S. Chien, S. Choo, M. A. Schnabel, W. Nakapan, M. J. Kim, S. Roudavski (eds.), *Living Systems and Micro-Utopias: Towards Continuous Designing, Proceedings of the* 21st International Conference of the Association for Computer-Aided Architectural Design Research in Asia *CAADRIA 2016,* 383–392. © 2016, The Association for Computer-Aided Architectural Design Research in Asia (CAADRIA), Hong Kong.

MULTIVARIATE SCHEMATIC DESIGN TOOLING

C. BEORKREM[1], J. ELLINGER[2], P. BERNSTEIN[3] and A. HAUCK[4]
[1,2] UNC Charlotte, Charlotte, NC, USA
{cbeorkrem, jefferson}@uncc.edu
[3,4] Autodesk,Inc. Boston, MA, USA
{phil.bernstein, anthony.hauck}@autodesk.com

Abstract. This paper will examine the results from a research collaboration between (BIM Software Manufacturer) and (School), whose problem statement focused on supporting robust interoperability by defining goals focused on multivariate conceptual design tools. The collaboration included design faculty, students and software professionals, the latter providing access to a broad range of design simulation tools either commercially available or currently in development. The tools were developed first through case studies and background research, followed by the design and implementation of novel computational methods advancing the architectural design workflow by seeking to create comparative tools which allow a designer to connect multiple data typologies in a single model. With advanced computational tools employed both as standalone resources and embedded in parametric loops, we sought to provide immediate feedback on design goals.

Keywords. Building information modelling; simulation and prediction; education; optimization; scripting.

1. Introduction

The earliest and still most ubiquitous use of computing in the AEC sector remains 2D CAD, which reproduces and incrementally improves the production of document sets. A growing segment of the industry continues to adopt Building Information Modelling (BIM), substituting manually managed sets of 2D files with models from which instruments of service are partially derived. While an improvement over the more cognitively dependent CAD process, BIM largely persists as a repository of decisions made outside the computational environment. The application of multidisciplinary data integration through BIM modelling improves building performance when compared with typical design methodologies (Wang et al 2005). As an improved CAD, BIM excels at more adroitly capturing and delivering instructions for what is to be built, but literally leaves everything to the imagination when addressing why something should be built. BIM used in this way maintains the status quo in the project methodology as it has been established prior to the use of computers, maintaining a persistent paradigm of design formulation and project delivery. However, with a reconsideration of how BIM is used during each respective design phase, a significant shift in methodologies could occur.

Advanced computational methods provide the opportunity to enhance building design by delivering encoded expertise and context, applying technology to do far more than record and communicate building decisions while helping the designer focus precious cognitive resources on the subjective and/or tasks more suited to human completion. As cognitive psychologists have proven, true multitasking is not humanly possible (Meyer and Kieras, 1997). What is perceived as multitasking is actually, quickly switching between tasks. It has also been shown that humans, once derailed from their primary task take longer to get back to capacity on the original task. Computers, however, can process several task streams simultaneously. The project outlined in this paper will describe new methods for leveraging multivariable computation during the schematic design process to collapse the distance between analysis and design and create more immediate feedback loops between a designer and the understanding of design performance.

This paper will examine the results of a research collaboration between Autodesk, Inc. and the University of North Carolina Charlotte School of Architecture, whose problem statement focused on supporting robust interoperability by defining goals focused on multivariate conceptual design tools. The collaboration between the academy and software professionals provided access to a broad range of design simulation tools, either commercially available or currently in development, coupled with current research trajectories into leveraging computation to better inform a designer of the implications of their design strategies and decisions. The tools created were developed first through case studies and background research into existing computational tools and methods. This was followed by the creation of comparative tools, which allow a designer to connect multiple data typologies into a single model. The results linked advanced computational tools employed as standalone resources, embedded in parametric loops within discrete programs and as scripts designed to facilitate program interoperability. The common goal for each was to provide immediate feedback to the user during the design process while implementing novel computational methods advancing the pace and accuracy of the architectural design workflow.

2. Context

In the current era of the Anthropocene, where the question of sustainability has become the primary focus a new theoretical redefining of our design practices can emerge.

Manuel de Landa (2002) has described a design technique that begins with the conceptual or 'cerebral' genesis of form, imposed on materials where the form or design is the primary focus of study – a teleological approach. A non-teleological technique in which materials are not inert receptacles of the cerebral idea is more aligned with a holistic energy-conscious design strategy whereby form (here material is understood in the broadest sense to include phenomena produced as matter and energy interact and then by extension in the field of architecture to include the social and programmatic), material and idea work in concert as multivariate partners towards a singular solution. De Landa builds an analogy for what this design methodology might be, based on physical material behaviors and the trajectories that emerge through gradient differentials to produce effects like phase transitions. De Landa offers up this analogy for a design method, whereby internal constraints and external conditions produce unique and specific results. If we can synthesize design constraints into similarly intricate logics or ecologies we can better understand their impact in the most holistic way; embodied energy, the properties of materials, and how those materials behave within a larger infrastructure to produce better design solutions from the lenses of the sustainable to the formal to the social.

At a theoretical level, De Landa's approach implies that only advanced computation can negotiate this complexity. The iterations (known as cycles in computation) required to complete even the simplest material simulations to understand their interrelations are beyond analogue capacity unless the simulations involve material computation such as Gaudi's hanging chain models. Material computation models such as Gaudi's are not entirely unique. Frei Otto's experimentations with minimal surfaces as thin-shell constructions are also examples of using material computation in design evaluation. However, even in these ground- breaking examples, the analogue technique does not go beyond the study of the relation between form and structure. The challenge De Landa is presenting is to construct a computational framework that behaves like the analogue models of Gaudi's – to immediately provide feedback that assist design decision- making.

At a practical level, this approach requires coordinating several discrete software packages, each of which possesses a singular focus. Recently, advances in the parametric capabilities of BIM packages such as *Revit* allow for geometric manipulations to be interconnected to substantial behavioral depths – beyond sharing data through .csv files. However, even the most robust packages only have capacity for very basic analytics when compared to discrete single function software packages – the question becomes is it possible to leverage the advanced analytics within a conceptual design framework in a useful way?

3. Multivariation

The intent for the organization of this project was to link seemingly disparate design elements together to create more intelligent schematic design systems. Inevitably each of the variables became more intricately linked as the tool was developed and relationships were formed through research into the values associated with each variable. By combining these data types and testing their relationships the project intended to discover new combinations of testable parametric fields, potentially including areas that had not been linked before. An abbreviated list of individual possible schematic design variables includes:

- Program organization
- Social capital
- Circulation
- Diagramming
- Site analysis
- Solar performance
- Daylighting performance
- Lateral loading and wind analysis
- Structural analysis
- Urban design performance
- Demographics
- Zoning
- Site connectivity
- Façade design optimization
- Rainwater control and collection

Each project was continually mapped using a logic diagram, to allow for further development by users and to allow for creative development by many involved in the research team, explaining how the tool works and allowing for others to test various iterations and create a collective workflow for each tool. The example below illustrates the tool logic outlined for the use of a workflow from *Robot Structural Analysis* through to Revit and back again, through the use of various data types through each piece of software and *Dynamo*.

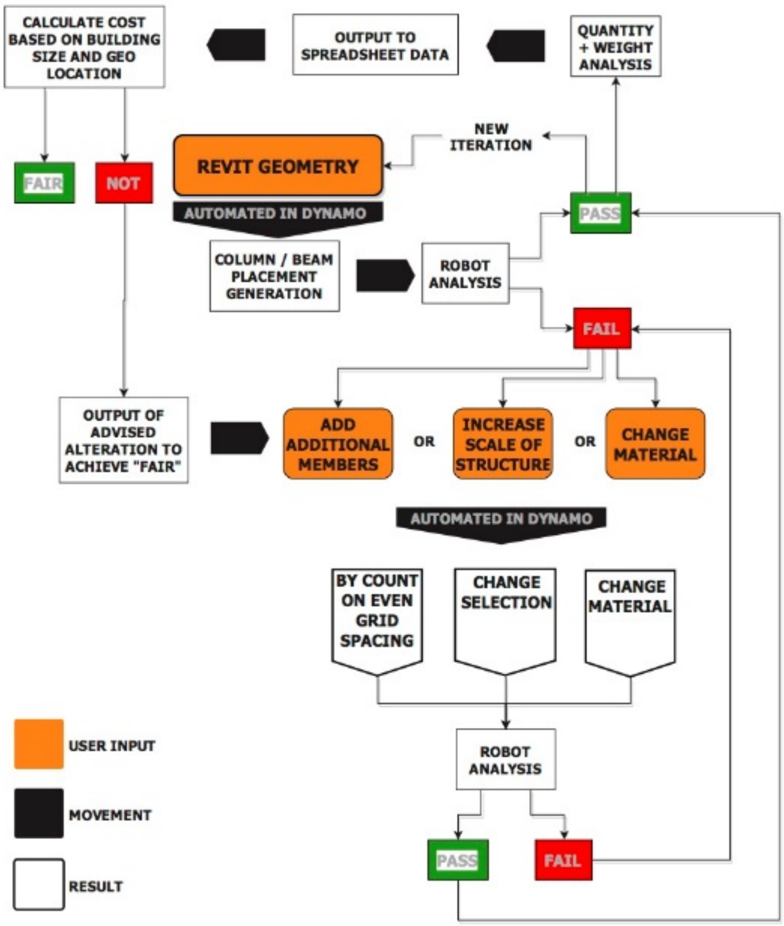

Figure 1. Logic Diagram of Robot Structural Analysis tool to Revit

4. Examples

Selected examples of projects that were successfully completed include:

- Creating a live link between *Robot Structural Analysis* software and Revit to evaluate structural member selection and spacing during schematic design phases.

- Using GIS information with a developed algorithm to evaluate each square foot of every floor in a proposed building assigning a relative value for each square foot based on available views.
- Writing a *Python/Dynamo* Script that interfaces through the API in *Autodesk Simulation CFD* and *Autodesk Revit* to evaluate façade design to optimize passive stack effects in tall buildings.
- Creating a value system for the creation of social capital (as a quality which improves workplace productivity) within a given programmatic assembly in large buildings.
- The creation of an adjacency tool for testing the performance of various organizational strategies for complex programmatic typologies.

Below we have unpacked some of these examples and provided a brief explanation of each of their data typologies and the link that was created between them.

4.1. STRUCTURE INFORMING PROGRAM

This tool attempts to empower the designer early in the design process by easily including basic structural design and analysis from the very beginning of a project's conceptual design. Any level of structural analysis this early on in a project is a common characteristic of a project delivery method known as integrated project delivery, however typically in practice, the structural system isn't quantitatively analyzed until the design has been more thoroughly developed. By empowering the designer with this tool, rather than relying completely on engineers to specify sizes for structural member sizes in a floor, wall, or roof assembly, the tool can automatically generate the structural model into schematic three-dimensional models that are sized according to design loads.

This tool gives the designer the ability to populate an entire building massing with structure as various massing models are attempted. This structure is then analyzed and sized automatically to conform to best practices in structural design. The resulting system shouldn't be perceived as the best solution or as a replacement for the work of an engineer, but rather a realistic representation of what the engineered system will eventually become as part of the project. This output equips the design team early on with structural quantities for the purpose of cost estimation and provides fail-safes that will prevent design changes due to poorly coordinated programs and their respective structural members.

Furthermore, this tool provides fully populated structural models that the design team can carry with them into design development. These models are easily updated and modified based on design alterations. Having structure integrated so early in the schematic design of a building can increase productivity and dramatically reduce errors and conflicts between various consultants. (Asl et al, 2015)

Using *Python* scripting as an interface, the tool develops a simple organizational strategy for structural members in a *Revit* massing model, and then uses the proven structural modeler *Robot* to test for functionality. If successful, the script populates the *Revit* model with member sizes. If the system fails to define a realistic structural design, a range of modification options were presented to the user, including shifting the massing, changing constructions types or spacing, or the primary direction of the members. After any design interventions were completed, the calculation would begin again. When a successful model was tested, a cost estimation was provided, so that the designer might be better informed about how each scheme performed compared to schemes with alternative geometry, spacing or construction type.

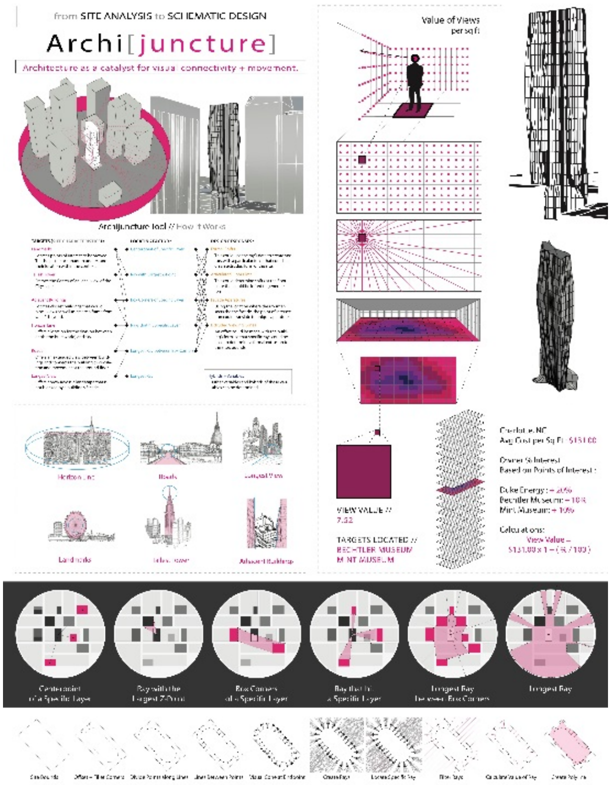

Figure 2. Diagrams explaining a tool which assigned value per sq. ft. based on view analysis from a given sq. ft. or zone on a floor plate.

This next tool attempts to assign value to a variable, which every building must consider but few understand parametrically, the quality of views. The design of this tool was produced with a combination of *Python* and *Dynamo* scripts and was based around rays projected in three-dimensions from eye level locations for every square foot on each floor plate to represent the regard of a building inhabitant from that point. The solution of the ray's intersection point with objects in a site model can be collected to create a data set that contains the potential for at least two clear methodologies. The first motive would be to assign value to all of the individual square feet on each floor plate so that a range of views throughout the building can be identified before it has even been built. The valuing system is flexible and dependent on each owner's criteria for potential points of interest and their marketing strategy. The second motive would be to use the tool for architectural design schematics so that more efficiently enhanced views of the site can be made.

4.2. THE BREATHING TOWER

The *Breathing Tower* generates performative geometries that encourage optimal, constant airflow conditions between intake and exhaust membranes of a building. The adaptability of feedback-driven forms makes this a flexible platform that can be applied on a range of scales. By simply selecting which profiles the tool can manipulate, the designer can limit the influence to a single facade, or allow it to shape the form of the building itself. This tool enables the designer to maximize the passive flow of air through a double-skin cavity, plenum space and core. This tool will allow the designer to place program elements (walls, cores, large furniture) as they desire, and let the algorithm optimize the geometry of the skin and core to achieve the desired flow at key measurement points.

This tool employed *Python* scripting to link the evaluation of each floor plate of a given *Revit* tower building massing. The floor plates were tested using *Simulation CFD*, which tested the rate of airflow through a double skin and across the floor plate to its exhaust. When objects impeded the flow of air across the floor plates, or as the height of the floor changes, or temperature shifts, all potentially would alter the rate of air interchange on a given floor plate. The system would identify changes to the geometry of the skin which would create more appropriate speeds and volumes for the given floor plate. The resulting geometry creates a map of air circulation through the double skin, while showing how differing designs for individual floor plates affect the overall tower massing when associated with its performance.

4.3. SOCIAL CAPITAL

The Social Capital (SoCap) tool attempted to create vertical and horizontal relationships in a tower design between programmatic components of any architectural typology into configurations, which would induce the creation of more social capital as a consequence of the design. Social Capital in this instance is defined by a variety of theories on understanding of human behavior and interaction in public spaces, including but not limited to Jacobs (1961), Briggs (2005), and Hsieh (2013).

All of these sources emphasize the need for programmatic organizations that encourage diverse groups of people to meet and collaborate for the sake of economy, potential for innovation, and societal impacts of performance. Most large cities encourage some level of interaction between diverse groups; but buildings largely fail to encourage these same kinds of genuinely diverse interactions.

SoCap analyzes the capacity for social capital in buildings at any stage of the design process. To use this tool, a designer first defines the types of occupants associated with each room (using a room tag) in a *Revit* model. SoCap then analyzes each floor for its capacity for social capital (much in the same way as occupancy loads are measured) by looking for matches between types of occupants. Example: if a floor has three types of occupants (Types A, B, and C), a full match between all three types counts as a full point towards the floor's match score while a match between types A and B only is weighted less than a full match. Then, the total match score is divided by the total possible number of matches to produce a diversity score.

Based on this analysis, SoCap makes recommendations to the designer that would increase the building's SoCap score. These recommendations may include connecting floors, by stairs and atriums, or inserting new programmatic elements onto certain floors. The designer can continue to make changes and analyze their designs at their discretion.

5. Conclusions and future work

While 21st century designers feel increasingly capable of managing multiple variables and compiling increasing amounts of information into a design process, we now know that humans are not capable of speedily shifting between tasks. Recent cognitive science research (Meyer and Kieras, 1997) has proven our brains are not capable of true multitasking or handling multiple streams of information simultaneously – rather we are good at switching quickly between tasks with a productivity penalty. However the amount of information a designer must compile is ever increasing. The only logical solution as mapped by other professions is to find ways to allow for technology to better manage the information streams which can be automatically calculated so the human user can better focus on the tasks that they are more capable of completing. The cognitive limitations of designers and their capacity to synthesize ideas and information into a single solution will always define the ability to factor ever increasing amounts of meta-data into schematic design strategies.

Further constrained by computational limitations despite vast improvements in speed and scale and parallel computing methods, the linkages between design variables are typically complex and often abstract to the human mind. Even the best designers can't manage this complexity and constraint alone. Collaborative work scenarios create better solutions while computation and software continue to improve our abilities to use technology to move beyond design based on "expert solutions" and to begin calculating design solutions. Only then can we expect designers to create "non-teleological" design solutions, which results in holistically ecological *and* phenomenal spaces.

References

Briggs, X. S.: 2005, *The Geography of Opportunity*. Brookings.

De Landa, M.: 2002, Philosophies of Design: the Case of Modelling Software, *in Verb: Architecture Bookazine*. Actar.

Hsieh, T.: 2013, "The City as a Startup Lecture". Available from: South by Southwest Austin <https://www.youtube.com/watch?v=aYjv4dKl7OM>.

Jacobs, J.: 1961, *Death and Life of Great American Cities*. Random House, New York.

Asl, M. R., Stoupine, A., Zarrinmehr, A. and Yan, W.: 2015, Optimo: A BIM-based Multi-Objective Optimization Tool Utilizing Visual Programming for High Performance Building Design, *in* Martens, B, Wurzer, G, Grasl T, Lorenz, WE and Schaffranek, R (eds.), *Real Time - Proceedings of the 33rd eCAADe Conference - Volume 1*, Vienna University of Technology, Vienna, Austria, 673–682.

Meyer, D. E. and Kieras, D. E.: 1997, A computational theory of executive cognitive processes and multiple-task performance: Part 1. *Basic mechanisms. Psychological Review*, **104**, 3–65.

Wang, W, Zmeureanua, R and Rivard, H.: 2005, Applying multi-objective genetic algorithmsin green building design optimization, *Building and Environment*, **40**, 1512–1525.

Wood, J.: 2007. *Design for Micro-Utopias: Making the Unthinkable Possible*. Aldershot: Gower.