of the steps taken by the creature depends on the curvature of the grid cell at the time it takes the step. The larger the curvature, the smaller the step sizes, with 1.5 being the smallest step and 2.5 the biggest.



Mutation of the genome by means of crossover.

The evaluation model and fitness function

With each step the creature takes, it calculates the amount of grid cells that are covered by the latest segment. The fitness function measures the deviation of the normals at the tip of the strip from a plane as a measure of developability. The covered area in the strip step is scaled based on how close the strip is to being developable. A better fit creates a higher score and therefore a better fitness. If the normals are co-planar the scores of the gird cells covered by the last segment get added to the total score of the phenotype. If the normals are not co-planar then the grid cells covered are not counted. This was a design decision in the system in order to allow for regression that is for incremental adaptation of the strips to the surface. If the creature were not allowed to move when it did not fulfill the normal condition, most strip creatures would get stuck at certain points on the surface.

With the possibility of not fulfilling the normal condition and being punished for it through the fitness measure, the evolution should select the strips that are more developable. The strip creature behavior can adapt and improve from generation to generation.

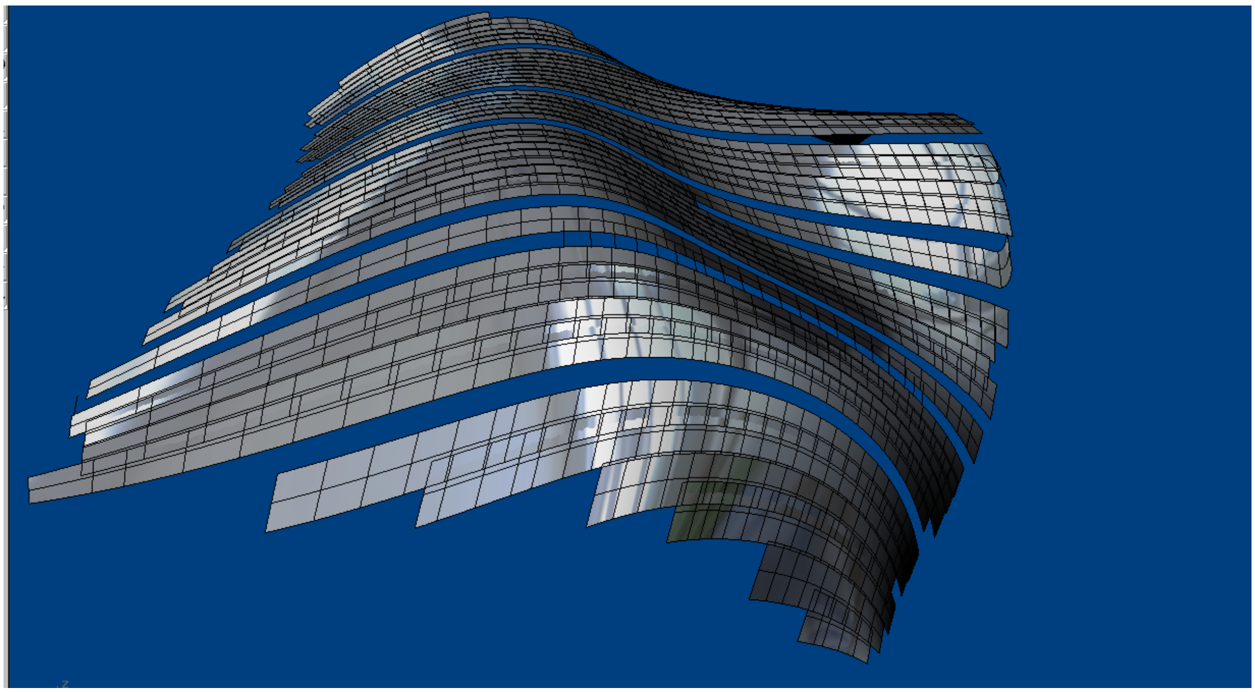
The evolution process using point mutation and crossover

Mutation of the genome takes place after all individuals of a generation have run on the surface and their scores recorded. The top 10% of the individuals are stored and subjected to point mutation and crossover. The probability for point mutation is 1% per bit. Different probabilities can be set to increase the likelihood of mutations. The initial crossover was a three point crossover; in the later runs it was changed to be one crossover. A set of randomly picked genomes are selected, and split at three randomly chosen locations. The resulting pieces are swapped and combined with their counterparts from the other genome and two new genomes result. Point mutation simply selects a single bit and flips it based on a variable probability.

The resulting mutated genomes are inserted into a new population of randomly generated genomes. This is an important part of the evolutionary process that adds new variations to the pool. If the evolution were run with the selected top 10% of the genomes, there would be stagnation over time. For this experiment, runs with both point mutation and crossover mutation were used, as well as runs with only point mutation. Because the focus of the experiment is on demonstrating the approach, the results are omitted here for brevity as they did not show any significant differences.

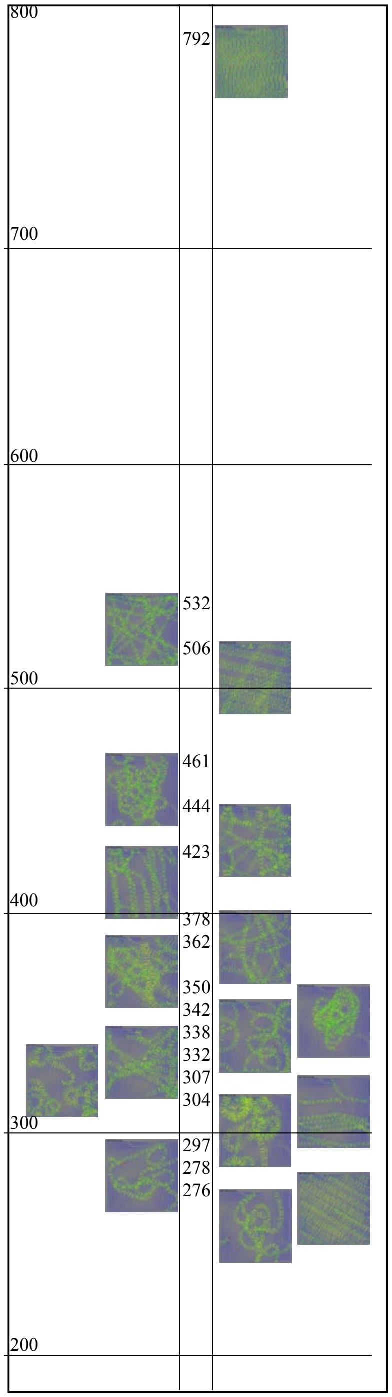
The resulting strips

After running the GA with 1000 individuals for 200 generations

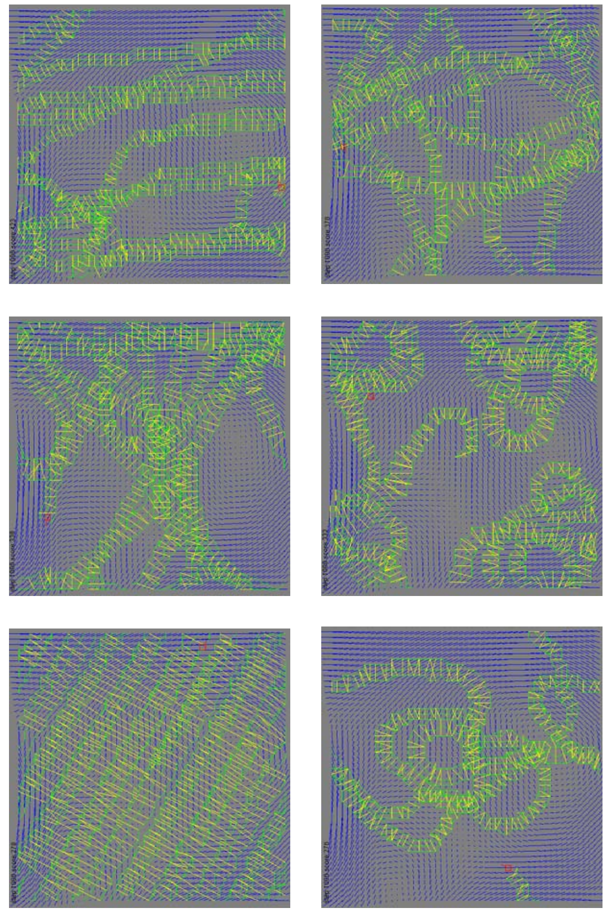


One reasonable result from a run of the GA. The problem is though that the most successful strips are the ones that have evolved to ignore the surface traits and just move straight for maximum distance.

using 1% probability of point mutation and three point crossover, a range of strips could be observed not based on scores alone but instead gathered visually based on appearance of the track pattern. The stripes are the result of 1000 steps of the creature on a 50x50 grid. A common result was a centrally located patch formed by the strip running in circles around it and wrapping back onto itself. Another type was the diagonal pattern with strips dividing the UV space in roughly parallel strip patterns. This type must act relatively isolated from the surface feature inputs in order to run



Ranking of the phenotypes based on their performance in laying down developable strips.



straight for such a long time. Since the surface is toroidal, it can be covered with one strip that wraps around over and over without changing direction.

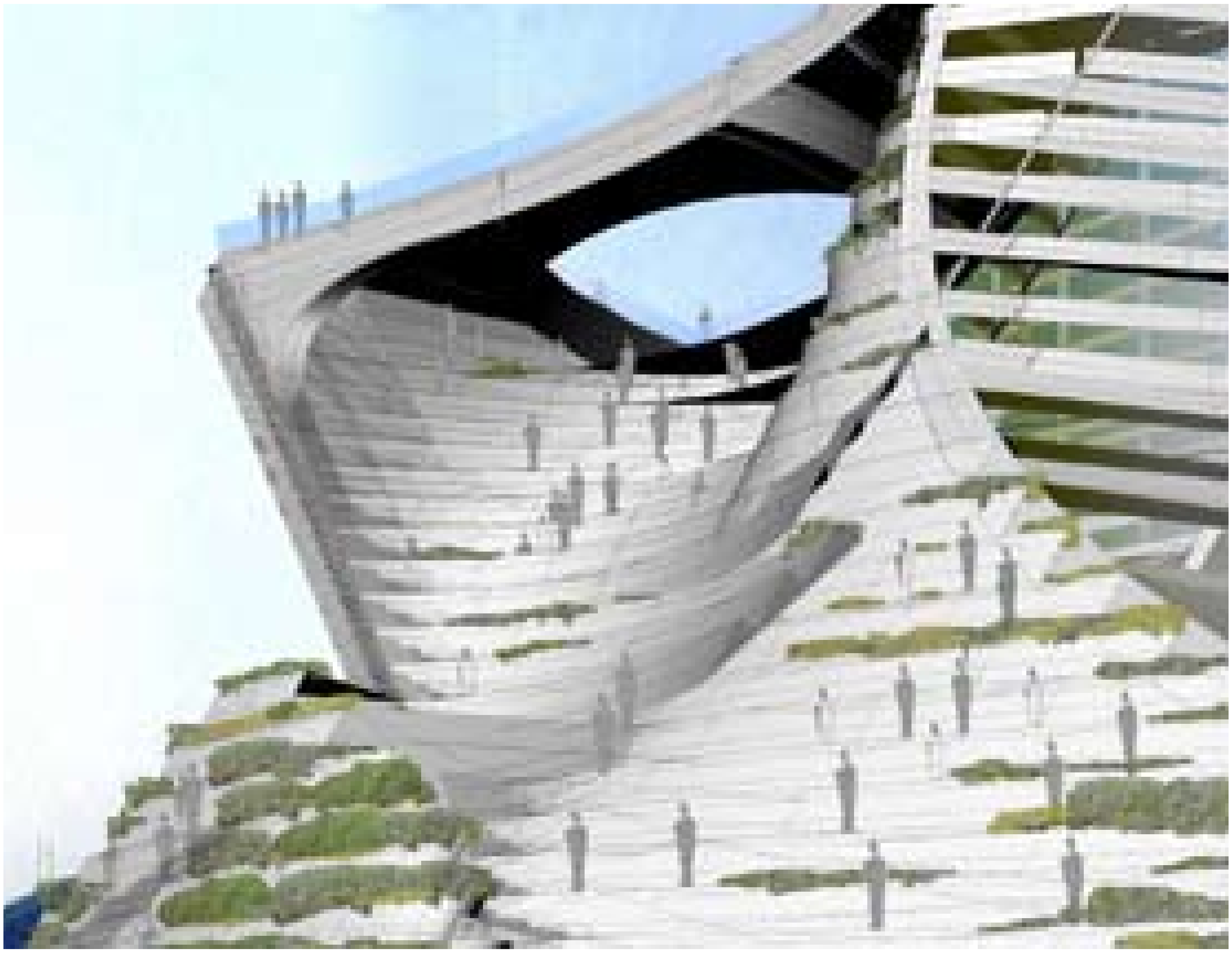
The crisscross pattern

The strip conforms to the surface features and follows ridges and folds caused by the toroidal surface model that cut across the UV plane from different angles. This is the most promising output, although it is very chaotic it does open up the opportunity of a new approach to subdividing the NURBS surfaces with strip patterns that emerge from the surface features.

Strip results

The strips on the surface present a wide variety of shapes and forms and it is difficult to evaluate them. One method is to look at


XX 100001



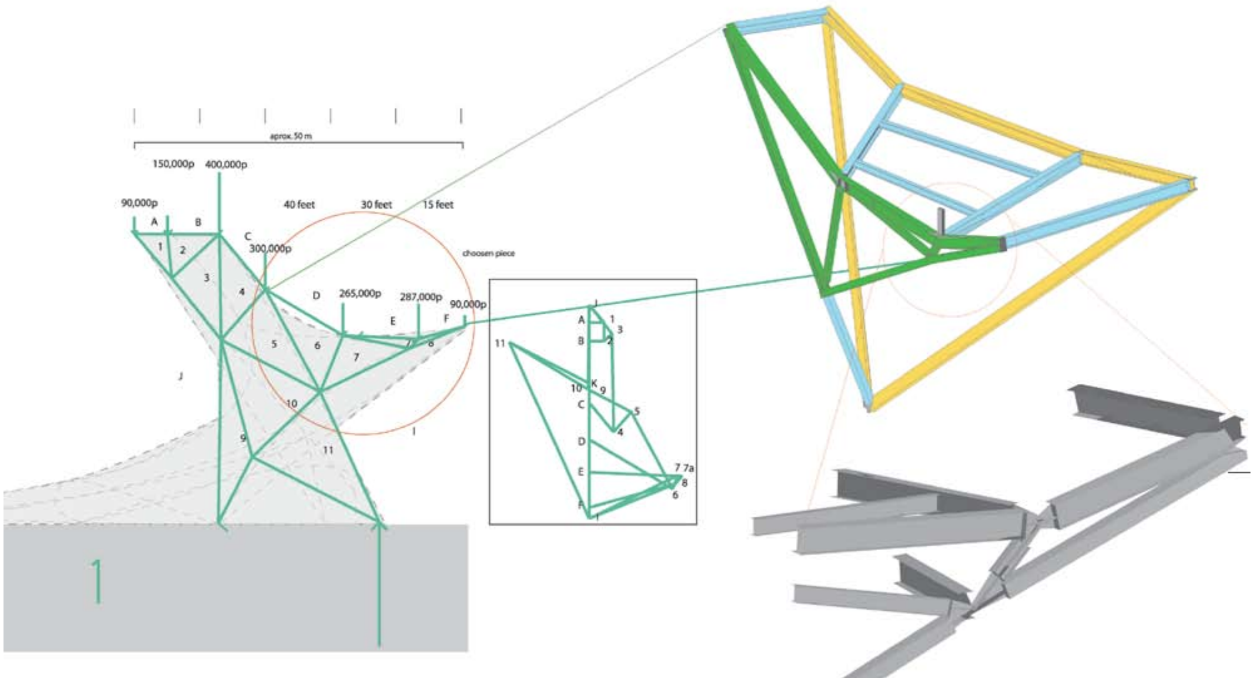
Competition entry to the Busan tower competition, 2004.
Team: Axel Kilian, Michael Fox, Elite Kedan.
The further development of the tower by the author led to the interest in an integrated form finding approach.
The following studies show structural frame explorations of this project.

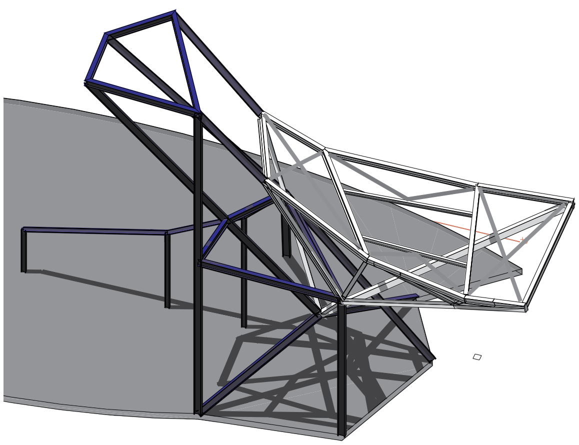
Facing page: One of the original competition panels submitted by the team.

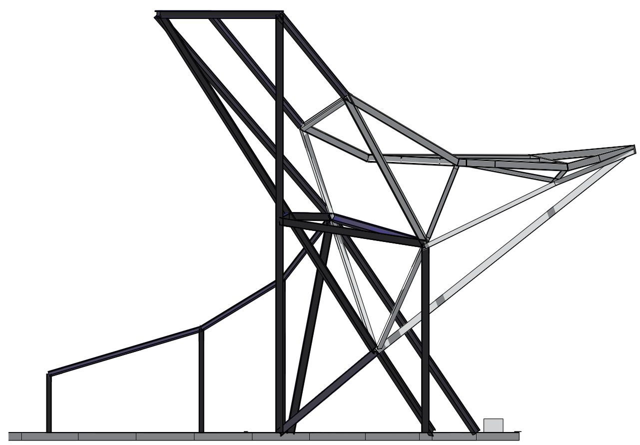
the scores and therefore see their overall compliance to the normal test. That gives some measure of how successful the individual was in constructing a phenotype in the surface environment. However, there are functional and stylistic considerations that are not necessarily captured by a simple ranking like the current scoring implementation.

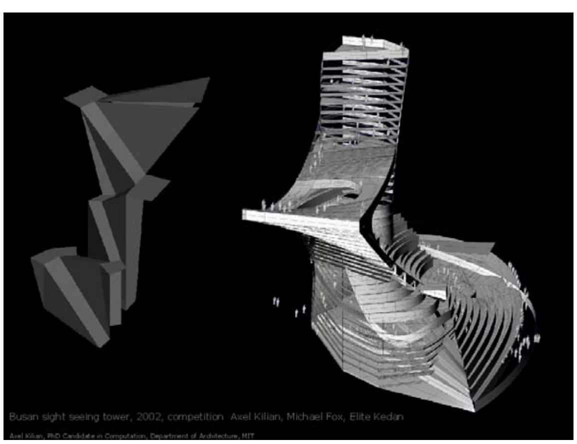
Ranking

The diagram ranks the produced strips based on the resulting scores. The somewhat parallel/diagonal strip pattern can be seen high on top the scale. Crisscrossing strips dominate the upper middle region where the creature follows the surface features directly by turning and varying its strip width and thereby weaving a pattern across the surface. In the lower middle region one can find circling



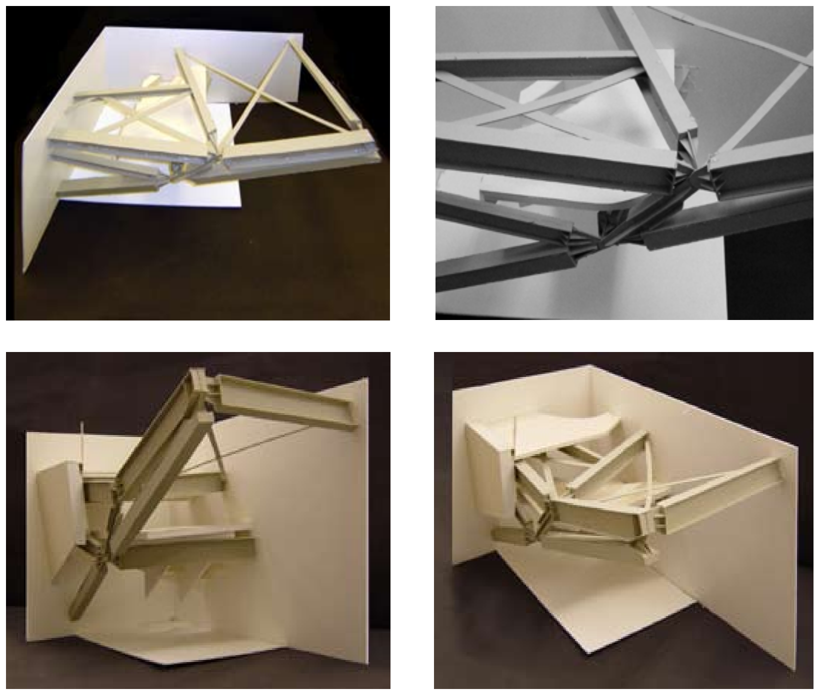






Form and force study for the design entry for the Busan competition. Graphic statics were employed to create bidirectional constraint geometries that allowed for the live exploration of the interaction of force and form polygon using the bidirectional solver in CATIA.

strips, which scribble across the surface and often intersect their own path, creating knots and large blobs of surface patterns. By creating a scale, it is possible to fine-tune the results by adjusting the score region that is selected for reproduction to a region that contains the individuals with the desired results. Of course it is not guaranteed to produce only strips of that kind because diagonal types that do not perform as well are also selected.



Different views of a selective prototype of a part of the theater in the sky section for the cantilever of the Busan tower.
Physical models from digital geometry.

5.3.3 Main Experiment: Digital Hanging Model

The following sections are based in part on the paper "Combining Form Finding and Fabrication" published by the author at Acadia 2004 (Kilian 2004).

Form finding environments shift the focus to the design of systems rather than geometries. Physical hanging models are very compelling design devices. But a number of factors limit their use for designers. First, they need to be relatively large in scale to give accurate results and allow measurements with reasonable tolerances. Equilibrium solutions can be scaled if the proportional distribution of mass is kept and the geometry of the lines of forces is scaled proportionally. This holds true even though mass does not scale proportionally to geometric dimensions. The results of the physical model are therefore usable for a full-scale building design, at least as a starting point. But the amount of work needed to construct a fully detailed hanging model is large. Also, it is a very time consuming task to adjust the model when larger changes within its geometry occur, as the model is inherently interdependent. A small change can ripple through large parts of the model requiring adjustments which themselves cause shifts elsewhere. The model will eventually find a state of equilibrium, but it is not guaranteed that the new state matches the desired form. This may require several iterative adjustments.

The second major disadvantage is that a physical model is hard to measure accurately and in reasonable time, as measuring requires physical access to the model. The measurement of forces within the strings of the model is even more difficult, as it requires the installation of strain gauges, which is time consuming and can potentially disturb the model. In addition, the measurements are not part of the design process. The design is frozen to allow for iterating through the load measurements throughout the model in a given state. If one ignores a small earlier hanging model, Colonia Guell was Gaudi's only design developed with the aid of a hanging model. The model was produced between 1898 and 1908 by a highly qualified team. (Tomlow, et al 1989)

The digital version, in contrast, allows simultaneous measurement and creation as well as editing of geometry. These measurements can directly drive other dimensions in the model. In the digital model, editing and creating the string weight is less limited by the availability and preparation of the physical material, which in the case of a complex model can slow the process. Furthermore, the use of generative techniques allows for the rapid placement of complex string constructs and allows the observation of their behavior before investing time into an elaborate physical model.

Finally, it is possible to create the topology of the model in a frozen state to establish the pure connectivity of the model and to then subject the model to simulated gravity and observe the form that emerges. This is a substantial advantage over the physical model, where gravity cannot be suppressed, which makes topological layouts harder to manage.

5.3.3.1 Physical form finding precedent

The work by Gaudi and others has already been discussed in the introduction chapter and in the precedent chapter. The following is a short a reiteration in the context of parallel exploration.

Antonio Gaudi's hanging models

Antonio Gaudi developed integrated design exploration the furthest by developing physical design models to explore, store and calculate his designs in a collaborative, physically shared approach. He relied on physical space as his computational environment, as gravity and strings enforce axial force conditions through their interaction. But as shown above, he did not stop with merely finding or optimization of geometry. He also interpreted the resulting forms and treated them architecturally, an important step to not slavishly follow every detail of the abstract model but rather knowledgeably replace and adjust, and ultimately heighten the architectural reading of the desired space. All executed within the design intention embedded in the hanging models design exploration.

Hanging models enable the designer to determine the optimal form of structures carrying loads purely in compression, particularly those that mainly consist of vaults. (Tomlow et al 1989) Although Gaudi's hanging models are the best known examples, earlier, less sophisticated attempts with hanging models were made by Heinrich Huebsch (1795-1863) and Giovanni Poleni. (Tomlow et al 1989)

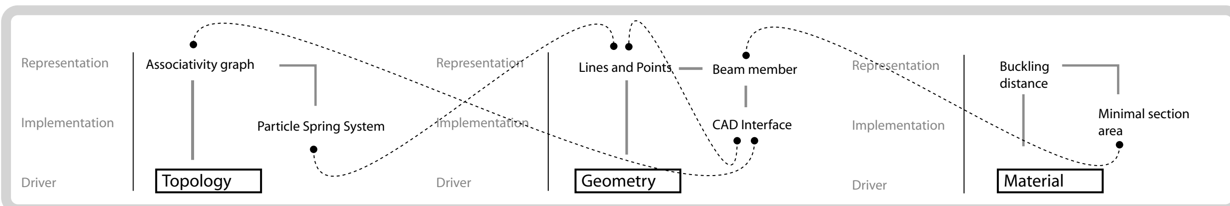
For some of Gaudi's projects like Sagrada Familia (not derived with a hanging model but from a staking plaster model), the translation of the force diagram into form was guided by two constraints: The extrusion process of the plaster model building, and the technique of a stonemason to build surfaces with straight lines of ruling. Therefore, Gaudi's Sagrada Familia combines two design constraint models coming from different ends. One is the overall structural geometry designed to be in equilibrium and in compression only, the other is the mode of construction using ruled surfaces only. There was a strong parallel between the work of the plaster mould-makers and the actual full-size construction of the window (in the Sagrada familia). In both cases, the straight lines in the ruled surfaces were used in a similar way to generate moulds. (Burry 2001)

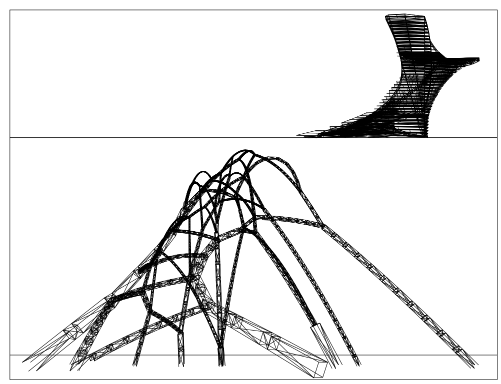
However, the hanging models do not implement the ruled surface constraints for the envelopes, and neither does the ruled surfaces rule take the exact distribution of mass along the structure into account. It requires an expert's interpretation to make them work together. Gaudi saw forms and once he had determined them mentally, he sought the means to transform them into physical, buildable objects. (Giralt-Miracle 2002) A combination of multiple constraint models in a digital simulation would yield a wider range of exploration in both overall proportion and composition in correspondence with the surface of the building components.

Frei Otto

Otto is different in that he was at the same time a researcher and teacher. He collaborated widely and therefore his beliefs, knowledge and architectural posture was not limited to his oeuvre alone but in fact built up a following in southern Germany and around the world. His buildings follow the principles of lightweight building over individual architectural expression. Otto also generalized the principles of form finding based on physical models to a large group of physical phenomena beyond strings and chains, this extended to soap bubble minimal surface and highly accurate photometry methods for extracting data from the physical models. But as for Gaudi, the models are not taken as a given but interpreted by knowledgeable and critical engineers and architects who add the architectural and engineering details making them reality in their material and scale context.

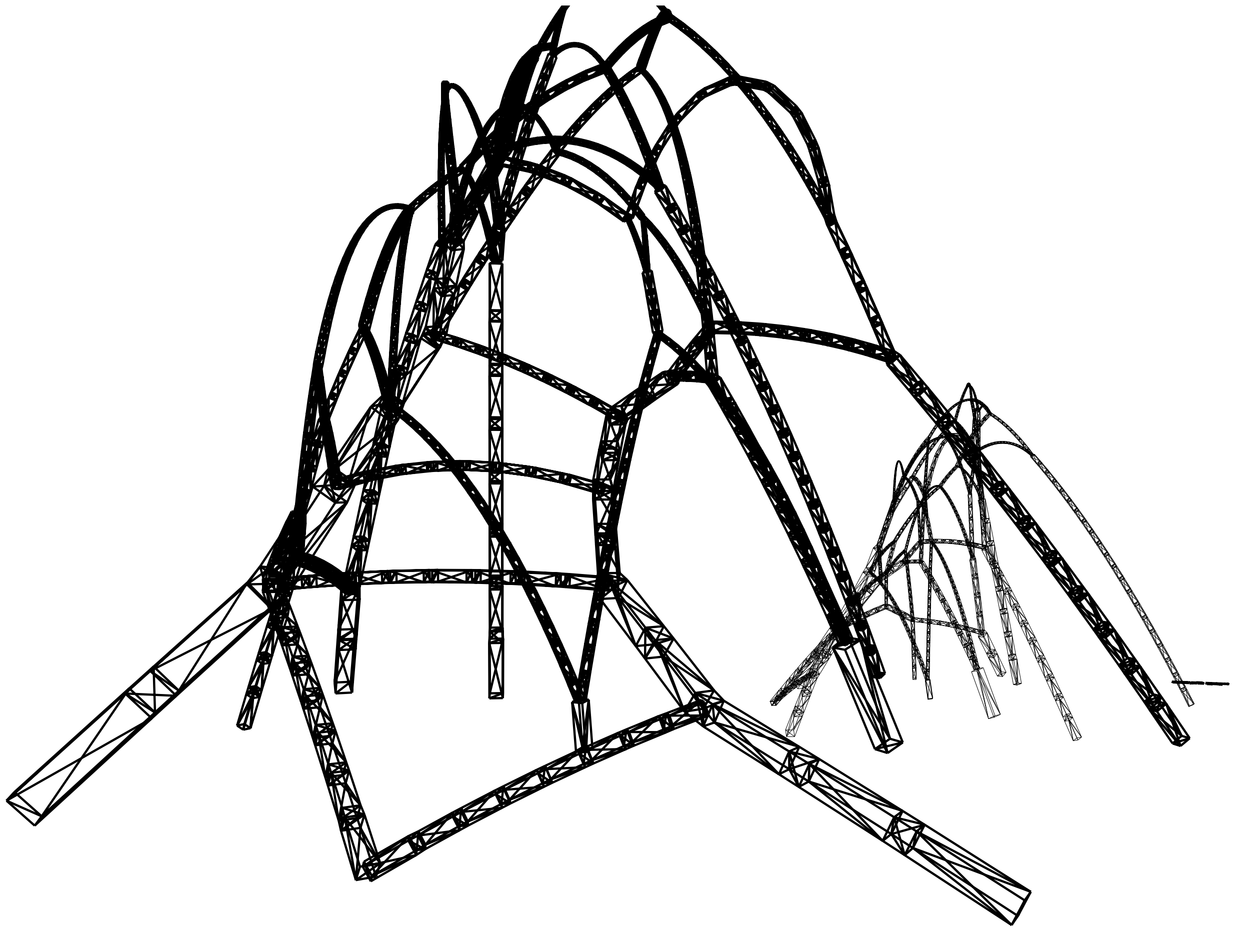
Heinz Isler – Physical form finding models



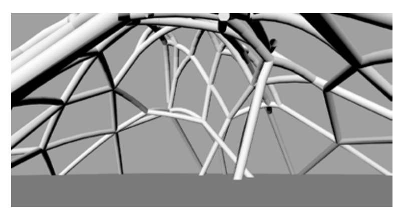


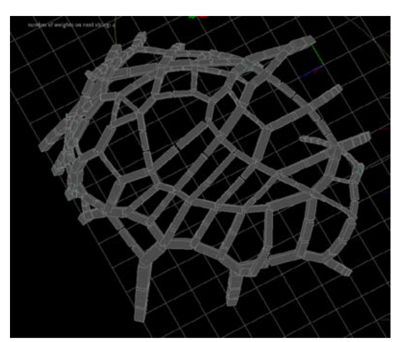
Extended form finding experiment

Isler is probably the biggest perfectionist and most isolated of the three. Working delinquently on his shell structures in the basement of his house, taking great care in the creation, measuring and even monitoring of his shell structures he has an intimate bond with his design work which also reflects a level of respect for those fragile and complex engineering structures understood only by few in the depth of Isler. His example raises doubts whether it will ever be possible to work with such level material reduction in less regular structures when an expert like him is not available for such projects. But maybe some of the knowledge can be captured and generalized and made available in digital environments to calculate and evaluate novel forms with the elegance and simplicity of Isler's shells, which at the same time open up the architectural



Hanging models generated with the tool



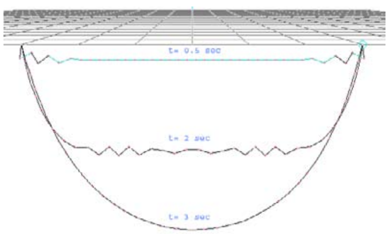


possibilities.

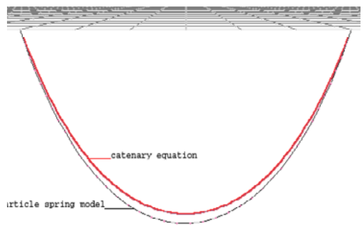
Once the measuring is completed, it is a very time consuming process to make any alterations to the design. But physical models built in the real world also ensure that the model is not the artifact of a selective simulation strategy that may or may not include key parameters necessary to model the structure accurately. Despite the challenges they pose in achieving accuracy and scaling of material and mass, physical models do ensure a more holistic simulation of the problem than an image based computer simulation.

5.3.4 Extending the Concept of Form Finding

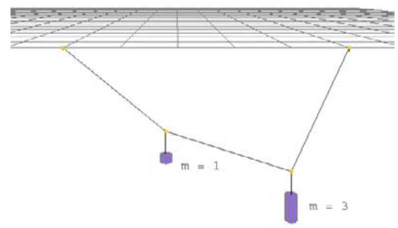
Using physical form finding techniques as an engineering technique has precedents reaching back to the Renaissance. In the early 20th century, Gaudi pioneered them as a design tool in three dimensions using hanging chain models. He produced stunning works of architecture that convince through structural and sculptural elegance. However, little is written about the process whereby the abstract string based chain model with point loads is translated into a volumetric geometry with distributed loads and structures with self-weight. A weak point of Gaudi's technique of



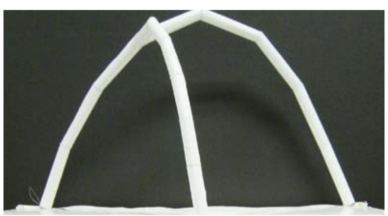
Progression of the particle spring chain falling over time.

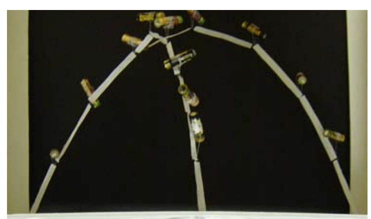


Slight deviation from the equation based hanging chain are possible with high spring counts and very low stiffness.



The weights can be edited and show the variation of the hanging line.

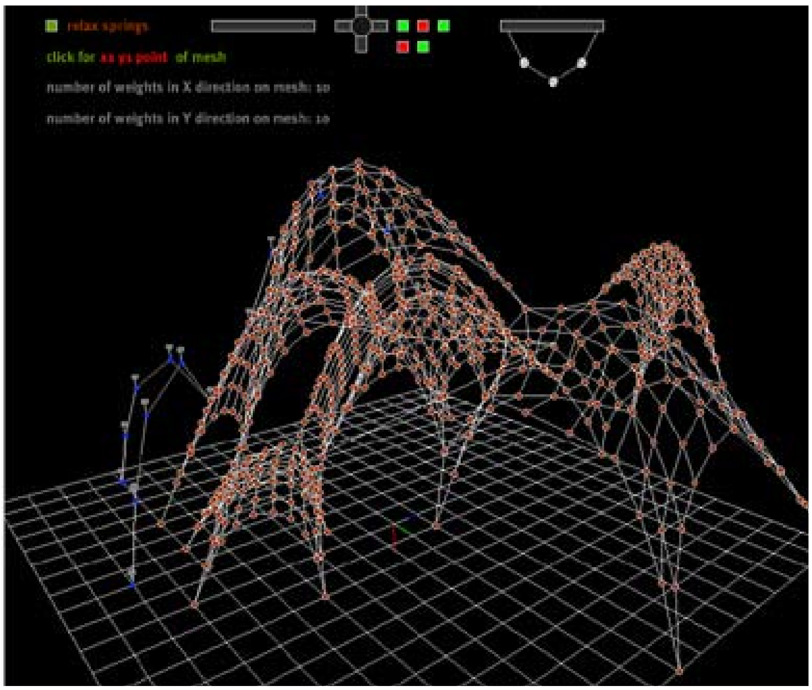


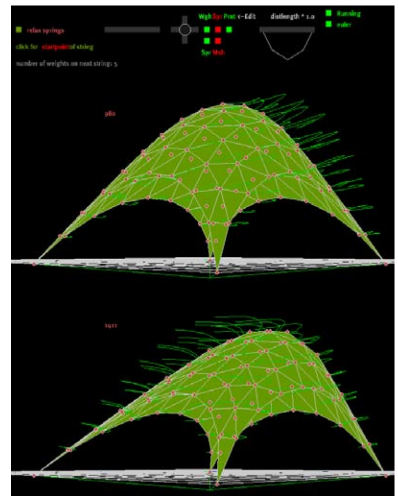


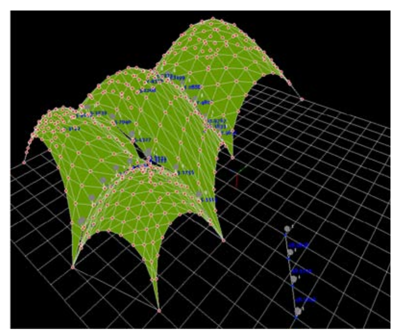
Comparison between digitally created and fabricated standing form and its equally dimensioned hanging equivalent..
Left model in compression with a material cross section. Right model an inverted string model using the same geometry

the physical hanging chain model is to not give information on the optimal distribution of stress in the material of the built form (Tomlow et al 1989).

The author proposes techniques to integrate this translation into the design stage of the hanging models. This thesis presents a digital hanging model based on a particle spring systems library implemented by Simon Greenwold using Euler and Runga Kutta solvers programmed in Java. The goal is to provide a real-time three-dimensional-modeling environment that allows the design of gravity based forms following the hanging chain principle. By using the same building components it is also possible to model any mesh topology, for instance approximations of Heinz Isler's grid shells or fabric like surfaces. The initial tool was further developed and tested in a workshop setting in order to validate results and provide a more solid implementation and input from students to the tool. The relationship between form finding model and the translation into volumetric form was explored in a series of small models.



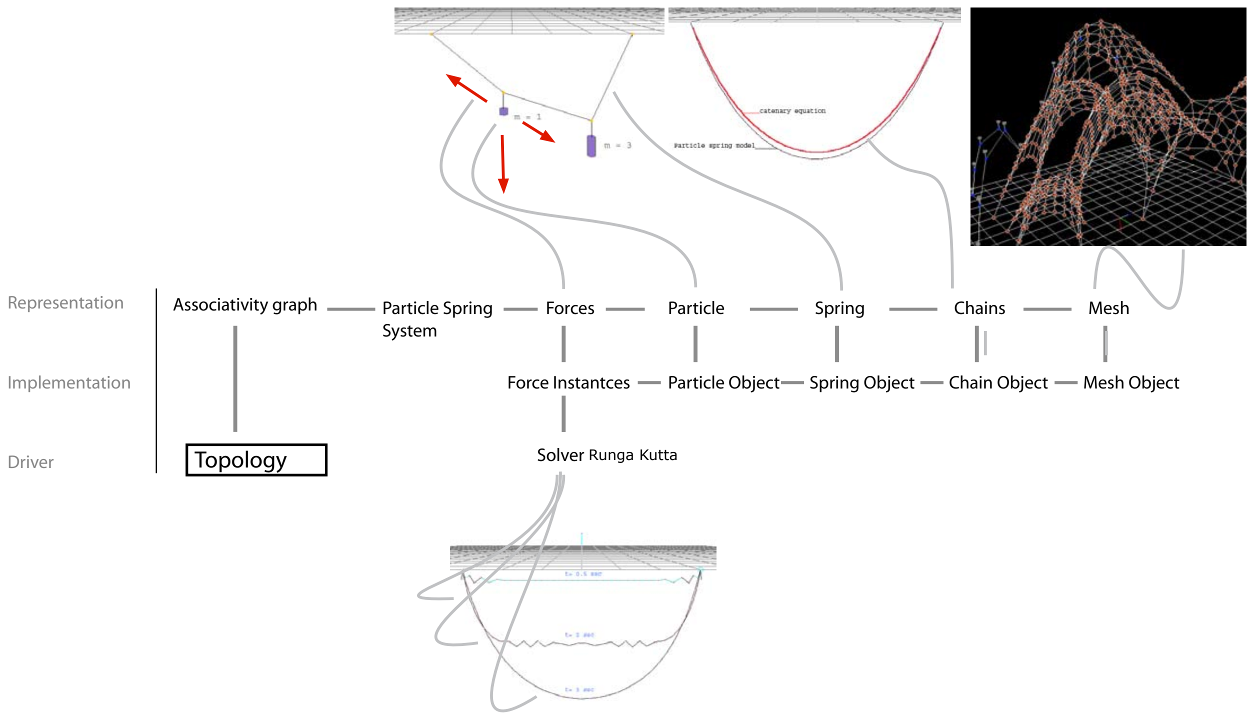




Different features for exploration were tested. One was the particle tracker during simulated sideways forces. The position of each particle was traced over time and gave a good visual sense of how the structure was responding
Stitching of surface patches through zero length springs.
On the left an example of a nicely balanced mesh. All examples were developed as extensions to the Java application by the author.

5.3.4.1 Equation-based approach

It is possible to calculate a catenary curve between two support points given a string length based on a parametric equation. The hyperbolic cosine forms the basis for the catenary equation and allows the calculation of any point on a hanging string of uniform weight, which is supported at two points. The vertex corresponds to where t = 0, a parameter that determines how quickly the catenary "opens up." (Weisstein 1999). The amount of "sagging" is related to the string length and the distance between the supports. The equation can be adapted to uneven support points as well. Paul Cella noticed that textbooks of mathematics, mechanics and engineering practice produced what appeared to be a settled conclusion: When the supports of a catenary are



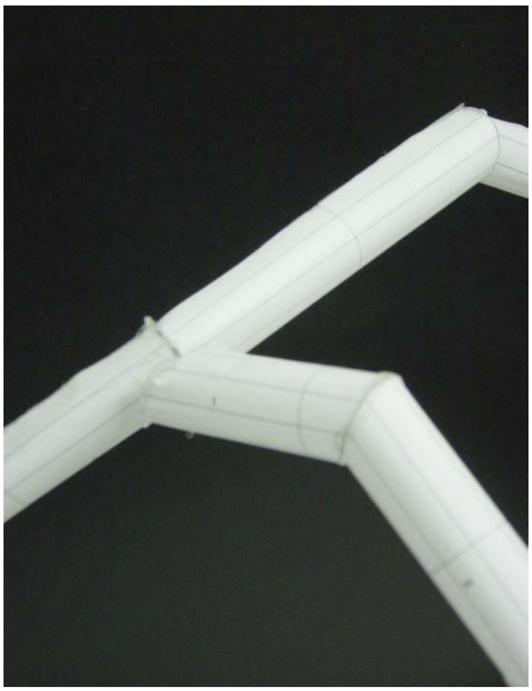
The topology constraint drives the initial setup of the model
The topology relevant information is represented with a particle spring system and implemented with instances of members of the system such as springs, particles and surface meshes.

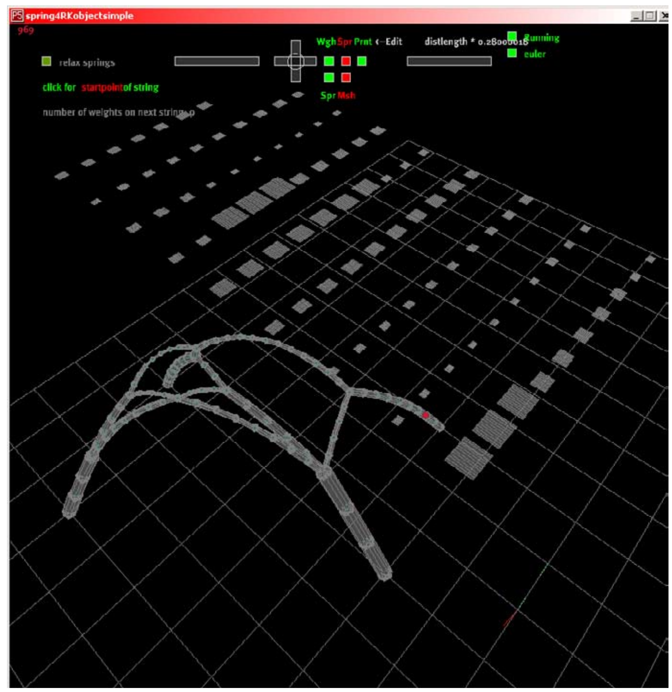
at different elevations the mathematical complexity precludes a theoretically correct solution and a parabolic approximation is the recommended approach. (Cella 1999). He subsequently derived the catenary equations to calculate the uneven support catenary problem.
What the catenary equation approach does not provide is a way to solve undetermined structures, for instance if four strings are joined in one node and no single solution exists. The subparts of the catenary between support points can still be solved with an equation, but the location of the supports in the general case can only found through a solver-based approach. This is where the use of a solver is necessary in order to determine the overall geometry for the equilibrium of forces in the structure.

5.3.4.2 Computational solvers for form-force equilibrium

A more general approach to the problem is to use particles that represent a point mass in space and has a position and velocity as well as an acceleration property. Based on Newton's law, force acting on a body causes acceleration, which is inversely proportional to the mass of the body in the direction the force is applied. We can formulate a system of equations that can be integrated analytically

Joint studies in CATIA exploring the intersection of different diameter tubes.



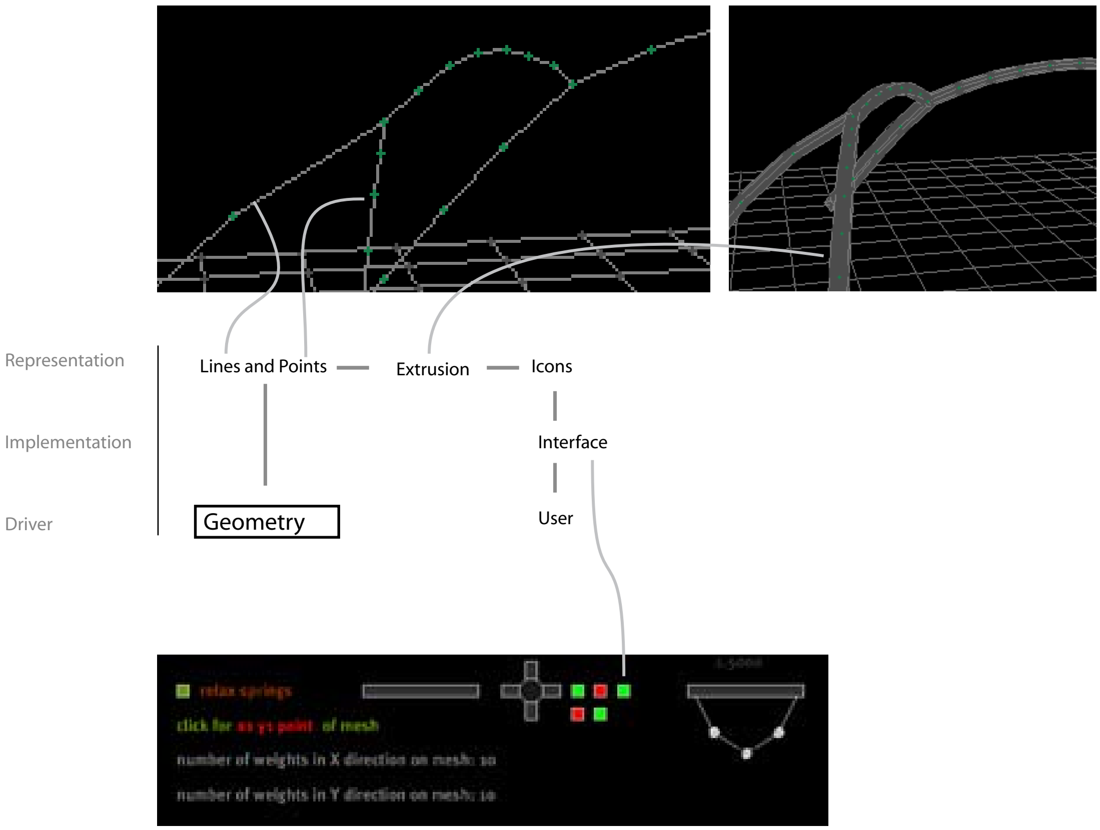


Volume based on the forces present in each of the members. The patches in the background are cylindrical unfolded surfaces of the tube segments, which are being updated live with the moving structure.

On the left a paper mockup of an intersection of the structure.

to solve for the position of a point mass with respect to time. With the introduction of gravity, linear spring force, and viscous drag it is possible to construct a particle spring system with simulated gravity. For the simple determined cases, analytical integration works. But in more complex cases, it is necessary to integrate numerically. Different methods have been developed, among them Euler, midpoint, and Runga Kutta. (Baraff and Witkin 1999)

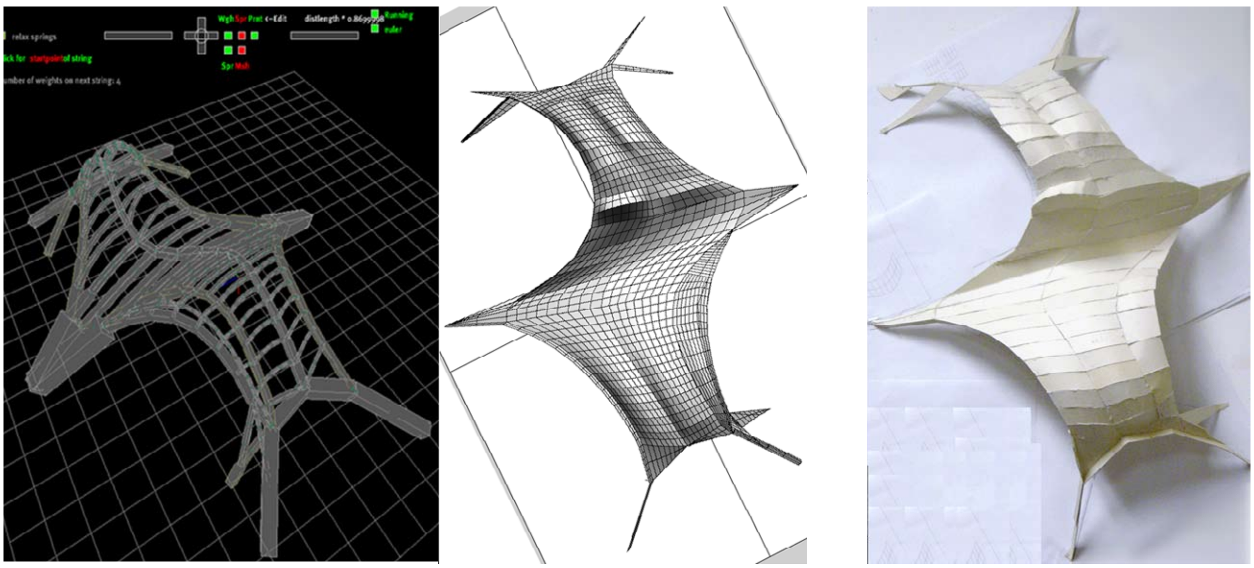
The particle spring library uses an explicit Euler solver for mesh simulation, which is satisfactory for relatively low stiffness of the springs. During a workshop at MIT in the spring of 2004 initiated



The geometry constraint of the hanging model. The length and positions of fixed particles are specified and influence the emergent overall structure. The geometry constraints are enforced through maximum spring lengths and fixed position constraints.
In addition a geometric volume is generated based on the spring force in each segment. The area is generated based on the maximum local buckling length for the cross section under a given material.

by John Ochsendorf and co-taught together with the author, Barbara Cutler, Eric and Marty Demaine, and Simon Greenwold, a second implementation was written by the participants, which can handle meshes with very high stiffness and uses an implicit version of Euler and Runga Kutta to get stable solutions. Cloth strongly resists stretching motions while being comparatively permissive in allowing bending or shearing motions. This results in a "stiff" underlying differential equation of motion. (Press et al 1986) Explicit methods are ill-suited to solving stiff equations because they require many small steps to stably advance the simulation forward in time. (Baraff and Witkin 1998) Strings in the hanging model have similar characteristics as cloth as they have very stiff, meaning non-stretching segments making up the string.

This computational method allows the interactive construction of string-like constructs out of particles and spring elements that approximate physical hanging model behavior when subjected to gravity. This approach is not new but is well established in the computer graphics community and the animation industry. It is novel to use it as an interactive modeling environment for designers that allows not only optimization but also playful exploration of



A series of steps in the form finding based fabrication of a sample roof. The form finding model shows the forces in proportion to the segment cross section area, based on local buckling length. The buckling length equation uses a material parameter to determine the maximum allowable force for a given cross section and segment length.

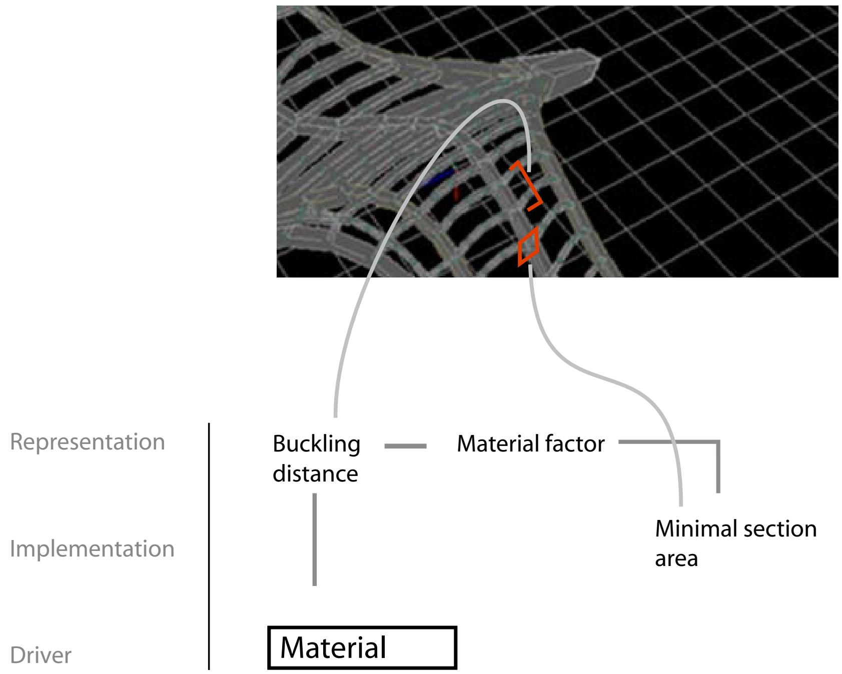
the evolving structure. The system is referred to either as mass spring system or particle system. For the remainder of the chapter the term "particle spring system" will be used.

5.3.4.3 Evolving form from topological and quantitative constraints

The parallel structure of the exploration in the form finding example includes topological constraints in form of the connectivity diagram, and quantitative constraints on the length of the connecting elements, in this case the spring elements.

5.3.4.4 Secondary constraints

Secondary constraints are the quantitative constraints of the allowable forces for a given material in the generation of the



The material constraint is embedded through a material parameter in the buckling length equation and optionally as a factor in self weight of the structure.

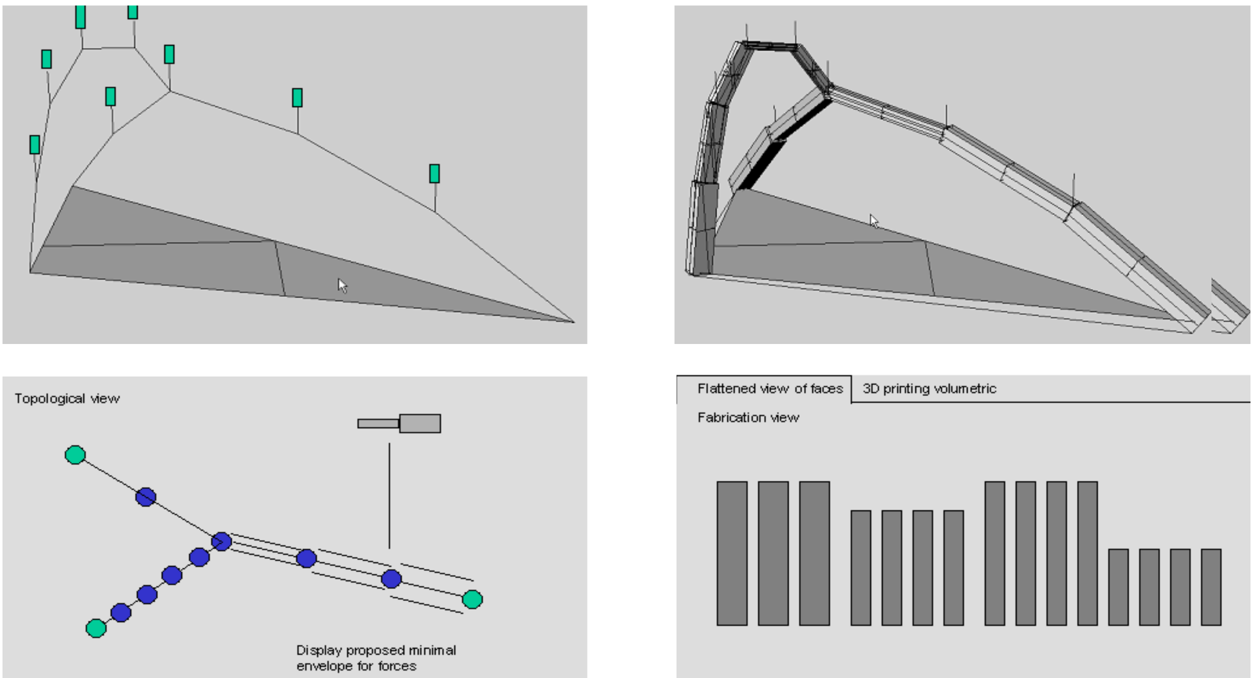
force dependant cross sections of the evolving geometry. In the implemented example, the envelope weight is not feedback into the form finding cycle due to the challenge of creating circular dependency that might cause the solver to fail.

5.3.4.5 Implications for design

The availability of general purpose solver architectures could influence the design process in terms of the design process as the initial approach to a design problem could use constraint explorers that allow for the general formulation of the design task and a response from initial design gestures through the solver architecture. The interesting aspect is that the potential for faster construction and adaptation of both the design explorer and the emerging designs could be facilitated in a digital environment and the chosen approaches would be less dependant on the properties of physical space to be implemented which opens up the field to more abstract design constraints to be integrated,

5.3.4.6 Iterative adjustments rather than top down design

The formulation of a design must change from the earlier presented



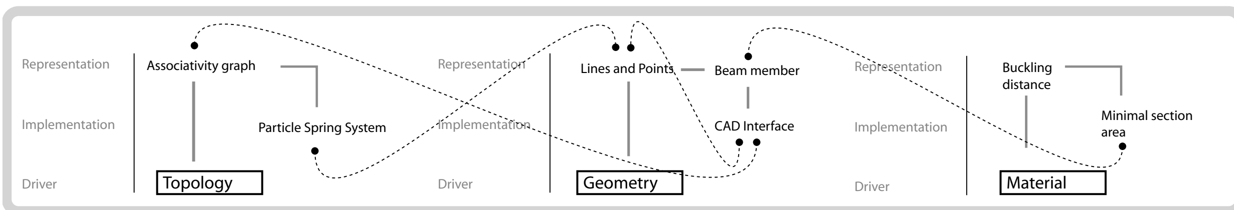
An interface sketch inspired by the parallel approach to design in Generative Components and in the "Alice" programming environment. It shows different design representations: topology, geometry, fabrication views. In an ideal scenario the design can be driven from any of these representations interchangeably.

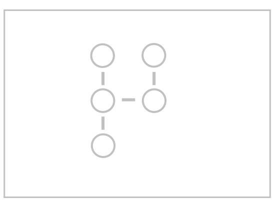
examples, which rely on top-down descriptive geometric hierarchies and are very useful in structuring a design project and introducing dimensional and limited topological variation but they proof less helpful if constraints and design goals are co-evolving with the exploration construct.

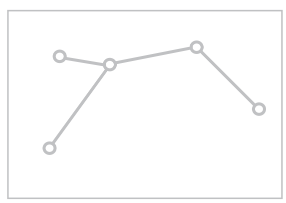
5.3.5 Between Design Intention and Optimization

The form emerges from the interplay between design intention and constraints. There is no optimum without constraints. The interpretation and weighting of the constraints is part of the articulation of the design intention.

Engineering presents optimization and computational methods for constraint resolution often as the optimal result. Optimal might mean numerically accurate within the framework of the analysis and optimization but the boundaries and definition are subject to design considerations as well. The exposure of the boundaries and the constraints in a dynamic interactive simulation, such as the hanging model, allows for the exploration of both the optimal solution within a given boundary, but also for the adjustment of those boundaries on the fly to observe the design response.







The constraints and design drivers in overview:

Topology
Geometry
Material

The expression of design intention shifts away from the literal form giving and towards the specifications of the contextual parameters and starting conditions and of course the system of optimization itself. In the light of this argument optimization becomes more of an iteration, or design exploration, where both solution and starting condition are shifting constantly in response user adjustments and system feedback. It is likely that different points of equilibrium emerge in such a process than when the optimization is linear and only start goal oriented. The setup of the system resembles more the human system interaction described by Wiener in his book Cybernetics, in its feedback cycles between user interaction and system response (Wiener 1942).

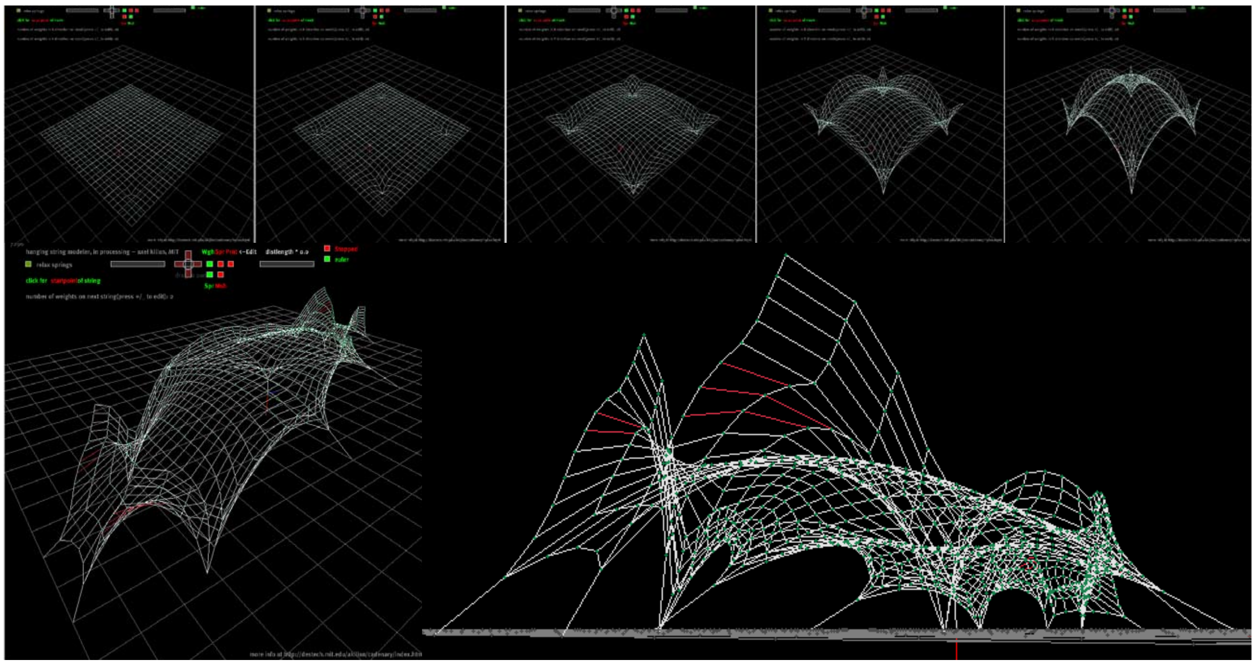
5.3.6 Translation of Topology to Physical

Architects who have used hanging models struggled with the translation of the two-dimensional elements of the network of strings into the complex spatial curvature of the three-dimensional surfaces necessary to stay true to the model. In "Das Modell" it is stated,

> "Apparently as a side effect of this struggle to follow constructional forms based on hanging models, for the first time in architectural history the hyperbolic paraboloid form was tried out in a building." (Tomlow et al 1989).

The hanging model provides a line model for the load paths for a given distribution of weight. However, it does not specify where the envelope lies in correspondence to the load path.

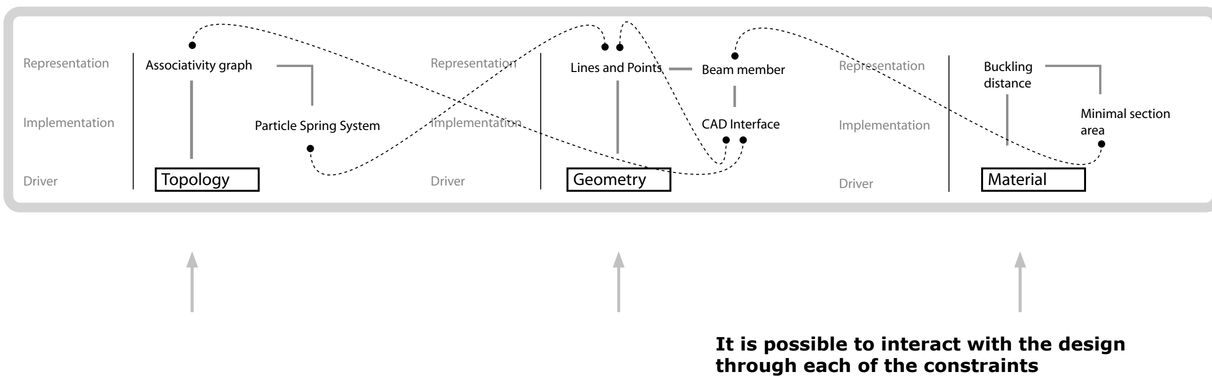
In general, the self-weight of the load-bearing member contributes only negligible amounts to the structure locally and therefore does not substantially affect the hanging curve form. If there is no load present other than the weight of the structure itself, the self-weight becomes the dominant form giving factor. The cross section has to provide enough area for the forces traveling through it. A further optimization of the structure, for example with the aim of achieving



Changes in the design process due to the form finding process. The first step is the definition of the constraints of the environment. The second step is the specification of the topology. The starting geometry based on the topology reveals little about the emerging geometry. As the design evolves and responds an overall form slowly settles from the sum of all the constraints acting on it. Due to the solver function even if a form reaches an equilibrium point it is still constantly moving.

uniform loading in compression throughout the same material by varying thickness, was not undertaken by Gaudi. (Tomlow et al 1989).

The digital hanging model does create varying thickness of the extrusion along the members based on the forces present in each member. The resolution of the simulation is also a factor in determining the volumetric form. In order to keep the number of



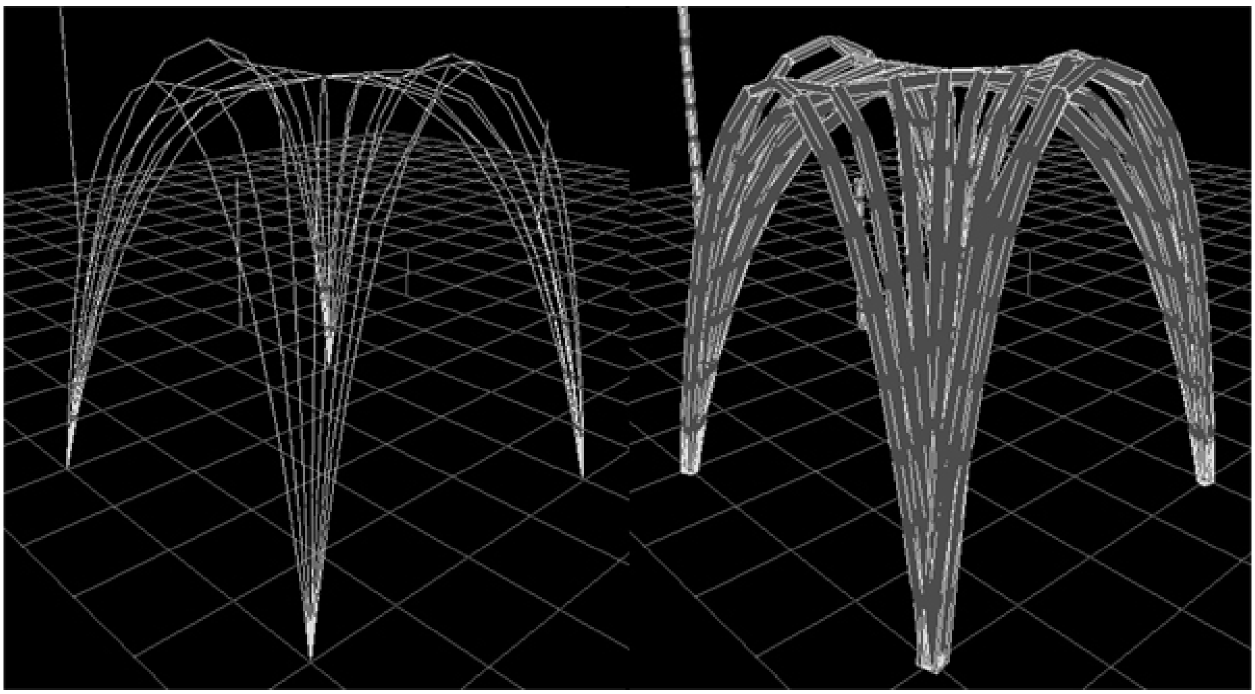
Each constraint provides an interface to the user to interact and edit the design with,

particles in the simulation low, the number of particles a hanging line is usually relatively low. This leads to polyline forms with straight spring sections in between. If the resolution is too low, it significantly offsets the load path from the ideal catenary. In a later version, dynamic subdivision of the springs based on offset measurements from the ideal catenary will be implemented.

The most straightforward translation from the line skeleton into a volumetric entity is an extrusion of a profile along the spring vectors with its diameter being approximately scaled to the forces present in the particular section of the string. Additional factors for determining the cross section are local and global buckling.

The shape of the extrusion can be varied to produce different cross sections depending on what material is being used. The load path

must lie completely within the geometric envelope in order to keep the structure in equilibrium. Furthermore, if the material cross section is not supposed to be subject to tension, the load path has to lie within the innermost third of a symmetrical cross section.
A bigger question is how to develop the intersection between members of differing cross sections and spatial orientations. At acute angles, the intersections can become quite large in



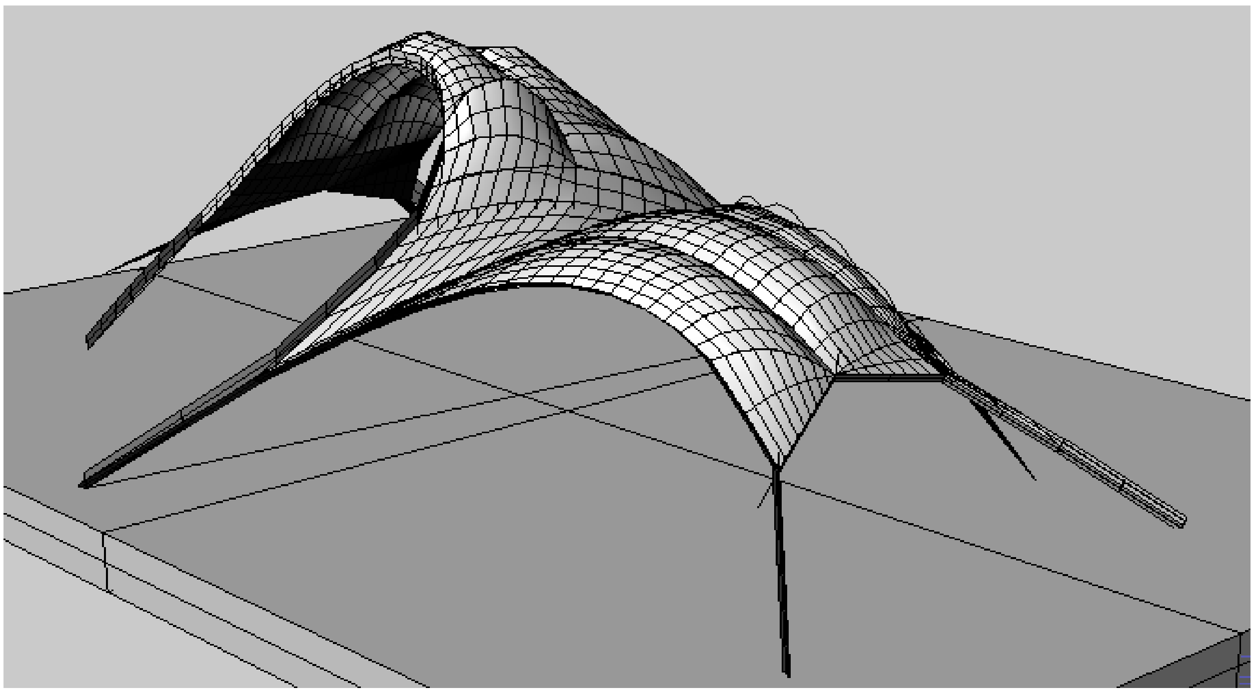
A truss based roof sketch.

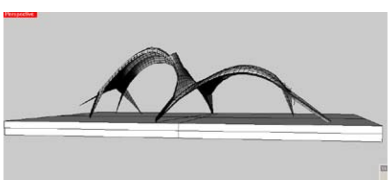
comparison to the members themselves and create difficulties in applying a uniform joint system. Parametric studies have been done to study joint systems to take this variation into account.

5.3.7 Roof Example

The roof example was a short term physical mockup of a roof network constructed and form adjusted in the particle spring system. It showed satisfactory correspondence to the predicted form and structural responses of the digital model. Under load, the flexing of the support beams showed the expected force accumulation seen in the digital model.
An example model was produced from a digital hanging model and

fabricated to validate the approach. For the translation of the wire frame model into a surface, Rhino was used as an intermediary.
The physical model was subjected to vertical loads similar to the forces in the form finding simulation and the response of the supports was observed. The deflection due to the accumulated loads of the shell clearly shows at the two middle supports of the roof in the physical model, which corresponds to the force





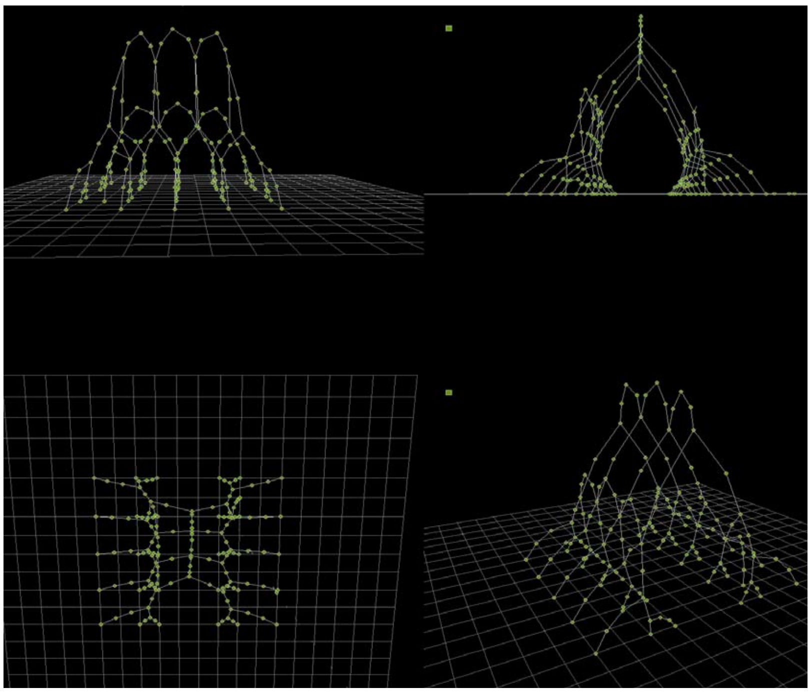
Digital model of the roof.

distribution in the form finding model. In addition, the physical model clearly demonstrates the structure's vulnerability to forces other then gravity. When the structure is loaded laterally, it is much more elastic in its response than under a vertical load and does not act as a shell.

5.3.7.1 Experiment 1: Designing in dynamics vs. analytical approach – design by discovery

Current design software supports the creation of geometry through geometric operations aimed at creating solids, wire frames and surfaces. This geometry captures the design intention to a point and serves as the communication platform for many interdisciplinary discussions. Structural analysis is usually done using this geometry, or with specifically created geometry based on it. This analytical step requires a relatively large investment of time and does not easily allow the designer to go back and change things. In addition, the results of the analysis do not immediately provide a remedy for correcting potential problems.

This is where the learning by discovery that is enabled by interactive tools comes into play. In interaction with a live, force-



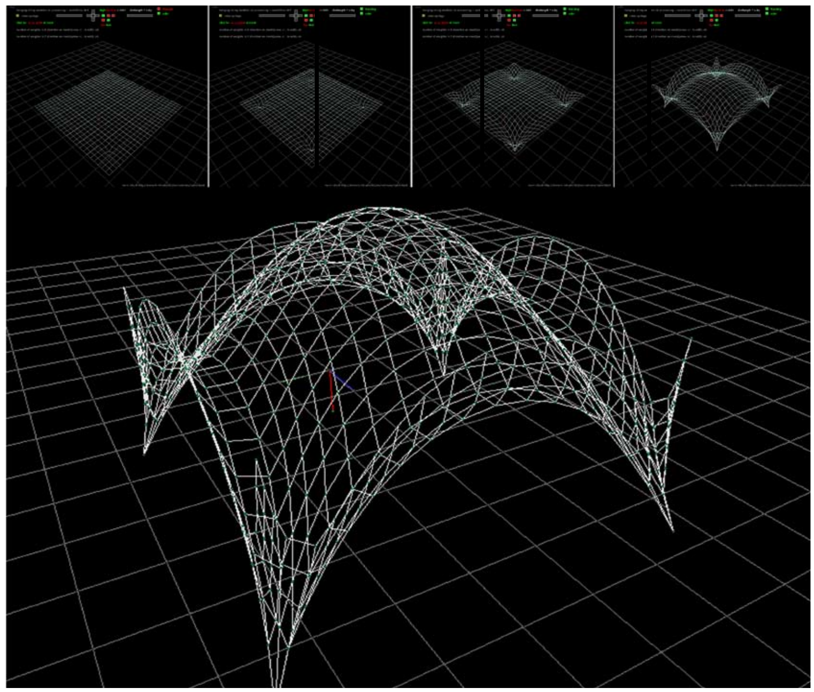
Wireframe cathedral sketches show how complex spatial arrangements develop quickly from simple topologies.

geometry linked structure, a designer can directly observe the range of structural responses while exploring possible forms. This encourages an explorative approach to design and supports unconventional solutions that integrate and respond to the designer's intent.

Structural and dimensional evaluation of form is not an afterthought but an essential part of the design process. Innovative structural solutions for shapes that are not limited to post and beam convention require innovative translation of the design intent. Structural behavior of complex form is hard to predict. Therefore, the ability for early structural feedback is important. Discovery of design in interaction with the tool rather than optimization of early design sketches. Iterative back and forth between design moves and a structural response allow for integration of the structural properties. Varying degrees of optimization – The design goal cannot possibly be driven by optimal structural shapes only – designer has a choice to go less efficient.

5.3.7.2 Optimization as a part of design

The notion of optimization as a design support tool is questionable

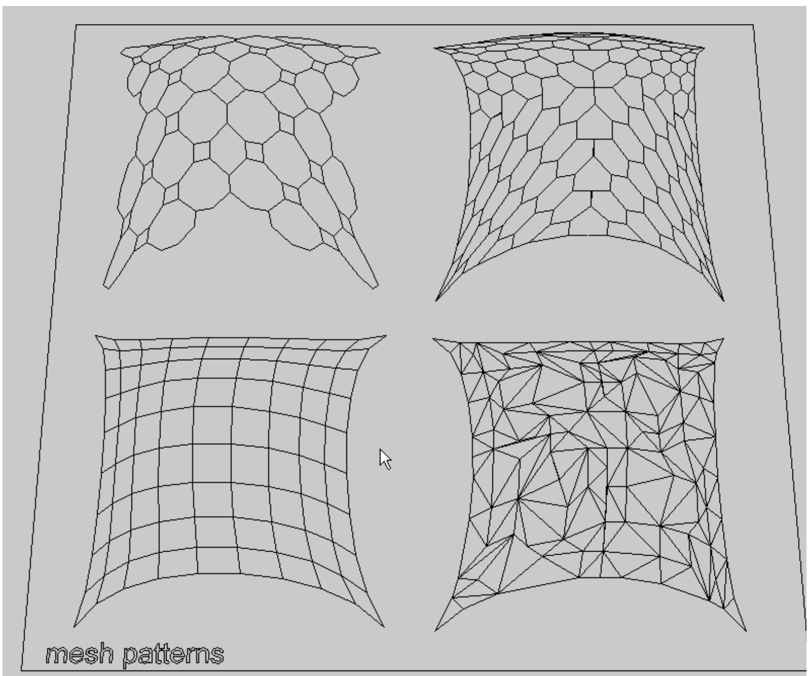


Mesh test to see whether Isler's edge condition is reproducible in the digital model.

if it remains unchecked. Initial design moves often are exploratory in nature and may or may not have an impact on the eventual outcome. To optimize an intermediary design move may cause paths to be abandoned or curtailed prematurely. It maybe more appropriate to speak of discovery rather then optimization. Design discovery could be defined as opening up potential design paths to the designer in the light of environmental influences acting on the design such as for instance gravity. The effect of such external parameters on the design may be mediated or weighted accordingly.

Optimization should not be the sole driver in design, as choosing an optimization objective is already part of the design choice. To choose a goal is to set a design process. Although it might seem like

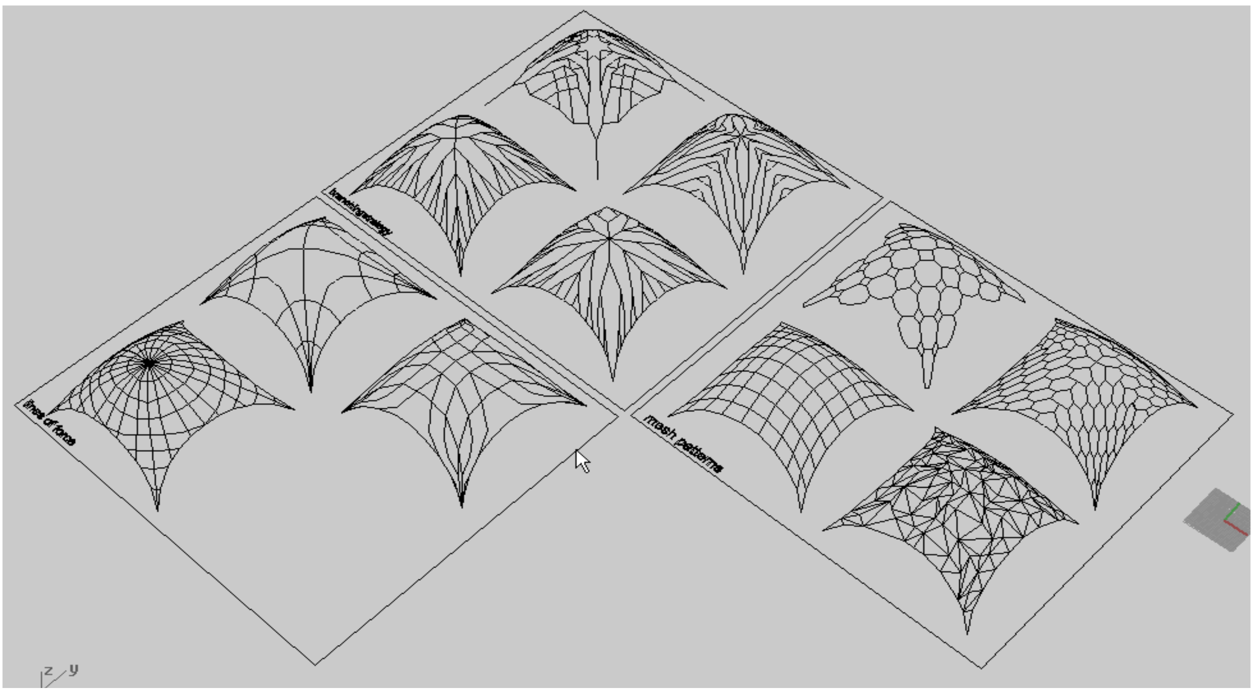
structural performance and material usage are out of the question for design considerations, some of the most radical differences in design approach constitute themselves in the differing positions on structural rational and material efficiency. What often is missing in design optimization is the variation of the starting premise.
Kristina Shea's Eifform is one of the few examples where the optimization is coupled to a generative technique, allowing the



Variations on topology shown to the left. Change in the topology is the next step in form finding. Kristina Shea's EifForm application is a great example of integrating several factors related to form and topology into a design explorer.
A change in the topology can have significant impact onto load paths in a structure. For instance, Nervi's roof structures closely follow the lines of force in his structures.

recreation of the structure topology based on the performance analysis. (Shea and Cagan 1997) Currently the form finding tool in development based on the workshop does some dynamic editing of the mesh topology based on tension and compression distribution in the structure. If a strut goes into compression it is eliminated. Over the course of the simulation only struts in tension remain. More research into the dynamic topology response is needed along the lines of Kristina Shea's research in order to make optimization a true component of design exploration and to allow for direct intervention by the designer.
As soon as the springs form a network that is more than a linear linked chain of springs and particles the structural behavior becomes more complex. No longer is there one unique load path.

Multiple paths between support and loads are possible and no single determined solution exists. Also, the behavior approximates that of a shell with increasing mesh density. An additional artifact of the simulation approach becomes apparent, as the spring approach does not ensure tension, only members. If the distance between two particles falls below the rest length of the spring the spring generates compression forces. Detecting the state of



Variations on load paths for the same hanging model surface. The patterns are modeled manually as quick sketch iteration of possible patterns.

the spring at any point and alerting the viewer of its compression state or alternatively removing it from the solution can avoid this problem.

In determining the thickness of shells that are approximated by grid meshes one can refer to Heinz Isler's work. Isler identifies instabilities in his shells as follows: First, at the supports second, due to general buckling; third, due to local buckling of the free edge (for which the counter curvature is so important); and finally, due to other modes. (Chilton 2000)

The author explored a less rigorous visual approach to testing the stability of a simulated form. By adding additional forces besides the vertical force of gravity, the structures can be triggered to sway. The relative amount of swaying of each particle is traced for the previous

200 positions. This allows a visual comparison of the displacement of the structure in different areas. Stable structures will show little, uniform displacements. Instability becomes apparent when traces vary widely. This approach could be developed further to integrate proper wind load and earthquake displacements. For now it is a simple, interactive way to explore a form for structural redundancy beyond one optimal load case.

5.3.7.3 Experiment 2: Load path

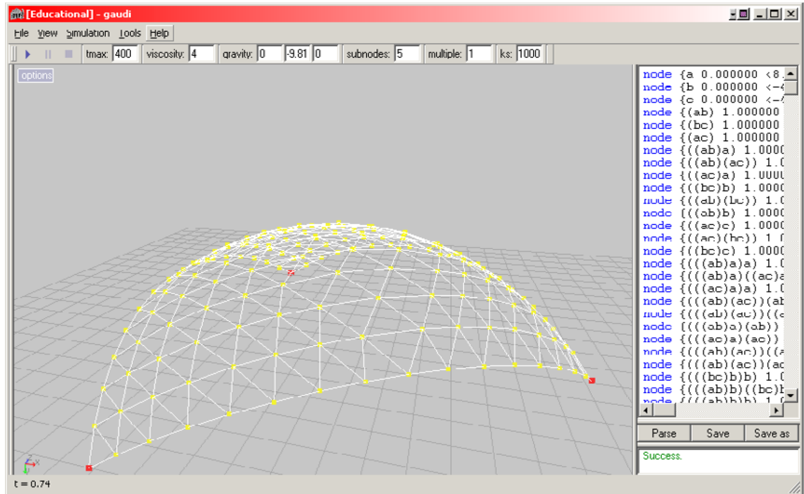
The load paths in a structure are very much dependent on the topology of the mesh and the geometry of the individual members. Often a topology change to avoid a singular dominant load path does much more to make a form more efficient then the geometric optimization of the starting topology. For instance, Isler was surprised to find that 90% of the loads in his shells are traveling into the supports in the corners of his shells. The edge supports receive only a fraction of the weight. This is due to the varied stiffness of the shell areas and subsequent variant resistance to loading. Therefore optimizing compression-only structures does not guarantee evenly distributed loads. In order to achieve uniformly distributed loads in a structure the thickness has to be adapted in accordance to the loads present after the initial form finding.

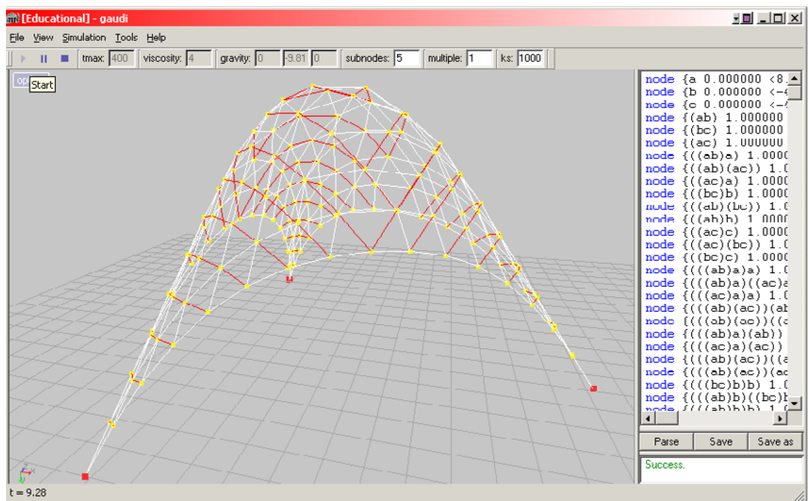
5.3.8 Topology Finding – The Next Step

In the particle spring system and real structural systems, the performance and dimension of a structure is integrally connected with its topology. Therefore, the system cannot stop at optimizing geometry but has to address the optimization of topology as well. Kristina Shea's Eifform's program is an example of such a process where both topology and geometry are subject to rule guided optimization with a semi-spherical target shape.

But finding topologies has to go beyond the variations of connectivity in fixed target geometry as it ultimately expresses the design goal as well. It leads to much larger issues of goals for design intent rather than structural performance alone. Performance can only be measured within a given framework, usually specified by the design guideline but novel topology might in fact challenge the starting conditions.

The mesh topology has a substantial effect on the form of the





The hanging model application that resulted from the workshop. The workshop was initiated by John Ochsendorf and co-taught by Barbara Cutler, Axel Kilian, Eric Demaine, Marty Demaine and Simon Greenwold.

The development of the application shown to the left was headed by Ryo Shimizu, a participant in the workshop. The application is written in C++.

The model shown here, by Barbara Cutler, is at first the current shape of the Kresge dome geometry as a mesh. The sphere patch is not an optimal structural form though and when subjected to the form finding process turns into the shape shown below.

The red members are members in compression.

Image: Ryo Shimizu and Barbara

structure and on the distribution of forces within it. The mesh topology fundamentally influences the performance of the structure. To optimize a structure cannot only mean to find the most efficient form for a given topology, but to find the most optimal topology for a given load case. This introduces the notion of topology finding or structure generation in addition to form finding.

5.3.9 Teaching – Hanging Model Workshops

The work on the hanging models was followed up with two workshop initiated by John Ochsendorf and co-taught together with the author, Barbara Cutler, Eric Demaine, Marty Demaine, Simon Greenwold.

The shown implementation of particle spring system is based on a particle spring library developed for the Processing (Fry and Reas 2005) environment by Simon Greenwold.

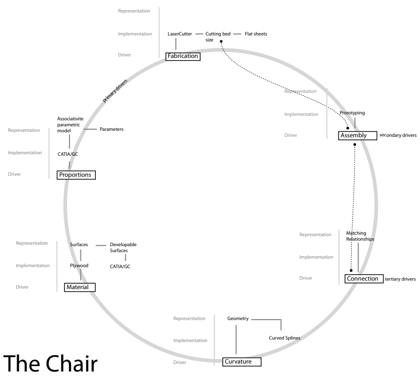
In the spring semester of 2004, instructors and students from civil engineering, architecture, computer science and computer graphics conducted a workshop to investigate the problem from an interdisciplinary perspective.

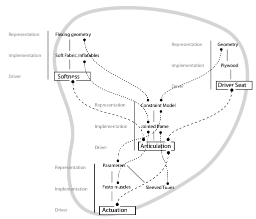
A robust and scalable implicit solver was implemented in C++ by the students. It allows the handling of larger particle spring meshes and faster processing times. The application was tested in its initial stages and will be applied in a design context in the near future. For validation purposes a spherical roof shell structure was modeled and evaluated in the software.

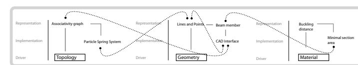
5.3.10 Conclusion

The goal of the experiment was to show how design focus could shift from the geometric objects to systems that produce and vary design beyond geometry. Some of the most intriguing design emerged from the intersection of strong constraints, often constraints that are contradicting. Those design conditions can not easily be dealt with by developing isolated design solution for each constraint individually; in fact a solution to an isolated constraint might be completely useless in the presence of another constraint.

Surprising results can emerge at the intersection of the conditions that need to be satisfied and the more complex the interactions become and the design object the less likely it can be dealt with fluently. A digital parallel constraint representation as in the hanging chain example offers the fluidity without reduction or abstraction of the elements involved. Such fluidity is not easily







The Chair

Material Fabrication
Proportion Connection Assembly

Circular Exploration
Fine tuning the constraint relationships

The findings of the circular explorations were published in ACADIA 2003 in "Fabrication of partially double-curved surfaces out of flat sheet material through a three-dimensional puzzle approach", Indianapolis, 2003.

The chapter three "Constraints in design exploration" will be published in the International journal of architectural computing (IJAC) in 2006 and is reproduced here with permission.

The chair project will be published in CADDRIA 2006., Japan.

The Athlete Car

Articulation Actuation Soft skin
Control

Branching Exploration
Establishing the constraints

The findings of the branching explorations will be published in game set match II, TU- Delft 2006,

The Hanging Model

Topology Geometry
Material

Parallel Exploration
Exercising well understood constraints

The findings of the parallel explorations were published in ACADIA 2004 as "Linking hanging chain models and fabrication", and in parts in the IASS journal 2005 article "Particle-spring systems for structural form finding" together with John Ochsendorf.

obtained as many constraints don't have direct mappings but at least provide an example of what could be accomplished in the ideal case.

The experiment lays out the potential for the integration of form finding techniques with fabrication strategies in a digital, integrated, expandable environment.

Form finding techniques in an interactive digital modeling environment can support the design process by giving continuous feedback to the designer, allowing the designer to integrate structural principles into the creation of form rather than to structurally optimize the finished form at the end of the design process. Digital simulation makes a range of numerical outputs of the form available for the generation of additional geometric information (like a building envelope) in response to the forces present. However, the approach to design should not a priori be driven by form finding or structural optimization. The design tools should always allow the designer to intervene and define design optimization principles.

It is certainly not the goal of the approach outlined here to promote a certain style or to create a "Gaudi design machine". The goal is to understand that expressing design goals as computational principles can support interactive design exploration and enhance the design experience. If the tools can provide a higher level of sophistication in doing so and enable handling complex competing design constraints in an interactive way, they will stimulate and challenge the notion of the design process in modeling environments today.

6 Results

In the following sections, the author summarizes the key result of the experiments conducted in the framework of this thesis: How to use constraints as design drivers in design exploration. He points to the areas where this approach could have the most effect, speculating on the challenges for future developments in the underlying conceptual framework of digital design environments, the structure of computational design environments themselves, professional practice and fabrication. The author strongly believes that it is time to move beyond the mere geometry-based design representations of today.

6.1 Constraints as Design Drivers

Constraints are usually perceived as limitations in design exploration. With computational models and the ability to model even complex dependencies their role maybe changing towards that of design drivers. The thesis showed several experiments that started from a very limiting premise and in some cases developed unique aesthetic or functional designs. The main reason constraints can become obstructions in design exploration is when the wrong choices for abstractions are being made. An abstraction that focuses on the suitable representation of constraints that allow the creation of constraint networks will be more successful than an abstraction that focuses on the reduction to the limitation. One simple example could be a height constraint. A fixation onto an absolute value will allow little exploration, whereas relating the height to a related entity responsible for the constraint will create constraint relationship that allows for more meaningful explorations. This is when constraint relationships can become formulations of design intent and eventually become design drivers. The constraint type can vary and go beyond the quantifiable, but the principle stays the same. The illustration of this translation is a key contribution of the thesis.

6.2 Types of explorations

The key contribution of this dissertation is the articulation of a novel approach to design exploration. Through the externalization of exploration processes, constraints have the potential to become design drivers that can be exercised through design explorers. Design explorers are the computational and conceptual constructs that capture the dependencies in a design problem externally.

Design explorers are constructed design specific. The first step is to diagram the constraints that play into defining the design problem. Through diagramming cross-dependencies and the type of exploration can be identified and constraint solvers can be developed, if necessary. The type of exploration for the definition of a design problem varies depending on the constraints.

Redescription of a design solution using different forms of representation aids in the generation of novel design, as each translation forces a reduction of the design on the shared constraints and produces a set of robust and overarching function and conceptual description that aid in design exploration. Through the experiments the thesis identifies three exploration types. They differ in their relation to the emergent design space of the exploration.

6.2.1 Branching exploration and design space

The first type is establishing a design space through a branching exploration. Through the addition of constraints a design space is progressively created starting from an initially undefined problem. The opening up of design space creates possible design variations.

6.2.2 Circular exploration and design space

The second type is a circular exploration. It either follows the branching exploration or the exploration begins from a well defined design problem from known constraints. The term circular refers to an exploration with a set of weighted constraints revolving around an established design space. It also refers to potential loops between constraints which propagate changes in one design representation to all others. This tendency to propagate and to create feedback loops makes this circular type of exploration particularly difficult to implement and fine tune.

6.2.3 Parallel exploration and design space

The third type is the parallel exploration, which has a set of known constraints and well defined relations between them but an open-ended design space. The variations in the design emerge from how the constraints are weighted against each other.

6.3 The three exploration types in relation to the constraints

6.3.1 Defining the constraints

If a problem is not well understood, then constraints need to be analyzed. Functional constraints allow for a component independent design exploration that, through the regrouping of different functions into new functional groups, can create novel design implementations. An example for this process is given in the writing device study as well as in the car studio for the creation of a seat-bucket-door combination.

6.3.2 Refining the constraints

For design problems that are well understood in terms of their constraints, a different type of exploration emerges. This exploration focuses on establishing the role of design driver among competing constraints. The product of the integration of constraints, implementation, and representation choices into an overall frame work is referred to in the thesis as the design explorer. The design explorer is an emergent entity of the exploration as much as it is the vehicle of the design exploration itself. The thesis experiment of the chair came about at the intersection of curvature, proportions, material, fabrication, and dependant assembly and connection constraints. The translation of abstract constraints into design representations and their implementations is a fundamental step in articulating the exploration. It eventually allows for a smoother form of design exploration through exercising the explorer.

In the chair example, the implementation choices play as important a role in the creation of the design as do the abstract drivers. It is important to note the aesthetic influence of the fabrication technique from the beginning of the process on both the overall design and the details. This is illustrated in detail in the experiment section in the design explorer diagrams.

6.3.3 Exercising the constraints

The third type of exploration occurs in the case of a set of well understood constraints where a computational representation exists or can be developed. The resultant design explorer can be interacted with as an entity embodied in the computational model that is more than the sum of its parts. The hanging model is the example given for this exercising of the design constraints in a seesaw like fashion between parallel constraints is the particle spring hanging model. The primary goal of the parallel exploration is the discovery of novel designs from exercising several constraints in parallel. If the cross dependencies are as complex as in the hanging model, abstraction of the design problem to make it graspable for a human designer is not an option. The digital explorer offers the possibility of a seesaw like shifting of the design driver from constraint to constraint in guiding the exploration. In the hanging models this means editing the topology, the geometry or the material properties respectively.

6.3.4 Discovery versus optimization

The implementation of a design explorer as an interactive entity opens the possibility of design as discovery from the exploration of the design form within the design space defined by the explorer. Innovative solutions may defy conventions. Optimization as design strategy poses the danger of measuring improvements in design against a conventional notion of the solution space. In fact the evaluation criteria of an optimization are as much part of the

design as the solutions that are being evaluated. Optimization assumes a set of stable evaluation criteria, design exploration as defined in this thesis explores both variations within the solutions to a problem description and variations of the problem description itself.
Most search and optimization processes have this shortcoming built into them such as genetic algorithms. An exception is the EifForm application by Kristina Shea (Shea 2000), which combines generative changes to the design with analytical evaluation of the designs. The optimization of the dome topology takes place under a set of force constraints, and in the following iterations alternative topologies are generated from rule based systems to compete against the initial optimum. This was accomplished out of the combination of several established engineering methods and could serves as a role model for future modular design explorers whose strength is derived from intelligent recombination of models from different domains into a design specific design explorer that achieves much more than any of the modules taken on its own.

6.3.5 Design is in the details

Variations of designs are infinite even for a well defined design problem but not any solution that satisfies the constraints is a good solution. As any designer will know the design is in the details as much as in the overall approach. Complexity and scale often freezes that design process at a stage when not all design relevant constraints have been integrated into the exploration. The exercise of fine tuning a design in the presence of all possible constraints is the art of innovation for a well understood problem. Innovation then is in the details and the formulation of a design explorer very helpful in externalizing and tracking the state of the overall design. The chair example proofed this hypothesis through the amount of development and integration and balancing of constraints was necessary to realize an only partially successful prototype from a straight forward geometric sketch.

6.3.6 Embodied constraints versus descriptive design

Choice is crucial in design exploration. It is all too easy to confuse the investment in effort and time into a design variation with its value. The two efforts are not related and generative explorations help to shift the focus on design on the conceptual and systematic level while increasing choice. In addition the choices are embodying the constraints. In the ideal case of a bidirectional exploration the products of the exploration can become the starting point of redefining the exploration itself.

The design explorers are domain specific and design specific and the types identified are prototypes that stand for a much larger variety of explorations. The term design explorer has been used before in the context of design exploration, for instance in (Gross 1984), but Gross referred to them as defining the relationship of components in the exploration of a design problem.

The definition of design explorer as a design specific construct made from the building blocks of design representations, design constraints and their respective implementations is a novel concept. It is novel because it takes into account an emergent trend of hybrid design descriptions in the context of design. The design processes are hybrid in their parallel existences of conceptual, implementation related, physical, and digital representations, all factoring into the design exploration. This poses a unique challenge for the creation of design explorers as exercisable constructs the designer can interact with. The author also suggests that conventional design software is not particularly well suited to play the integration platform for such a design exploration. The majority of CAD programs focus on geometry centered design representation with a minimal amount of higher level design description dividing the actual design representation in form of geometry. Parametric design systems at least offer a certain level of geometric and topologic variability but the overall rigidity of the design description prevents them from being good design explorers. The design explorers are as varied as the design problems that and they have to be redefined in parts every time a new problem is created. That does not rule out the use of existing or future conceptual building blocks that capture recurring conceptual and functional findings. Such building blocks, most notably the concept of simulation, which is at the core of virtually all digitally based endeavors, are introduced in the thesis in the initial introductory experiments.

6.4 Conceptual Building Blocks –

Simulation, Surface, System, Search

The dissertation has identified, analyzed and built upon four conceptual building blocks central to the formulation of design explorers. This provides the background for the discussion of further developments in computation and design representation.

Digital computation is shaped by the emergence of simulation, surface representation, systems and search. The history of the medium is ingrained in its conventions, knowing that history helps to work with and around those conventions and to expand the vocabulary of the medium.

In conclusion a summary of the four main concepts discussed in the introduction is given to point out possible directions to further develop them in the context of design exploration.

The first concept is simulation, at the heart of any design exploration. It is the abstract foundation of creating a representation that can be used to make predictions without embodying the entirety of what it represents. In computation, the concept of simulation has played a key role in pushing the conceptual and hardware development of what we view today as computers. The idea of simulation made it possible to exercise mathematical models for the sake of getting reliable predictions about much larger phenomena outside the limited scope of the machine. In terms of the development of visualization, the numerical world of early simulations quickly latched itself onto the feeble visualization capabilities of the emergent screen displays. Once the memory barrier had been broken, the pressure exercised by the directness of the visualization of a data set helped to fuel the development of visual systems.

Geometry was the first choice for visualizing numerical data. It was, quite frankly, the only choice available, since other output methods did not yet exist. Even the development of the screen was a long process and its outcome not at all obvious. But once the track for line-based geometry was chosen, it quickly prospered. With increasing machine speed and decreasing cost of memory, three dimensional geometries became possible. Surfaces were soon to succeed the wireframe geometries of the first decades. And another key building block in design representation of today had been introduced.

The dominance of surfaces in design representation can be directly traced back to the role of simulation. Simulation relies on images and surfaces. They provide, similar to points and lines, efficient way of visualizing three dimensional data. Accordingly, computational geometry for surfaces was developed in part to satisfy the increasing need for computational ways to describe complex objects like cars and planes (Bezier 1966). In fact, the automobile and airplane industries became the earliest clients of the nascent computer industry.

Surfaces still represent a major challenge for design representation today, and the latest increases in computational speed have increased their influence even further. Even the development of solid modeling could not break that dominance, as solid modeling still relies on watertight surface assemblies. Surfaces dominate digital design representation. This was less of a problem in the visually oriented output environment of computer graphics. But in the last decade, with the increasing availability of computer numerically controlled machines (CNC), the new challenges of translating surfaces in design representation linked to implementation in physical space have become apparent.

This leads to the third conceptual building block of design exploration: the concept of systems. In the history of digital computing and the history of systems, the complexity of computational problems and the changing computing tasks for the machines confronted the computer engineers of the 1940's with persistent problems in reconfiguring their hardware. In retrospect, the concept of stored program machines evolved surprisingly late after years of hard wiring machines. With the availability of program storage, however, the time was ripe for the creation of systems that were specific to a task, but could be quickly changed and adapted to a novel task. The engineers developed programming languages that captured low level concepts in reusable form. Today, software and programming languages are key building blocks of any implemented computational construct. But they have their limitations. The increasing inertia of any large, complex system makes it harder to accommodate fundamental change.

With the increased computational power and the vast increases in storing and sorting capability of data, the amount of accessible data multiplied exponentially and the fourth building block, the concept of search, became essential to deal with data. Anyone using computers today is familiar with the use of search engines like Google, an activity which has become an essential part of interacting with data. The concept of search as opposed to the use of previously defined data structures points to conceptual shift. As the needed data is unpredictable, it makes little sense to structure a data set a priori into organizational entities. Instead, categories are created by the process of searching, a concept that is related to emergent design features in design exploration.

Search, for instance in the form of genetic algorithms, is a powerful concept for design exploration. Tree data structures rely heavily on search for sorting and accessing information. In contrast, generative, rule based systems do not need search procedures, as any result of applying a rule is a valid member of the exploration set. However, the concept of search still applies to choosing and generating new rules.

In summary, the core concepts of design exploration identified in the thesis are simulation, surface, system and search. The author demonstrated their interdependency through their historic and conceptual development in computation. This provided a foundation for tracing their influence in the thesis experiments dealing with surface and material constraints, systems and the concept of search.

The discussion on these concepts also provides the background for discussing the implications of the experiments for the development and teaching of computational constructs. Is there a future for CAD software the way we know it now, Autocad, Microstation, ArchiCad, surface focused applications such as Rhino and Maya, or parametric systems such as CATIA and Generative Components? Beyond software what is the role of programming in the education of an architect, or more broadly speaking, the education of a designer? Has the increasing influence of the digital medium had an impact on how the design process is taught and conducted?

From the design experiments conducted in the framework of this dissertation, the author concludes that the clear distinction between computation as a conceptual approach and computation as an implementation cannot be upheld in the context of design exploration. Furthermore, the author predicts a shift away from geometry as the central design representation towards more varied design representation. Design explorers that incorporate design specific constraints and generative descriptions that go beyond the pure geometric representation of today will play a key role. Design explorers as agglomerations of design representations will become more hybrid incorporating performance digital models and physical explorations, functional and performative aspects.

6.5 Fabrication and Translation between Representations

Current fabrication machines have very little interpretive capabilities built in for pre-processing and translating the data that is given to them. If one compares the role of a crafts person in interpreting and reading the instructions produced by an architect with that of a CAD software instructing a CNC machine the crafts person might be less precise but augments and translates the abstract information in many ways into choices of tools and fabrication processes that are far more complex than any machine code. There is the advantage of the direct and precise translation of the geometry for the geometric specification into G-code for driving the cutter of a flat bed laser cutter. However there is no evaluation of the geometric data for validity of the cuts to be executed. Nor is there any representation for the functional dependencies of the geometry to be cut in case material or other parameters are changed. Responses are constrained to the machine inherent parameters as for instance the cutting speed and power of the laser but conceptual changes or even design changes the crafts person would be capable of applying his or her expertise are not part of the translation process. Ironically the initial role of CAD systems was the generation of G-code for CNC milling and not the generation of geometry. Geometry information was converted into digital form from blue prints initially (Farin 2002, Farin 2001).
The three dimensional printing company Z CORP© has developed more sophisticated support software for their powder based printers that actually checks the submitted geometry for water tightness and invalid surface objects. It can also repair certain flaws in the geometry and do printing specific adjustments to the geometry.
A step further would be machines that take any geometry and translate it into printable volumetric geometry, for instance starting from surface geometry. There could be attributes attached to

geometry similar to line weights and line colors defining preferred volumetric properties in the conversion. But other then that the translation into volumetric data would be fully dependant on the chosen scale, material and mode of printing. Such translation programs exist in parts but a general translation approach that is expertise based poses similar challenges as expert design systems do for architectural problems and there has been very little progress in resolving the artificial intelligence questions connected to this problem.

An example of such a fabrication specific translation of a design surface is given with the three-dimensional-puzzle surface and the three-dimensional-space truss. Both were generated from a NURBS design surface. The surface is the base input to two different generative scripts, which in one case produce flattened geometry strips with a puzzle connection detail for rebuilding the curved surface back up in three-dimensional space and in the other case produce a space truss. The truss is a structural, material saving way of translating a geometric surface with no thickness into a volumetric object.

6.6 Computational design environments

Computational design environments need to respond to this demand for integrating multiple representations by providing more transparency both in the underlying concepts and in the implementation strategies. With the current state of programming languages, it is hard to conceive of every designer as being trained as a programmer. However, there are developments, such as the processing environment (Fry and Reas 2000), where designers develop design specific abstraction layers for other designers that allow entering the realm of programming for design at different levels of complexity. Recurring models of computation are captured and provided as starting points, focusing on the removal of mechanical obstacles while keeping them conceptually flexible.

Other larger software development efforts, such as Generative Components by Robert Aish at Bentley Systems (Aish 2001), combine several approaches to digital design in a parallel structure. Visual and manual modeling interfaces are combined with script based associative topology representations. Script based design generation is created by the system from the design history. And last a user has the ability to expand the computational universe of the software from each of the interaction levels by adding components to the compiled runtime environment. Both provide an educational segway from visual graphical interactions to computational concepts by discovery. They also allow for the flexibility to interact with the evolving system at the appropriate level of abstraction. Generative Components is far from perfect as a design tool in terms of interface and computational languages. A key quality is that conceptually the environment is created so the program is evolving with the contribution of users. This is the first step in handing over control of the software environment and its features to the users rather than control it from the software provider's side. This step is essential in moving software development towards design specific development and is a radical departure from conventional software development cycles if it is followed through to its full extent.

While computational environments must continue to develop towards more transparency and accessibility for designers, the visual based representation of design will remain the most powerful approach of conceptualizing and generating design. Computational systems will have to develop to provide adequate support in this area. In parallel, however, the author predicts the departure from descriptive approaches in favor of generative approaches in design. Design geometry then is just an interim instance in a larger set of design representations governed by constraint networks and design drivers of various kinds.

In order to expand the type of possible design drivers in computational environments the performative disciplines linked to design, such as building technology and structural design, must push the development of approximate bidirectional design exploration methods to support design exploration. These will allow for the robust interaction with design variations based on quantifiable feedback. Current computational engineering solutions are overwhelmingly analytical in nature. In contrast to the proposed bidirectional design exploration methods, these analytical constructs calculate results based on provided design geometry. They lack the computational apparatus for allowing seesaw-like back and forth between analysis result and design geometry, where the role of driver and driven changes at will. Some progress has been made in research projects initiated by John Ochsendorf from MIT (Block 2005) (Kilian and Ochsendorf 2004), based on particle spring systems (Baraff and Witkin 1998).

In the domain of lighting analysis Julie Dorsey has developed several computational projects for such "inversed processes" for opera stage lighting and other settings. The problem of lighting design is complex and in many cases not computable for the lack of any one solution or too large a set of possible solutions. This is another instance where constraints as design drivers could play an important role. They could provide the crystallization points for dividing computational solutions up in sets around the constraints. For instance, the introduction of a ceiling for the calculation of a light source for a given light spot does not solve the problem. However it provides a constraint that divides the possible solution space in several subparts providing the designer with choices in the exploration of possible solutions.

One might start with a design goal and work out the design necessary to achieve this goal. Let us assume this goal is a patch of light in a particular location of a room. If there are no constraints present there are infinite possible constellations of light sources that may create such a light spot. An analytical approach would not yield a single answer as there are too many unknowns in the relationship between desired outcome and starting condition. This is though an ideal situation for a solver based approach as we already saw in the hanging model or the rigid body vehicle simulations. Solvers are designed to incremental move towards a possible solution closest from the starting condition.

Few such design challenges are free of context. Any additional constraints, such as the presence of a roof surface, provide additional input to the reversal of the goal-driver relationship. In addition, the problem can be structured qualitatively which does not reduce the number of solutions but gives them additional properties – for instance one could distinguish between parallel light sources and point light sources. For parallel light sources the patch outline determines the distortion of a light source or opening that masks the light source or constellation of occluding objects. So even without solving for a particular result one can introduce constraints that trigger design consequences and ultimately design decisions such as do we think of this light source as a point light or a parallel light. Is it a single source or are they constellation of different lights? How is the outline of the patch created? Form the masking of a bigger light source? Or through the composition of several objects or maybe the light is caused by an opening in an object? Essentially this process is similar to the function chain creation – a phenomenon is analyzed based on possible conditions responsible for its existence. The constraint constructs increasingly limit the possible sources for the light patch and add a qualitative nature to the problem that triggers a design response. At the core of this process is once the re description of a phenomena through many translations all centered at the same design driver – light in a particular spot on the floor.

Thereby what initially seemed to be an under-constrained problem without a single solution can in this way be turned into a solution space structured by design driver. This process suggests the revisiting of analytical processes in architectural design and other design disciplines. While analysis .is reactive, design explorers using bidirectional processes can be generative.

6.7 Design Process and Expertise

Instances of design explorers can be identified everywhere in practice. Frank Gehry's use of crumpled paper in sketching building designs can certainly be described as a design explorer. The paper is a physical embodiment of a constraint solver. The paper sheet has a set of responses and properties in response to applied forces. It is collaborative as it has a shared physical presence and the process can be repeated in a similar fashion easily. The paper properties act as a reference point in the office for the design language and its translation into building designs over the course of a project. This does not mean that Frank Gehry views his use of paper in that way. But the model could be used to support the argument for how powerful externalized design representation and constraints are even in the context of a seemingly irrational design behavior.

The results from the experiments suggest a rethinking of the way digital design tools and design processes are structured. The decade old focus on geometry at the center of the design representation in the digital design realm might not be adequate for today's digital processes and design process in general. The results of the experiments show that the cross dependencies that exist, even in a relative simple design project such as the chair, turn out to be of considerable complexity if externalized.

In the chair design experiment, geometry functions as the lowest common denominator for design processes in a collage of heterogeneous design representations. There are no predictable shared forms of representations that could be assumed by default. An exploration in the early phase of a design will be less predictable in its use of representation. The constraints and representations need to first be defined in parts through the exploration itself.

The formation of a design process can be compared with establishing expertise in a process. With the integration of more and more constraints into the design process or the exploration, the level of complexity increases and the exploration is better understood.

Another issue that has surfaced through the thesis experiments is the sequence of the design exploration process. The exploration diagrams were created after the experiments were completed, as introspection into what happened in the experiment. A detailed and resolved dependency diagram requires time to be created as well as insight into the process that may only develop during the exploration. Unfortunately, the diagramming process is unlikely to keep up with the associative brainstorming of designers. But the core idea of the exploration diagram is not subject to that limitation. The identification of a design driver played a key role in all the experiments even before the full exploration became clear.

6.8 Beyond Geometry - Conclusion

Judging the results of the experiments it is necessary that architecture overcomes the current focus on geometric, descriptive representation of design and moves into an area where the multiple factors influencing and constraining design influence and generate geometry.

Overall, the thesis calls for a rethinking of the way digital design is approached in the architectural design field and in design in general. The geometric model at the center of design representation may be slowly fading and should be augmented by generative descriptions of the design processes. Real world design projects with their highly complex interdependencies and heterogeneous discipline structure centered on the building task require a far more

Bibliography

Abelson, H., Sussmann, G. J., Structure and Interpretation of Computer Programs, MIT Press, Cambridge, Massachusetts, 1996.

Aish, R. Custom Object: A Model Oriented End-User Programming Environment, draft paper submitted to the visual End User Workshop VL2000, 2000.

Alexander, C., Notes on the Synthesis of Form, Harvard University Press, Cambridge, Massachusetts, 1964.

Aranda, B., Lasch, C., Pamphlet Architecture 27: Tooling, Pamphlet Architecture Series, Princeton Architectural Press, New York, 2005.

Banzhaf, W., Nordin, P., Keller, R. E., Francone, F. D., Genetic Programming, San Francisco, CA, 1998.

Baraff, D., Implicit Methods for Differential Equations, Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH), 2001.

Baraff, D., Witkin A., Large Steps in Cloth Simulation, in: Computer

Graphics Proceedings, Annual Conference Series, SIGGRAPH 1998, 43-54, 1998.

Baraff, D., Witkin A., Physically-Based Modeling, SIGGRAPH Course Notes, 1999.

Bathe, K.-J., Finite Element Procedures, Prentice Hall, Upper Saddle River, NJ, 1996.

Baudrillard, J., Simulations, Semiotext (e), New York, 1983.

Bentley, P. (ed.), Evolutionary Design by Computers, Morgan Kaufmann Publishers, San Francisco, 1999.

Bentley, P. J., Corne, D., Creative Evolutionary Systems, Academic Press, San Diego, 2002.

Bezier, P., Denition numerique des courbes et surfaces I, Automatisme XI, pp. 625-632, 1966.

Bezier, P., Essay de denition numerique des courbes et des surfaces experimentales, Ph.D. Thesis, University of Paris VI, 1977.

Bezier, P., Procedede denition numerique des courbes et surfaces non mathematiques, Automatisme XIII (5), 1968, pp. 189-196.

Billington, D. P., Thin Shell Concrete Structures, McGraw Hill, New York, 1965.

Blum, B., Beyond Programming: To a New Era of Design, Oxford University Press, New York, 1996.

Boehm, F., Evolutionary Design?, in: Arch+ 153, Berlin, June 2002.

Braitenberg, V., Vehicles. Experiments in Synthetic Psychology, MIT Press, Cambridge, Massachusetts, 1984.

Brooks, R.A., Intelligence Without Representation, Artificial Intelligence Journal (47), 1991, pp. 139-159.

Burrow, A., Woodbury, R., Pi-Resolution in Design Space Exploration, in: Proceedings of the Eighth International Conference on Computer Aided Architectural Design Futures, Atlanta, 1999, 291-308.

Burry, M., Machine-Readable Models to Interpret and Explain Gaudí's Use of Ruled-Surface Geometries, The Proceedings of CAADRIA 2002: Redefining Content, 2002

Busza, W., Cartwright, S., Guth, A. H., Essentials of Introductory Classical Mechanics, Pearson Custom Publishing, 2002.

Calladine, C. R., Theory of Shell Structures, Cambridge University Press, Cambridge, UK, 1983.

Campbell, M., Aspray, K. and W., Computer. A History of the Information Machine, Basic Books, 1996.

Carlson, C., An Algebraic Approach to the Description of Design Spaces, PhD Thesis, Department of Architecture, Carnegie-Mellon University, Pittsburgh, 1993.

Carpenter, R., The Logic of Typed Feature Structures: With Applications to Unification Grammars, Logic Programs and Constraint Resolution, Number 32 in Cambridge in Theoretical Computer Science, Cambridge University Press, Cambridge, UK, 1992.

Casteljau, P., Courbes et surfaces a poles, Technical Report, A. Citroen, Paris, 1963.

Cella, P., "Reexamining the Catenary", in: The College Mathematics Journal, Volume 30, Number 5, 1999, pp. 391-393

Ceruzzi, P., A History of Modern Computing, MIT Press, Cambridge, Massachusetts, 1998

Chilton, J., The Engineer's Contribution to Contemporary Architecture: Heinz Isler, Thomas Telford Press, London, 2000.

Cook, R.D., Malkus, D.S., M.E. Plesha, Concepts and Applications of Finite Element Analysis, John Wiley and Sons, 1989.

Coons, S. Graphical and analytical methods as applied to aircraft design. J. of Engineering Education, 37(10), 1947.

Csikszentmihalyi, M., Creativity, Flow and the Psychology of Discovery and Invention, Harper Collins, 1996.

Daniels, K., Low Tech High Tech Light Tech. Building in the Information Age, Birkhäuser, Basel/Boston/Berlin, 1998.

Day, A.S., “An Introduction to Dynamic Relaxation,” in: The Engineer, Vol. 219, 1965.

Diamond, J., Guns, Germs and Steel, W. W. Norton & Co., New York, 1999.

Dumitrescu, D., Lazzerini, B., Jain, L.C., Dumitresco, A., Evolutionary Computation, CRC Press, 2000.

Ervin, Stephen M., Intra-Medium and Inter-Media Constraints, in: CAAD Futures ‘91 Conference Proceedings, Zurich (Switzerland), 1991, 365-380.

Farin, G., Curves and Surfaces for CAGD, 5th Edition, Morgan Kaufmann Publishers, San Francisco, 2001.

Farin, G., History of Curves and Surfaces in CAGD, in: Handbook of CAGD, 2002.

Fry, B. and Reas, C., Processing web site, http://www.processing.org, accessed 12 January 2006.

Fry, B., Computational Information Design, Ph.D. Thesis, MIT, June 2004.

Fujiwara, J., Ohsaki, K., Uetani, K., Cutting pattern design of membrane structures considering viscoelasticity of material, in: Proceedings of IASS Symposium on Theory, Design and Realization of Shell and Spatial Structures, Nagoya, Japan, 2001.

Gero, J.S., Mt Brazier, F., Agents in Design, Key Center of Design and Computation, University of Sidney, Sidney, Australia, 2002.

Gershenfeld, N., The Nature of Mathematical Modeling, Cambridge University Press, New York, 1999.

Giancoli, D.C., Physics for Scientists and Engineers, Prentice Hall, Upper Saddle River, NJ, 2000.

Giralt-Miracle, D., Gaudí. Exploring Form, Space, Geometry, Structure and Construction, Planeta Publishing Corporation, Barcelona, 2002.

Greenwold, S., Allen, E., http://acg.media.mit.edu/people/simong/statics/data/index.html, accessed January 2006.

Gross, M., Design as Exploring Constraints, Ph.D. Thesis, Massachusetts Institute of Technology, Cambridge, Massachusetts, 1985.

Harada, M., Witkin, A., Baraff, D., Interactive physically-based manipulation discrete/continuous models, in: Conference Proceedings SIGGRAPH 95, pp. 199-208, ACM Siggraph, ACM 1995.

Hearn, D., Baker, P., Computer Graphics, Prentice Hall, Upper Saddle River, NJ, 1997.

Heyman, J., Equilibrium of Shell Structures, Oxford University Press, 1977.

Heyman, J., Structural Analysis: A Historical Approach, Cambridge University Press, 1998.

Horrowitz, P., Hill, W., The Art of Electronics, Cambridge University Press, 1989.

Hyman, A., Charles Babbage. Pioneer of the Computer, Princeton University Press, Princeton, 1982.

Isler, H., “New Shapes for Shells,” in: Bulletin of the IASS, No. 8, 1960.

Itten, J., Design and Form: The Basic Course at the Bauhaus and Later, Reinholt, New York, 1964.

Jefferson, D., R. Collins, C. Cooper, M. Dyer, M. Flowers, R. Korf, C. Taylor and A. Wang, Evolution as a Theme in Artificial Life:

The Genesys/Tracker System. In: Artificial Life II (C.G. Langton, C. Taylor, J. D. Farmer and S. Rasmussen, Eds.). Vol. X. of SFI Studies in the Sciences of Complexity. Addison-Wesley. Redwood City, CA. pp.417-434, 1992.

Kilian, A., Fabrication of partially double-curved surfaces out of flat sheet material through a three-dimensional puzzle approach, in: Klinger, K., Connecting >> Crossroads of Digital Discourse [Proceedings of the 2003 Annual Conference of the Association for Computer Aided Design In Architecture], Indianapolis, 2003, pp. 75-83.

Kilian, A., Linking hanging chain models and fabrication, in: Williamson, S., Beesley, P., Cheng, N., Fabrication: Examining the Digital Practice of Architecture [Proceedings of the 23rd Annual Conference of the Association for Computer Aided Design in Architecture and the 2004 Conference of the AIA Technology in Architectural Practice Knowledge Community],Waterloo, ON, pp. 110-125, 2004.

Kilian, A., Ochsendorf, J.O., Particle-spring systems for structural form finding, in: Journal Of The International Association For Shell And Spatial Structures: IASS, Vol. 46, pp. 77-84, 2005.

Kittler, F., Optische Medien, Merve Verlag, Berlin 2002.

Knight, T.W., "Color Grammars: Designing with Lines and Colors", in: Environment and Planning B: Planning and Design, Volume 164, pp. 417-449, 1989.

Knight, T.W., "Shape grammars and color grammars in design", in: Environment and Planning B: Planning and Design, Volume 21, pp. 705-735, 1994.

Kolarevic, B. ed., Architecture in the Digital Age: Design and Manufacturing, Spon Press - Taylor & Francis Group, New York & London, 2003.

Krzysztof, A., Principles of Constraint Programming, Cambridge University Press, Cambridge, UK, 2003.

Leach, N., Turnbull, D., Williams, C. (eds.), Digital Tectonics, Academy Press, New York, 2004.

Limming, R., Practical analytical geometry with applications to aircraft. Macmillan, 1944.

Loukissas, Y., Rulebuilding. Exploring Design Worlds through End-User Programming, SMArchS Thesis, MIT, Cambridge, Massachusetts, 2003.

Mahdavi, A., Mathew P., Kumar, S., Wong N.-H., "Bi-directional Computational Design Support in the SEMPER Environment", in: Automation in Construction (6), 1997, pp. 353-373.

Minsky, M., The Society of Mind, Simon & Schuster, New York, NY,1987.

Mitchell W.J., "The theoretical foundation of computer-aided architectural design", in: Environment and Planning B, Volume 2, 1975, pp. 127-150.

Mitchell, W.J., The Logic of Architecture, MIT Press, Cambridge, Massachusetts, 1990.

Negroponte, N., Being Digital, Vintage Books, New York, 1995.

ODE, www.ode.org, accessed January 2006.

Panofsky, E., Perspective as Symbolic Form, Zone Books, PLACE, 1997, originally published 1927.

Patrikalakis, N.M., Maekawa, T., Shape Interrogation for Computer Aided Design and Manufacturing, Berlin, 2000.

Press, W. H., Flannery, B. P., Teukolsky, S. A., Vetterling, W. T., Numerical Recipes, Cambridge University Press, Cambridge, UK, 1986.

Ramm, E., "Shape Finding of Concrete Shell Roofs," in: Journal of the IASS, 45(1), 2004.

Ramm, E., Schunck, E., Heinz Isler Schalen, Exhibition Catalogue, ETH Zurich, 2002.

Rasch, B., Otto, F., Finding Form, Axel Menges Verlag, Stuttgart, 1996.

Schek, H. J., "Force Density Methods for Form Finding and Computation of General Networks," in: Computer Methods in Applied Mechanics and Engineering, 1974.

Schlaich J., Schober, H., Freeform Glass Roofs, Structures 2005, Metropolis & Beyond, Proceedings of the 2005 Structures Congress and the 2005 Forensic Engineering Symposium, New York, New York, 2005.

Schlaich, J., Schober, H., "Eine veglaste Netzschale: Dach und Skulptur DG Bank am Pariser Platz 3 in Berlin", in: Bautechnik 78, H. 7, 2001, pp. 457-463, Verlag Ernst & Sohn, Berlin.

Schlaich, J., Schober, H., Verglaste Netzkuppeln, in: Bautechnik 69, H. 1, 1992, pp. 3-10, Verlag Ernst & Sohn, Berlin.

Schlaich, J., Schober, H., 2002, Geometrie Prinzipien für wirtschaftliche und effiziente Schalentragwerke, in: Bautechnik 79, H. 1, 2002, pp. 16-24, Verlag Ernst & Sohn, Berlin.

Schodek, D.L., Structures, Pearson/Prentice Hall, Upper Saddle River, NJ, 2001.

Schoeneman, C., Dorsey, J., Smits, B., Arvo, J., Greenberg, D., Painting with Light, in: Proceedings of the 20th Annual Conference on Computer Graphics and Interactive Techniques, pp. 143-146, 1993.

Serebryakova, K., http://www.uni-stuttgart.de/impulse/imp/alles.php?id=6, accessed January 2006.

Shaviv, E., Design Tools For Bio-Climatic And Passive Solar Buildings, Solar Energy, vol 67-4, p. 189-204, 1999

Shaviv E. A Method for the Design of Fixed External Sun-Shades. Build International (8). Applied Science Publishers Ltd, England, (pp.121-150), 1975.

Shea, K., Cagan, J., Innovative Dome Design: Applying Geodesic Patterns with Shape Annealing, in: Artificial Intelligence for Engineering Design, Analysis, and Manufacturing, 1996.

Shea, K., EifForm: a Generative Structural Design System, in: Proceedings, ACSA Technology Conference, MIT, Cambridge, Massachusetts, 2000.

Shelden, D., Digital Surface Representation and the Constructability of Gehry's Architecture, PhD Thesis, MIT, Cambridge, Massachusetts, 2002.

Sigmund, Ole 1994), Design of material structures using topology optimization, DCAMM Report S.69, department of solid mechanics, Ph.D. Thesis DTU.

Simon, H., The Science of the Artificial, MIT Press, Cambridge, Massachusetts, 1996.

Sismondo, S., "Models, Simulations and Their Objects", in: Science in Context 12, 2, pp. 247-26, 1999.

Stiny, G., "Weights", in: Environment and Planning B: Planning and Design, Volume 19, 1992, pp. 413-430.

Stiny, G., What is a Design?, in: Environment and Planning B: Planning and Design, Volume 17, 1989, pp. 9-103.

Stiny, G., March, L., 1981, "Design Machines", in: Environment and Planning B: Planning and Design, Volume 8, 1981, pp. 245-255.

Sutherland, I., Sketchpad. A Man-Machine Graphical Communication System, Ph.D. Thesis, Massachusetts Institute of Technology, Cambridge, Massachusetts, 1963.

Testa, P., O'Reilly, U.M., Kangas, M., Kilian, A., MoSS: Morphogenetic Surface Structure – A Software Tool for Design Exploration, in: Proceedings of Greenwich 2000: Digital Creativity Symposium, University of Greenwich, London, pp. 71-80.

Thompson, D.W., On Growth and Form, Cambridge University Press, Cambridge, UK, 1942.

Tomlow, J., Graefe, R., Otto, F., Szeemann, H., Das Modell / The Model, Institut für Leichte Flächentragwerke, University of Stuttgart, 1989. (Edition in German, English and Spanish.)

Tomlow, J., The Model, Information of the Institute for Lightweight Structures, No. 34, University of Stuttgart, 1989.

Turing, A. M., "Computing Machinery and Intelligence", in: A Quarterly Review of Psychology and Philosophy, Vol. LIX, No.236, pp. 433, 1950

Webster, www.webster.com, accessed December 2005.

Weisstein, E., Catenary From MathWorld, A Wolfram Web Resource, http://mathworld.wolfram.com/Catenary.html, accessed June 2005.

Wiener, N., Cybernetics, or control and communication in the animal and the machine, Cambridge, Massachusetts: The Technology Press; New York: John Wiley & Sons, Inc., 1948.

Wille, R., Concept lattices and conceptual knowledge systems, in: Computers Math. Applic., 23 (6-9), pp. 493-515, 1992.

Winograd, T. et. al., Bringing Design to Software, Addison-Wesley, Reading, Massachusetts, 1996.

Wolfram, S., A New Kind of Science, Wolfram Media, Champaign, Illinois, 2002.

Woodbury, R.F., Burrow, A.L., Datta, S., Chang, T.W., Typed feature structures in design space exploration, AIEDAM Special Issue on Generative Systems in Design 1999.

Zalewski, W., Allen, E., Shaping Structures and Statics, Wiley, New York, 1998.