Parameters Tell the Design Story: Ideation and Abstraction in Design Optimization

Erin Bradner[1], Francesco Iorio[2], and Mark Davis[1]

[1]Autodesk Research
One Market, Suite 200
San Francisco, CA, 94105 USA
{erin.bradner, mark.davis}@autodesk.com

[2]Autodesk Research
210 King E, Suite 500
Toronto, ON M5A 1J7, Canada
francesco.iorio@autodesk.com

Keywords: Design Optimization, Optioneering, Ideation, Architectural Design, Engineering Design, Qualitative Research.

Abstract

We report qualitative findings from interviews and observations detailing how professionals generate and evaluate design ideas using design optimization tools. We interviewed 18 architects and manufacturing design professionals. We frame our findings using the Geneplore model of creative cognition and classify examples of ideation and abstract design thinking arising from optimization workflows. Contrary to our expectations, we found that the computed optimum was often used as the starting point for design exploration, not the end product. We also found that parametric models, plus their associated parameters and simulations, serve as an alternate, highly valued form of design documentation distinct from engineering schematics.

1. INTRODUCTION

We focus our research on optimization in a professional design practice. We select this focus because computational power that was once restricted to large government entities has become increasingly accessible to private entities (Thibodeau, 2013). Meanwhile, the professional use of parametric modeling has provided adaptive structures against which iterative simulations can be run. We've observed that computationally intense, heuristic searches of design spaces are becoming more commonplace in professional practice than in the past.

The proprietary nature of professional design processes, and their resulting designs, led us to choose an ethnographic approach to our research. While the use of simulation tools has begun to be systematically documented in the literature (e.g. Tsigkari et al. 2013), to our knowledge there are no published, comparative accounts of goal-driven design and optimization used professionally. The qualitative findings reported here begin to fill this gap.

As software designers and high-performance computing experts, the high-level objective of our research is to expose the opportunity for new or improved computing architectures and user interfaces for generating, exploring and describing design spaces via optimization. In this study we advance toward that objective by first detailing how professional designers use design optimization to ideate – i.e., to generate and then explore solutions, to discover new and unexpected ideas, and to focus or expand their understanding through data visualization. We then articulate the multiple levels of abstraction engendered by optimization workflows, including: problem definition, evaluation, coding and documentation.

2. DESIGN OPTIMIZATION

Design optimization tools are creation tools that use parametric modeling, performance simulation and mathematical optimization to systematically generate and evaluate design alternatives (Holtzer et al. 2007). Design optimization, also known as design optioneering (Gerber et al. 2012) and computational design (Arieff 2013), is a departure from traditional architecture and engineering practice. Typically, architects generate a relatively small set of design alternatives that represent specific points in a multi-dimensional design space (Flager and Haymaker 2007). In architecture, this small set of design alternatives may be communicated in the form of two or three laser physical cut scale models or a few dozen digital photo-realistic visualizations. Even with the support of state-of-the-art computer-aided design tools (CAD), individual designs are iterated relatively slowly and with considerable design effort. (Ibid.)

Conversely, architects and engineers using design optimization practices generate orders of magnitude more design alternatives by specifying design objectives in the form of design parameters and parameter ranges (Tsigkari et al. 2013). They use stochastic search methods, such as genetic algorithms, to automatically and iteratively compute large sets of design alternatives (Holzer et al. 2007). The designs that best fit the architects or engineer's predefined acceptance criteria survive multiple generations to spawn successive generations of unique, new designs.

Contrasting with traditional design practices, optimized designs are computed parametrically and bred algorithmically. The numerous design alternatives that are produced are often represented by a multi-dimensional plot of solutions and might be coupled with a matrix of thumbnails of rendered designs, as in Figure 1. Researchers investigating the approach argue that design optimization enables designers to "more efficiently, and with more certainty, explore complex and tightly coupled design solution spaces" (Ibid.) than traditional design practices.

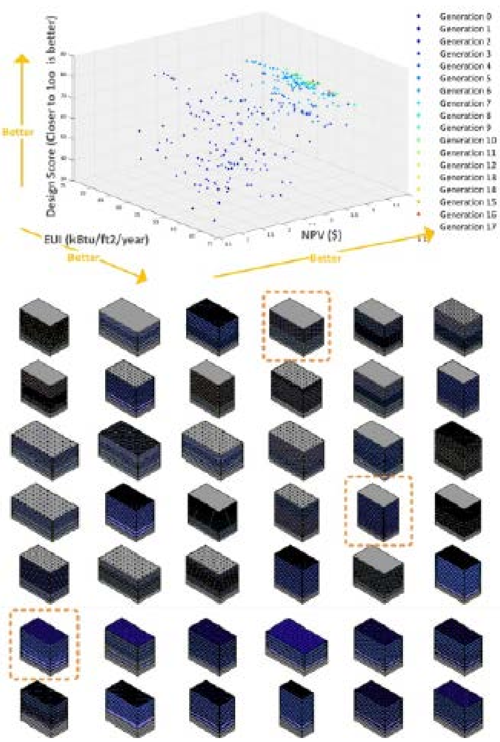

Figure 1. Sample Design Optioneering Results from (Gerber et al 2012).

2.1. Design Optimization as Creativity Support

Design optimization tools are an under-researched class of creativity support tools. Professional engineers and architects perform what we dub *constrained creativity:* their creative ideas must perform an intended function, and satisfy specific performance criteria (Shah et al. 2003). These professionals concern themselves with how to generate and explore ideas that are optimized against multiple, competing objectives, variables or constraints. In building design, for example, it is necessary to simultaneously consider multiple complex objectives including site utilization, structural design, building form, energy use, buildability, and operating costs.

To the extent that design optimization tools enable users to be "not only more productive but also more innovative" (Shneiderman et al. 2006), they fall into the application category of creativity support tools. Creative activities that lead to innovation include idea generation, easy exploration, rapid experimentation and fortuitous combination. (Ibid.) Designers using design optimization tools combine ideas through the algorithmic generation of design alternatives, they experiment through adjusting design parameters, and they explore through examining data visualizations of solution sets. Optimization tools do not simply automate idea generation. They support the creative process of idea generation, exploration and refinement.

2.2. Design Optimization and Creative Cognition

Engineering and architectural design are inherently generative disciplines. The set of design algorithmically computed alternatives for any given real-world design task is vast. Flager et al. calculate that for the one room, steel-frame building used in their energy and structural optimization study there were $55x10^6$ possible solutions (Flager et al. 2009). Cognitive limitations prevent humans from imagining even a small fraction of the possible alternatives in a problem with high dimensionality and millions of solutions. Not even the most skilled designer can handle this level of mental complexity. Variable interactions are particularly difficult to imagine—how do window size, glazing type and building orientation interact to produce the most energy efficient building? Furthermore, time constraints prevent designers from exhaustively exploring the solution set for top performing solutions.

In our qualitative research we observed that both architects and engineers explore a wide range of design parameters and constraints by applying iterative design techniques to "solve" for their designs. Geneplore is a heuristic model of creative cognition (Ward et al. 1996) that we found particularly useful in framing the design processes we observed. We will briefly describe this model now. Figure 2 shows the basic structure of the model.

The Geneplore model involves three fundamental cognitive processes: generate concepts, explore and interpret concepts, and evaluate the problem to focus or expand the concept. These processes map nicely to the design optimization process in architecture and engineering whereby designers use goals and parameters to abstract the design problem, then generate concepts. Together with the design optimization system, they explore the strengths and weaknesses of the candidate solutions, and then use insights gained from both the generation and exploration activities to refactor the design problem. Meanwhile, design constraints, such as building height, are introduced which reshape the problem definition and also reshape how concepts are interpreted.

A key contribution of the Geneplore model is the notion of a preinventive structure. The term preinventive is used to denote a germ of an idea, a “half-baked” sketch, or a design hunch that may hold promise. Patterns, 3D models and conceptual combinations (Ibid.) that prefigure creative concepts are all examples of preinventive structures.

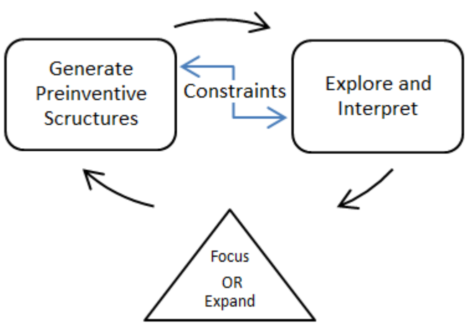

Figure 2. Geneplore Model from (Ward et al. 1996).

We acknowledge here that it is unusual to map a cognitive model like Geneplore onto a system model such as design optimization. Nevertheless, we were compelled to do so because we found ourselves drawing heavily on the Geneplore model to inform the analysis of our interview findings. We present our interview study and data analysis next. We contend that that preinventive problem statements, conceptual combinations and constraints are the primitives on which design optimization operates. We will also explain that idea generation, exploration, and adjusting the problem focus – the three pillars of Genepolore – are the three most common activities performed by users of design optimization tools.

3. INTERVIEW STUDY

We interviewed 18 design professionals in the fields of architecture (15) and manufacturing/fabrication (3). Interview duration ranged between 1.5 and 3 hours. The majority of our interviews were conducted at the participant’s place of work, though 5 (28%) were conducted over the phone with screen sharing. Interviews were recorded (audio only) and transcribed. Due to the proprietary nature of the designs produced by the workplaces we visited, photos were prohibited and we weren’t allowed to take away design artifacts. We coded the interviews and observations, applying a grounded theory approach (Strauss and Corbin 1998). Here we present a *thick description* of the workflows that professionals follow when conducting design optimization.

We knew enough about design optimization tools at the outset of the study to reject the hypothesis that it is used solely to conclude the design process by selecting the winning design from a set of all plausible alternatives. Through prior contact with architects and engineers we learned that it was being used by many at the start of the design process—at a stage called design conceptualization. This puzzled us. How was a tool that was designed to computationally *solve* design problems being used to *question* and *explore* design problems? We wanted to understand how and why an engineering technique that emerged from NASA (Schmit and Thornton 1964) to compute the single highest-performing design for an airfoil was being used by architects to compute the quality of penthouse views in design alternatives for a building in Bangalore (Tsigkari et al. 2013).

3.1. Analysis

In the analysis below we first describe the ideation process we observed. We briefly detail how professional designers use design optimization to generate and then explore solutions, to discover new, and sometimes unexpected, ideas. We also describe how they focus or expand their understanding through data visualization. We then articulate the multiple levels of abstraction: problem definition, evaluation, coding and documentation. Design insight and improved design quality are two examples of the value gained from design optimization. Yet abstractions are also the primary source of the user experience challenges, such as sufficiently understanding statistical correlations between design variables in order to reduce or expand the dimensionality of the design problem.

3.2. Ideation

Design ideas from architecture and engineering are the product of creative ideation whereby ideas are generated and evaluated for suitability using qualitative and quantitative evaluation criteria (Shah et al 2003). As indicated by the Geneplore model (Ward et al. 1996), there are two distinct activities: generation and evaluation. The traditional ideation process is iterative (Flager and Haymaker 2007). When design optimization techniques are applied, the design process becomes less iterative yet more design candidates, by several orders of magnitude, are produced. One participant described the difference in the two processes in these words:

"The typical design workflow is to design then throw to the analyst. Redesign. And then keep playing catch. It's inefficient. [Design optimization] captures the criteria that are important to you then have the cloud process all the permutations."

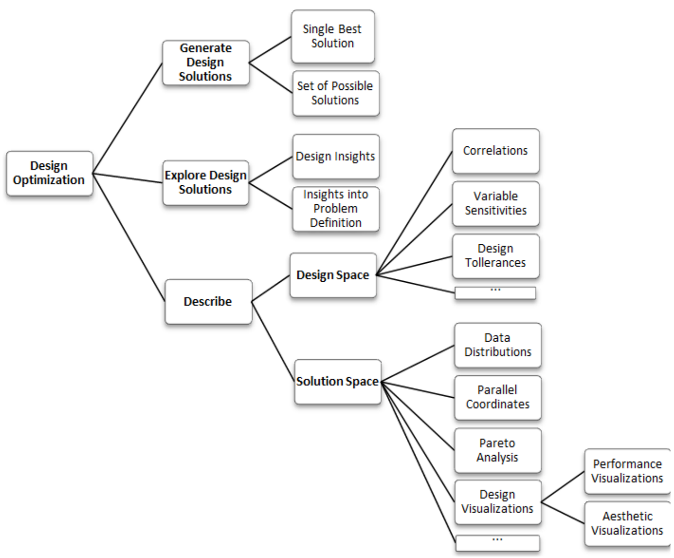

Figure 3. Uses of Design Optimization.

The starting point in the design optimization process thus shifts from specifying geometry in a CAD system:

design → evaluate → select or redesign

…to defining the design problem:

define → generate + explore → select or redefine.

Here we'll outline the activities of generating, exploring, describing, and selecting solutions, as summarized in Figure 3.

3.2.1. Generate Solutions

Architects and engineers often cannot compute nor comprehend the effects of combining multiple variables in a complex system. Design optimization tools procedurally generate design solutions from a machine-readable definition of the design constraints and parameters. Typically, the design optimization system returns a large number (100s to 1000s) of design solutions. One study participant explained this automated idea-generating process thusly:

"Good design has inspiration to it...if you have that vision you can encode it and parameterize it and explore it further. Now we have a rich flora of options."

Examples of designs that our participants optimized include high-rise buildings, hospital patient care wards, an engine manifold, and installation art. The elements of the design that were optimized include, but are not limited to: aesthetic form, structural form, mechanical performance, human performance (nurse walking distance) and building energy efficiency. *Liters per second* is an example of a design parameter from the engine manifold example. Engineering constraints are such things as "*no more than 2 bars of pressure drop.*" An architectural constraint might be "*must maintain an average of 40% natural light.*"

3.2.2. Explore Solutions

We appropriate the term *explore* from the Geneplore model of creative cognition to describe the way users purposefully adjust their parameter ranges and review the resulting solution sets. One user referred to this as an act of discovery: "*The key is the discovery phase. As you adjust ranges, you discover options you may not have expected.*" Surprisingly to us, both architects and engineers in our study said that the computed optimum was often used as the starting point for this exploration. In one case, the structurally optimal alternative design for a high-rise building was used to seed the exploration process for its aesthetic design. The structural engineer from this group explained that "*the optimum is a way to talk about form*" indicating that the solution they chose from the set of high-performing solutions in their structural optimization served as the conceptual sketch on which the aesthetic designers could iterate.

3.2.3. Describe the Solution Space

The solution space is the set of all solutions computed by the design optimization algorithm. We observed two primary ways that design optimization describes the solution space for users. These are visualizations and data plots. Visualizations can be further classified into performance visualizations (a.k.a. simulations) and aesthetic visualizations. A heat map showing regions of stress on a 3D model of a metal bracket under a simulated load is an example of a performance visualization. Our study participants explained that they use visualizations to develop their understanding of the impact of their design choices on aspects of performance. In one case, a mechanical engineer optimizing the design of a manifold attributed a major design insight to a computation fluid dynamic visualization he reviewed:

"[the visualization] gave us the insight to open the diameter… the diameter of the opening could not be expanded but the diameter of the shaft cold be flared to increase flow."

Aesthetic visualizations are also 3D renderings of design alternatives. When used by architects, these visualizations allow systematic exploration of building form. These visualizations are important since, as one structural engineer put it: *"slight variations to form, to a designer's eye, are either elegant or fat."*

Data plots, commonly in the form of scatter plots, are another type of data visualization we observed being used. Respondents expressed a range of opinions about the utility of different types of plots. One study participant sung the praises of parallel coordinate charts while another pledged his allegiance to Pareto plots. A manager said this about the computational designers he manages and their discussions of different plot types: *"they talk about tradeoffs in a spiritual way."* We suspected that different plot types allowed certain designers to extract more information about design tradeoffs more quickly than others. We theorized that there was an interaction between the sophistication of users' statistical knowledge and the utility they assign to the different types of plots.

3.2.4. Describe the Design Space

The term design space describes the mathematical definition of the design. It includes the design variables, and constraints; the ranges or discrete values assigned to those variables; and any other bounding criteria that can be expressed mathematically. Study participants used statistical analyses such as correlations to examine the interaction between variables. They actively sought these statistical descriptions of the design space because *"understanding dependencies between different systems [variable types] is very challenging"* and because statistical analysis gives them insights to add or remove variables when warranted. For example, they remove variables found to be highly correlated to reduce the "dimensionality" of their problem space; they split variables based on variable sensitivities; and they adjust parameter ranges up and/or down to account for design tolerances.

3.2.5. Select Solutions

Something we found counterintuitive was that many participants in our study selected the highest performing solution from a solution set to initiate their design ideation process, rather than to conclude it. This remained befuddling until we learned that our participants were using computational design more frequently in the earliest, most conceptual phase of design than in any other phase. They considered the optimum to be the computational equivalent of a back-of-the-napkin sketch. Borrowing again from the Geneplore model of creative cognition, the computed optimum is the pre-inventive structure that seeds the ideation process: *"instead of starting with nothing, you start with something…your optimum gives you a starting hunch."*

That said, it was not only the single highest solution that was important and useful, but often the full set of high performing solutions. Our study participants reported consulting Pareto plots iteratively in the conceptual design phase to rapidly identify and select "interesting" solutions. Pareto plots are a type of scatter plots commonly used to distinguish high-performing solutions in the set of all feasible solutions. The Pareto set is composed exclusively of the high performers: for any given solution in the Pareto set, it is impossible to increase performance along one axis without decreasing performance along another axis. A mechanical designer qualified his interest in identifying the highest performing designs during the conceptual design phase by explaining that he is looking to determine *"what direction the performance is trending."* He iteratively generates high performing solutions using visualizations such as Pareto plots to explore how adjusting the parameters by hand affects the overall quality of the resulting solutions.

3.3. Abstraction

To define a design problem requires abstracting the problem into mathematical descriptions such that the design optimization tool can compute alternatives. An empirical study of engineers has shown that the level of abstraction and precision in a problem definition affects the quality and quantity of ideas generated (Fricke 1999). Problems defined with low precision produce few and poor solutions because engineers fixate on concrete solutions too early. Conversely, too many solutions also produce poor solutions because managing too many solutions diminishes the engineers' ability to identify, evaluate and modify the best solutions. Successful designers balance their search for solutions. (Ibid.)

We did not evaluate the quality of design solutions our study participants produced by design optimization tools; however, our interviews with architects and engineers suggest that design optimization tools maximize the positive effects of precision in problem definition while minimizing the negative effect of large solution sets. Furthermore, we suspect that the demands of abstracting the problem into a precise definition, which our participants stress is nontrivial, improve the designers' ability to evaluate solutions. We discuss multiple levels of abstraction in the design optimization workflow next.

3.3.1. Problem Definition as Abstraction

"Think about the underlying logic and how you can transform it."

Using design optimization tools necessitates abstraction. Users must precisely define the problem space across multiple dimensions. One participant succinctly summarized his design problem for an office tower this way:

"I work to balance aesthetics and sustainability. We wanted a cohesive skin. [goal] The client wanted 40% natural light. [constraint]"

Our participants defined between 12 and 60 design parameters for the range of projects they worked on. They explained that it was not sufficient to simply describe the building or object geometry in the problem definition; rather, they needed to extract the underlying logic of their design problem. They routinely need to abstract variables so they can *"optimize things that aren't the same"* such as energy efficiency and structural efficiency. Designers also report that they often look at multiple competing criteria at the same time, such as daylight exposure, and view quality from windows in a building. Abstracting design goals and parameters into quantifiable representations is the first level of abstraction that design optimization demands from users.

3.3.2. Code as Abstraction

"You need to write the rules correctly."

The second level of abstraction required by design optimization is programming. Most, but not all, of the design optimization problems that our study participants work on require that they script algorithms to procedurally generate solutions. In the case of hospital design, the design goals and parameters interact according to a specific logic embedded in the optimization code, since hospital design is subject to rigorous building codes and regulation.

One participant explained the challenge of converting building code to software code in these words: *"you need to write the rules correctly for what you are trying to check."* In one architectural case 5000 lines of code produced 2000 design alternatives, each satisfying rules for solar gain, cost, buildability and acoustics.

When architects abstract designs into rules and make observations such as *"you need to write the rules correctly"* they begin to sound more like software programmers than architects. One of our interview participants rejected design optimization precisely because of this level of abstraction it demands. His view is that design optimization transforms the practice of architecture away from the "rational" or concrete practice of creating CAD drawings to a more abstract practice:

"Revit [a CAD tool] is rational. When you're working in Revit, your goal is set: I'm designing a building. I'm producing construction documents. In Grasshopper [a design optimization tool] there is no proscribed approach nor outcome. I can design a tree or a car."

This study participant asserted that when architects use design optimization to produce *"1200 variations, you're not being an architect any more. You are a computer programmer. A bus driver."* This comment implies that the level of abstraction inherent in the coding aspect of design optimization fundamentally alters architecture from a practice of documenting design into CAD, to a practice of abstracting rules into software.

3.3.3. Evaluation as Abstraction

"Then they start to push each other around."

Sound decisions about tradeoffs in multi-objective design problems require sophisticated statistical thinking. As we stated earlier, even skilled designers often struggle to comprehend the effects of combining multiple variables in a complex system. A particularly germane example from our data came from an interview with a structural engineer. He told us of output from design optimization that combined structural strategies in an interesting and unexpected configuration. Design optimization had produced a design wherein the widest section of the building he was designing had a tied fan structure with cross-bearings tapering as they went up, while two narrower sections used two different structural strategies: suspension and truss. When we asked for his reaction to the configuration he replied: *"you have these strategies in your mind, but you may not know how they will interact."* The abstractions in this example are the structural strategies. The design optimization system has no formal definition of a tied fan, suspension or truss. It simply computes the thickness and position of the beams. The designer abstracts what he sees in the output into structural strategies.

Our transcript from an interview with an architect/artist who designs and fabricates installation art is replete with abstract criteria that he used to evaluate the results from his optimizations. Examples of these criteria are: minimal surfaces, the absence of double curvature, and "*mathematical purity*". To visualize the design optimization process he output the calculations directly to a 3D model and watched the model transform as the problem was being solved. At the start, the design was very angular but once the design optimization ran, the geometric elements started to *"push each other around to find equilibrium."*

These two examples imply that evaluations may be concrete, such as the percent of solutions that meet or exceed 40% natural light, or they may be abstract, such as a structural engineering strategy or a mathematical purity.

3.3.4. Documentation as Abstraction – *"Not a perfect translation of genotype to phenotype."*

We were surprised to learn the extent to which architects abstracted their designs. One participant stressed that the *"parameters need to tell the design story."* An architectural firm we interviewed described the process by which they encoded the expression of architectural forms and performance parameters into an "architectural genome." Another architectural firm referred to the "master files" that they produced using their design optimization practice. These types of files, genomes and master files, are the symbolic description of the design problem. These files are the design story as told by the parameters. This type of documentation is a fundamental departure from the set of elevations and floor plans in traditional design. The obvious analogy, used by our participants, is that the master file is a genotype and the set of CAD drawings is a phenotype. One study participant cautioned us that there is *"not a perfect translation of genotype to phenotype"* implying that design optimization does not replace the expertise of the architects.

Design optimization requires that professionals abstract the design problem far more systematically and comprehensively than in traditional, iterative design processes. As a result of this obligatory abstraction, design optimization has enabled design professionals to generate and explore more mathematically complex design alternatives than they would otherwise. This process of ideation through abstraction appears to be the creative engine of design optimization. That said, abstract thinking is difficult even for trained professionals. One participant reported that plots of solution sets that are output from design optimization are more difficult for him to evaluate than CAD models: *"A [CAD model] is something you can respond to…I like that. I don't like that."* This was echoed by another respondent:

"the usability of the output is not there… even the visual examples are hard to digest for us, much less for the clients who don't have the expertise."

We were surprised to find this pattern in our interviews: user interfaces for data exploration in design optimization produce a poor user experience. This finding led us to wonder what attributes of these plots contributed to the poor user experience. At a cognitive level, we also wanted to learn how architects and engineers conceptualize the statistical analyses in design optimization, yet we have no data in our interviews on this topic. In a future study we plan to develop a framework describing how architects and engineers use data visualizations and apply that to prototype user interfaces for exploring solution spaces.

4. CONCLUSION

This research describes the use of design optimization tools across multiple professional disciplines including architecture and engineering. It leverages a creative cognition model of ideation (Ward et al 1996) to frame the activities we observed, including: generating design solutions, evaluating design solutions and describing both the problem and solution space. It examines an understudied creativity support tool, design optimization, and it articulates the role ideation and abstraction play in design optimization. One key finding is that that professionals use design optimization to gain understanding about the design space, not simply to generate the highest performing solution. Professionals reported that the computed optimum was often used as the starting point for design exploration, not the end product. A second key finding is that in some design organizations parametric models plus their associated parameters and simulations are serving as an alternate, highly valued form of design documentation distinct from engineering schematics. Much like a genome encodes a genotype, a parametric model and associated simulations encodes the expression of form and performance across any given set of parameters. The parameters tell the design story.

A next step for our research is to develop user interface design principles and prototypes that remove abstraction between solution sets and individual design solutions. User interfaces should enable users to easily pivot between exploring a solution set and examining a specific solution. They should also map decision-making criteria, such as parameter values and variable interactions, directly onto 3D models of individual designs. For any given design there are numerous nearby variations that are generated by incrementing up or down a variable range. We are exploring various user interfaces to present the detail of individual solutions while preserving the context of nearby solutions.

There are two distinct but related reasons why design optimization is important to the research community. First, this class of tool suffers from poor user experience (Maile 2007). Our professional users reported this in interviews, and expert academic users report “the lack of user-friendly, mature and comprehensive user interfaces limits the usage in practice.” (Ibid.) Poor user experience is due in part from a dearth of user experience research in the field of design optimization (Flager and Haymaker 2007). With this study, we hope to ameliorate the lack of research in this area.

Secondly, this research is important because users of design optimization are making buildings more structurally sound with less building material, they are making automobile engines more efficient, and they are improving the quality of care in hospitals by making them more comfortable places to work. Design optimization tools are used for sustainable design, yet are built on stratified layers of abstraction that we believe place considerable cognitive demands on the user. Abstractions exist when the design problems are defined, coded, and interpreted through data visualizations. Our findings suggest that the abstractions inherent in the workflows may conceptually distance designers from their designs, or lead them to make important design decisions based on incomplete information. This research may give tool developers, ourselves included, insight into how to amplify, focus and/or minimize these multiple levels of abstraction when designing optimization software.

Acknowledgements

We thank Michael Bergin for his coaching on design optimization techniques.

References

ARIEFF, A. 2013. New Forms that Function Better. MIT Technology Review http://www.technologyreview.com/review/517596/new-forms-that-function-better/

FLAGER, F. AND HAYMAKER, J. 2007. A Comparison of Multidisciplinary Design, Analysis and Optimization Processes in the Building Construction and Aerospace Industries. In *Proc. of the 24th International Conference on Information Technology in Construction.*

FLAGER F, WELLE B, BANSAL P, SOREMEKUN G, HAYMAKER J. 2009. Multidisciplinary process integration and design optimization of a classroom building. *Journal of Information Technology in Construction*, Vol. 14. 595-612.

FRICKE G. 1999. Successful approaches in dealing with differently precise design problems. *Design Studies*, Vol. 20, 5. 417-429.

GERBER, D. J., LIN, S.-H., PAN, B. AND SOLMAZ, A. S. 2012. Design optioneering: multi-disciplinary design optimization through parameterization, domain integration and automation of a genetic algorithm. In *Proc. Symposium on Simulation for Architecture and Urban Design*, Society for Computer Simulation International.

HOLZER, D., HOUGH, R. AND BURRY, M. 2007. Parametric Design and Structural Optimisation for Early Design Exploration. International, *Journal of Architectural Computing*, Vol. 5, 4. 625-643.

MAILE, T., FISCHER, M., BAZJANAC, V. 2007. Building energy performance simulation tools – a life-cycle and interoperable perspective. *Working Paper*. Center for Integrated Facility Engineering, Stanford University.

SCHMIT, L. A. AND THORNTON, W. A. 1964. Synthesis of an Airfoil at Supersonic Mach Number. National Aeronautics and Space Administration, *NASA Contract Report CR-144*.

SHAH, J., VERGAS-HERNANDEZ, N., SMITH, S. 2003. Metrics for measuring ideation effectiveness. *Design Studies* Vol. 24. 111-134.

SHNEIDERMAN, B., HEWETT, T., FISCHER, G., JENNINGS, P. 2006. et al. Creativity Support Tools: Report from a US National Science Foundation Sponsored Workshop. *International Journal of Human Computer Interaction*, 20, 2. 61-77.

STRAUSS, A. AND CORBIN, J. 1998. *Basics of qualitative research: Techniques and procedures for developing grounded theory*. Thousand Oaks, CA: Sage.

THIBODEAU, T. 2013. U.S. makes a Top 10 supercomputer available to anyone who can 'boost' America. In www.computerworld.com.

TSIGKARI, M., CHRONIS, A., CONRAD JOYCE, S., DAVIS, A., FENG, S., AISH, F. 2013. Integrated Design in the Simulation Process In *Proc. of Symposium on Simulation for Architecture and Urban Design*.

WARD, T., SMITH, S., FINKE, R. 1996. *Creative Cognition: Theory, Research, and Applications*. Bradford Books. 189-21.