Design Innovation through Constraint Modeling

Axel Kilian[1]
[1]*Massachusetts Institute of Technology, Cambridge, USA*
http://destech.mit.edu/akilian

Abstract. *This paper describes how constraint modeling can support design innovation. Furthermore, it lays out how constraints are employed in the construction and exploration of a model's design space. The paper places this approach within the larger context of design exploration using computational and conceptual representations of design. Four general constraint types are identified and examples from several workshops and design studios are presented for each of the constraint types. The examples range from product design to structural design to fabrication issues in architecture. Based on a review of the literature the most common constraints are of geometric, topologic, functional, and quantitative type. Based on the case studies the paper describes how the different types of constraints can be used as design drivers and help in the exploration of solution space. In conclusion the paper identifies the addition of bi-directional properties to constraint modeling as the next challenge in improving the application of constraint modeling in design exploration. Furthermore, the paper demonstrates the necessity to develop better constraint models for cross domain design.*

Keywords. *Design exploration, constraint modeling, parametric modeling*

Introduction

Constraints are generally viewed as limiting factors in design. But there is strong evidence in architectural practice and research that constraints can trigger the development of innovative examples and are a powerful way to drive solution space (Burrow and Woodbury 1999).

Constraints can help to focus design exploration and work within the boundaries of resources available. There are different types of constraints and constraints can be applied to different aspects of the design problem. A constraint on the use of a particular implementation may prove more limiting then helpful whereas a more open ended constraint on a functional description may trigger novel implementations. Constraints can provide the pressure necessary for innovative shifts.

In general four different types of constraints can be identified from the literature and research presented here.

Functional constraints
Topologic constraints
Geometric constraints
Quantitive constraints

The paper gives an example project for each of the constraint types in research conducted by the author. The examples are: First, a product design study looking at writing devices conducted in the concept car design class headed by William J. Mitchell, as an example of functional constraints as design drivers. Second, as an example of topologic constraints, a structural form finding implementation based on force equilibrium, which enforces tension only structural members for a given topology. It was developed in a workshop co-taught by John Ochsendorf, Axel Kilian, Barbara Cutler, Eric Demaine, Marty Demaine, and Simon Greenwold. Third, as an example of geometric constraints, a study of enforcing developable surface properties for the fabrication of approximations of double curved surfaces. And fourth, as an example of quantitive constraints, two projects based on searching the solution space of constraint models. One deals with an eight degrees of freedom articulated car design and the other with a family of vase designs.

Real world design problems are never isolated instances of one constraint or another. However, for this paper it is helpful to identify different constraint types and suggest possible strategies in modeling them. The larger challenge is to implement robust constraint models that allow for linking different constraint types and develop computational models for digital support of the exploration.

Constraints in the context of computational modeling

Modeling constraints computationally requires both a robust theoretical model as well as a working digital implementation. The paper shows some examples of successful implementations and several diagrammatic models. Solvers play a crucial role in constraint resolution and recent development in realistic cloth simulation for animated movies have pushed forward 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, most constraint design problems do not easily translate into an equation format. An important first step is the identification of the design problem. 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 new designs.

Identifying the design problem. Mapping the domain.

The definition of the design problem itself is a major step in setting up any constraint analysis. An example for the mapping of a design domain is given in the diagram example below. The task was to define new networking standards for the home networking market. 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. Here the graphing of the design problem reveals emerging properties on a visual level, but similar computational models have been developed.

Analysis of the constraints in the design problem

A description of a design problem involves analyzing the existing constraints and degrees of freedom for the unconstrained aspect of the problem. One possible approach is to identify the commonalities between a sample of existing design solutions to the problem. The example shown below follows this approach for the task of a writing instrument. Rather than choosing a component based approach a function-based approach was chosen. The function-based analysis allows for more flexibility in the description and more general description across solutions with very different form factors. A grammatical subdivision based on the parts or the form factor of the writing devices

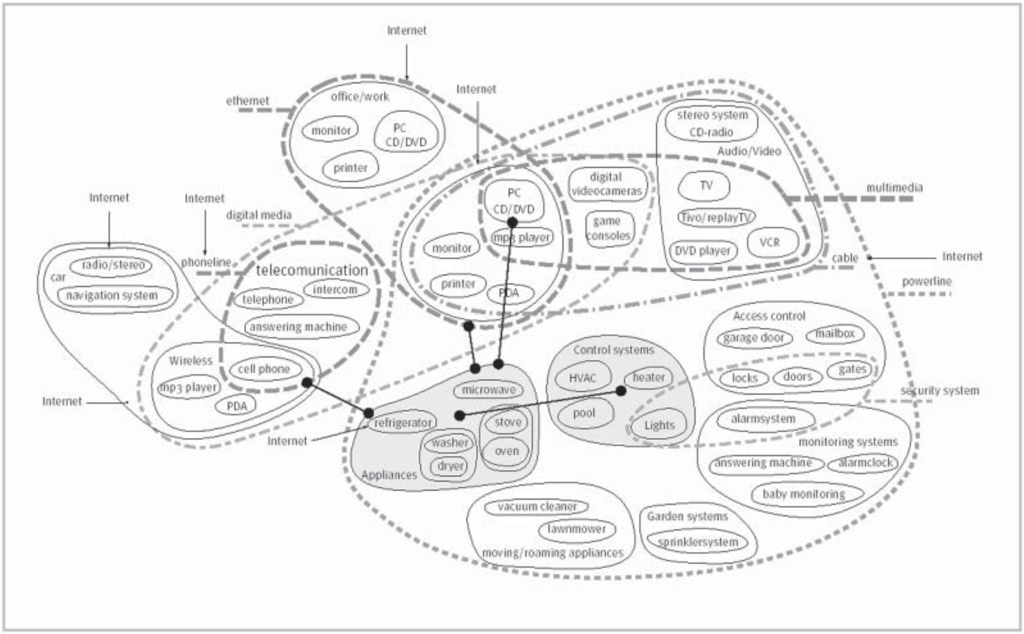

Figure 1. Home networking analysis - mapping a design space.

would be less successful in capturing the range of implementation from a single piece of chalk to a highly mechanized retractable pencil.

In the pen diagram example shows the different writing device functions coded by color. The first column shows the analysis of the exiting devices, from a chalk to a spray can.

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 for a design innovation. The goal is to break out of the established design examples with the help of an alternative description for the set of existing designs based on their shared functional constraints rather then their shared topological constraints.

The third column does exactly that. It provides a generalized functional description of the family of writing devices. It is based on five core functions of the set of studied writing instruments and chains them together based on their dependencies 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. The function tokens are replaced with choices for materials etc and step by step turned into a topology of parts that becomes a parametric geometric variation model for a new design instance.

The example is kept relatively abstract and it is not meant to provide an actual test scenario but rather emphasize the importance of the analysis of the design problem before setting up a constraint network in terms of parametric topologies or functions in order to allow for novel solutions. Parametric variation alone only covers a very small spectrum of the possible designs. The limitation of a parametric description is rather apparent but

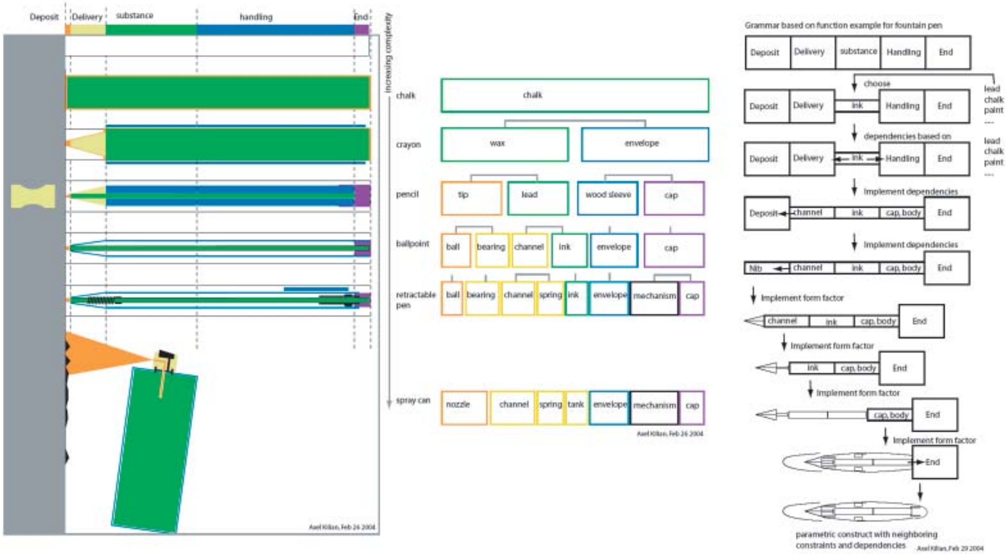

Figure 2. Writing device analysis in the context of the concept car design studio taught by Mitchell, W. J., Media Lab, MIT.

often neglected in current interest in parametric modeling.

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 implementation constraints but apply to an earlier design stage and have the potential to influence the design more dramatically then part variation alone.

Translating the constraints into design drivers.

The next step for using constraints to push a design forward is the translation of the constraints into design drivers. This means allowing for interactive variations of the non-constrained aspects of the design. The hanging model implementation demonstrates the use of a solver driven architecture based on particle and springs for tension only form finding (Kilian, Ochsendorf 2005)(Kilian 2004). In addition to adjusting the geometry to meet the structural constraints, the environment

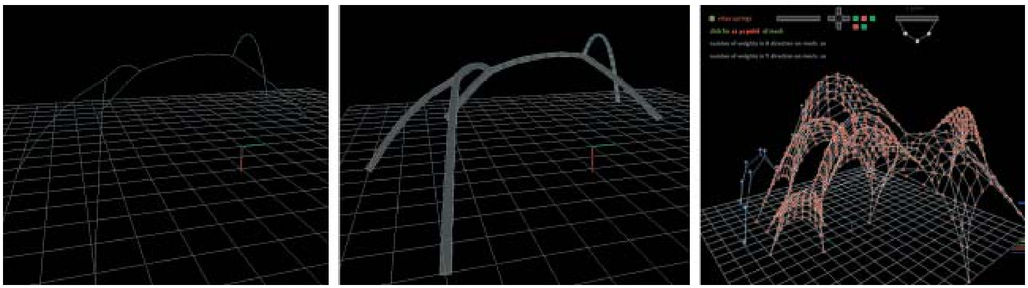

Figure 3. Topology of the hanging model. Linking the resulting forces to the structure envelope based on the material constraint. Mesh topology of the same environment, the constraint topology results in a geometric form

allows visualizing the approximate material envelope in proportion to the forces in the members. This links an additional constraint, in this case a maximum load bearing property of a chosen material, to the primary constraint resolution.

An Interactive model uses solvers to negotiate geometric constraints between geometric entities like lines and points. Properties are embedded in the behavioral model through the solver architecture.

Applying constraints to a problem domain

New constraints may also be applied to an existing design domain. This is, for instance, the case in architectural design with the increasing use of digital modeling in combination with CNC fabrication techniques (Kilian 2003). The fabrication techniques as well as the digital generation of the modeling information have their own constraints that apply to the design artifact. An example are fabrication techniques based on flat sheet material that have increased due to the availability of relatively inexpensive CNC machinery for cutting such as laser and plasma cutters. The developability constraint posed by the flat sheet material and the fabrication process has lead to a number of developments in the construction industry similar to developments decades earlier in product, ship, and airplane design (Shelden 2002).

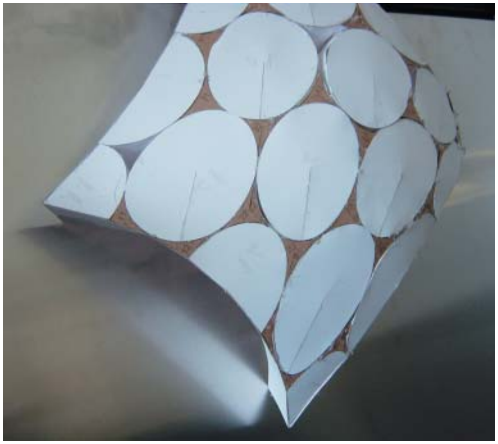

Figure 4. Cone based approximation of a double curved surface using only developable surfaces

At the core of this example stands the translation of free form shapes into developable surfaces using the developable primitive of the cone. The application of this primitive with the understanding of its limitations can produce interesting new approaches to low cost fabrication of free from 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 would control the spacing of the circles and their positioning similar to the MOSS example (Testa et al 1999) where a L-system approach is used to grow a surface or as in voranoid diagrams where the density of the points is determined by the local curvature as it was explored in a workshop at MIT by Michael Lehner and Charles B. Austin in the fall of 2004.

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 interesting aspect is that the circular base is double curved following the curvature of the surface whereas the resulting cone surface is strictly developable. The fabrication constraint is therefore embedded in the geometric property of the chosen primitive, which ensures the robustness of the approach.

The concept car design mentioned previously, explored novel car architectures with very little constraints applied to the process. One of the outcomes was "the athlete" a car that has many more degrees of freedom for steering then there are axis of movement in the plane it is driving on. This high number of degrees of freedom required constraint solving in the simulation of the behavior of the construct, which was conducted in CATIA using its three dimensional constraint solver.

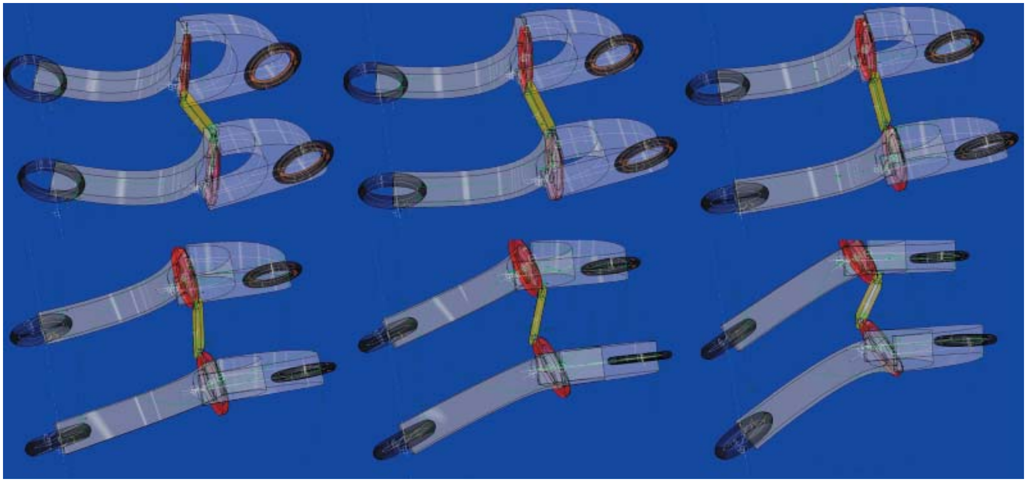

Figure 5. A three dimensional assembly for an articulated car with eight degrees of freedom driven by one parameter, the leaning angle. Thee remainder of the parameters are resolved through a constraint solver. Concept car design studio Media Lab, MIT– "the athlete" concept, Kilian, Joachim, Schmidt, Tan, Berrios-Negron, Gates-Spears.

The solver finds possible configuration of the eight degrees of freedom for a desired steering input and configures the parameters accordingly. Here constraints help to find a working parameter configuration for driving the vehicle.

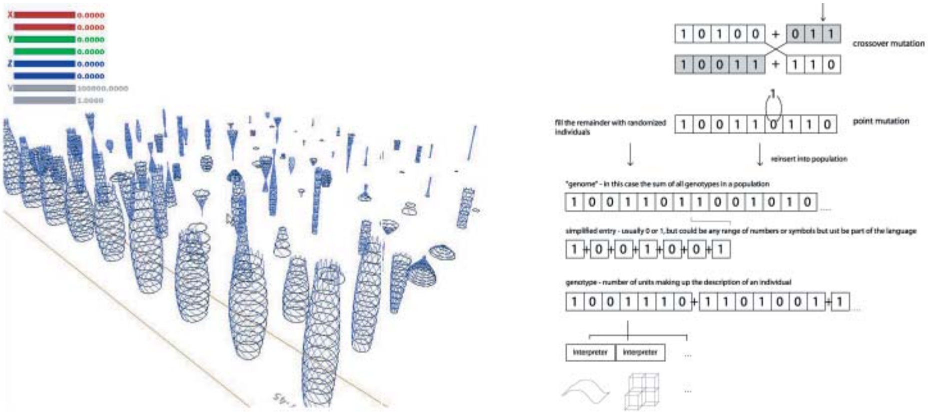

Figure 6. Genetic algorithm driven search for diagrid structure towers with an overall volume constraint

Exploring the solution space for a given 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 a target constraint that the search tries to satisfy.

For design exploration of lower dimensional parametric constructs, genetic algorithms often prove to be too cumbersome to set up or provide too little opportunity of user intervention 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

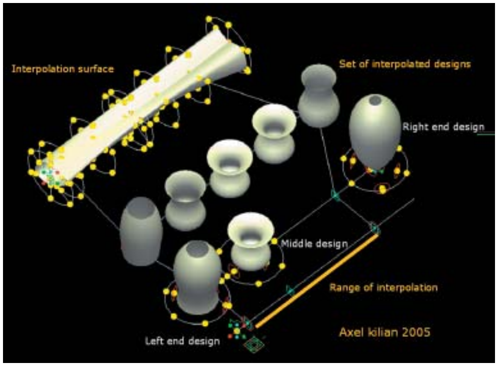

Figure 7. A vase controlled by six parameters. Three design instances define an interpolation surface that serves for the extrapolation of five additional design samples.

variations of the chosen values without loosing the context of the previous result.

Besides their ability of supporting dimensional variations, parametric models also model topologic constraints through their 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. The example shown here 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 design range

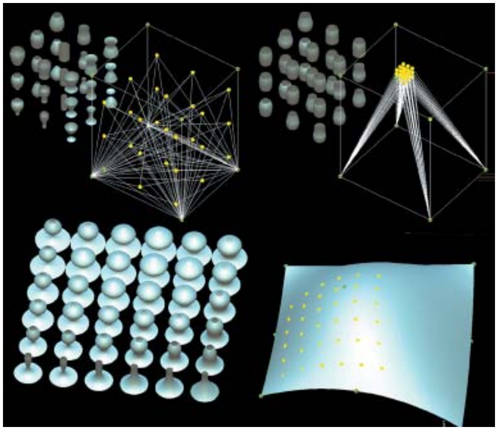

Figure 8. Mapping the solution space onto a parametric object. Exploration through sampling of the solution range via a point grid.

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 first example shows a family of objects whose parameters are each mapped to one point's XYZ value in a grid that samples a surface based on regular UV spacing. The moving of 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. As an alternative to a design surface one can use a design volume for the exploration. Each point is one possible sample within the exploration range. In order to increase the resolution of variations around one particular setting the sampling can be bundled around one parameter.

Conclusion

Types of constraints

In conclusion, the paper described a set of case studies in the use of constraints as design drivers giving examples for quantitive, geometric, topologic, and functional constraints. There are many more possible constraints, many of which are defined in the literature as for instance constraints between multiple media and representations (Ervin 1991). Gross (1985) gives a very complete account of the constraints in design as well.

The purpose of the paper is to demonstrate the heterogeneous nature of constraints and how using and modeling them made a difference in the case studies. Often formalism gets in the way of a design exploration and is not specific enough for the actual design problem in a studio or practice setting. The case studies show the relatively straightforward combination of a number of established computational principles such as genetic algorithms, parametric modeling, and graphs in support of design exploration. It would be ideal if modules that implement such computational models would be more easily accessible and combinable in the design environments in use today. Some computational environments are developing in this direction, most notably processing by Ben Fry and Casey Reas [http://www.processing.org].

A bigger challenge 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 could incorporate them all but improving the bridges between different isolated constraint models could improve the availability in design.

Bi-directionality of the constraint relationship is key in design exploration

To be truly supportive of design exploration constraint solvers need to be bi-directional (Mahdavi et al 1997). This means the constraint network cannot rely on a hierarchical structured dependency tree but must be organized in a graph fashion and use a bi-directional solver.

Constraint explorations are often only analytical in nature, meaning 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. A classic example of bi-directional exploration of a constraint network is graphic statics, where a force polygon is linked with a form polygon through geometric constraints and change can occur in both the form as well as in the form domain. This allows the exploration of the form or the force distribution respectively while each change in one representation affects the results in the other through the constraints.

References

Baraff, D., Witkin A., (1998), Large Steps in Cloth Simulation, Computer Graphics Proceedings, Annual Conference Series, pp 43-54.

Bentley, P. (ed.), (1999), Evolutionary Design by Computers, Morgan. Kaufmann Publishers Inc, San Francisco

Burrow, A., Woodbury, R. (1999), Proceedings of the Eighth International Conference on Computer Aided Architectural Design Futures, Atlanta, pp. 291-308

Ervin, Stephen M. (1991) Intra-Medium and Inter-Media Constraints, CAAD Futures '91 Conference Proceedings Zürich (Switzerland), pp. 365-380

Gross, M., (1985), Design as Exploring Constraints, PhD Thesis, MIT, Cambridge, MA

Kilian, A.: 2003, Fabrication of partially double-curved surfaces out of flat sheet material through a 3d puzzle approach, ACADIA 2003, Indianapolis.

Kilian, A., (2004), Linking hanging chain models and fabrication, ACADIA 2004, Cambridge ON, Canada.

Kilian, A., Ochsendorf, J. O., (2005), Particle-spring systems for structural form finding, Journal Of The International Association For Shell And Spatial Structures: IASS, Barcelona.

Mahdavi, A., Mathew P., Kumar, S., Wong N-H.(1997), Bidirectional computational design support in the SEMPER environment, Automation in Construction (6), pp 353-373.

Mitchell, W.J. (1990), The logic of Architecture, MIT Press, Cambridge MA.

Shelden, D., (2002) Digital Surface Representation and the Constructability of Gehry's Architecture, PhD Thesis MIT, Cambridge, MA

Zalewski W., Allen, E., (1997) Shaping Structures, John Wiley & Sons