AS502788

Using Dynamo and Generative Design to Simplify Complex 3D Schematic Design

Louise Schlatter
SSOE Group

Assisted by

Patrick Podeyn
SSOE Group

Alvaro Luna
Autodesk

Learning Objectives

- Learn how to assess a viable way to simplify complex problems.
- Gain insights that will help inform your path to automation and design optimization.
- Build upon your understanding of what working with Dynamo and Generative Design can do.
- Compare your experience with solving complex problems with this case study.

Description

What do you do when a building owner comes to you with a list of 100,000 space-occupying things they want you to organize in three dimensions, and then revises the list every month until it's done? This case study explains how SSOE worked through a complex series of relationships between objects involving multiple overlapping regulatory standards and environmental requirements. Computation and Generative Design worked together to help keep us sane, operate more efficiently, and improve customer satisfaction. This process is repeatable, and we think design optimization is real. Join us to discover whether our story inspires you to explore automation techniques. Learn from our experience with Dynamo and Generative Design.

Speaker

Louise Schlatter
Master Architect, SSOE Group
I fill roles for the architectural department, the manufacturing business unit, and the entire corporation that I jokingly refer to as “Answer Lady”. Highly technical problems, unique challenges, new market sectors, workflow improvements, training and mentoring are my core responsibilities. I am the Architect of Record for several large complex projects for our manufacturing clients. My history is filled with specification development, sustainable and value-

added building design. The thousand-employee strong SSOE Group has an international footprint covering 40 countries and 6 continents, and my experience with them includes half of that. Although the projects with which I am involved are highly confidential, a public one for which I was the Architect of Record and LEED Administrator is the Volkswagen Automotive Assembly plant in Chattanooga, TN; it was the first and largest LEED Platinum certified manufacturing facility in the world. My current focus is on supporting building designs for battery technologies and improving workflows internally and with our partners.

Patrick Podeyn (SSOE Group) and Alvaro Luna (Autodesk) are key members of the case study project and essential to the programming effort. They will be assisting with the Q&A and contributing their technical expertise. Their contributions and outcome of the Q&A portion of the presentation will be captured and added as supplementary material after Autodesk University.

Introduction

This case study is intended to provide insights for people in the architectural, engineering, and construction (AEC) community considering Revit-based automation. The automation tools are new and still developing. We will be exploring how combining old and new tools combine in powerful ways to reach project and automation goals and meet the Learning Objectives of this course.

The Challenge

What kind of a project is a candidate for automation?

The easy answer is any repetitive and mundane activity. This includes labeling details, views, and sheets, moving or shifting repetitive objects, updating parameters or text, collecting text for a list, and others.

What about activities that setup or create the basis of a design?

Are you one of those who thinks a new project starts with a blank sheet of paper or screen? Let me suggest to you this is seldom the case, particularly in the AEC industry. By the time members of the AEC team are engaged, the owner has some conception of what the project should be. Typically, the architect’s role is to translate the owner’s information and vision into graphic and technical information that the construction and facility management communities can use and understand. From an architectural perspective the building project starts with architectural programming and schematic design, and all the AEC aspects of the project flow from that.

I am an architect. This is what I do.

The tools of architectural programming are not unlike others with which you may be familiar: Lean-thinking, target-value delivery, agile, scrum, and of course, Excel spreadsheets. The information is collected, distilled, rated, and presented graphically.

The Story: One day a client brings you a project where they want 100,000 space-occupying things organized in three dimensions in the most efficient manner practicable. There are fifty different criteria for how they should relate to each other and another fifty rules for how they must be organized. In some circles these are called goals and constraints.

A two-dimensional solution (2D) will not work. The solution must be three-dimensional (3D).

Learn how to assess a viable way to simplify complex problems.

Now what?

How to deal with all this information? A random organization of 100,000 things in space will not work. Nor will plugging one hundred variables into a Dynamo script work; you will gain an appreciation for how bad this idea is when you get to Respect the Factorial discussion below.

Today, the raw data of 100,000 things is too complex to use all at once. We need to leverage architectural design tools to organize and simplify our understanding of the problem before we can process and propose a solution, with or without the aid of automation.

The first thing we did was create a planning unit. This move made the footprint of the 100,000 things modular and easier to make fit into the available space. Just for fun, let us call the planning unit a swidget. Figure *1* is a visual representation.

Figure 1: To protect the innocent, let us call this a swidget. It has length, width, and height, and characteristics that collect similar members of the 100,000 things into unique planning units. Now we have about 25,000 swidgets and every one of them wants their own best place to be in the available three-dimensional space.

Set-Based Design

The design process is by nature iterative, leveraging and managing the iterations is key to successfully delivering value to the client. One smart approach is to use set-based design, a technique defined by Lean Construction Institute as “a design method whereby sets of alternative solutions to parts of the problem are kept open until their Last Responsible Moment(s), in order to find by means of set intersection the best combination that solves the problem as a whole.” It has also been described as controlled convergence. The time constrained sequence is 1) frame the (next) critical issue and selecting options, 2) generate, evaluate, and refine or abandon options, and 3) select best one for satisfying the owner’s requirements and agree on accepting the associated risk before moving on to the next critical issue. If you are visually oriented like me, pictures help; Figure *2* is the graphic description that has helped me understand and describe set-based design.

Set-based design is a tool best used for those items that do not have an obvious answer. For example, it is not necessary to decide whether a room needs an egress door; every room requires an egress door. It could, however, be used to identify the optimum location for the room and its door by considering the options for its location: A more remote location may minimize the noise disruption but negatively impact exiting. A more central location will shorten the time to one's next location but reduce the desire synergy with the room in the other direction. A location near an exterior door shortens the travel distance to an exit but increases the quantity of mechanical tempering. Which one will be more successful in delivering a project that will meet the client's goals?

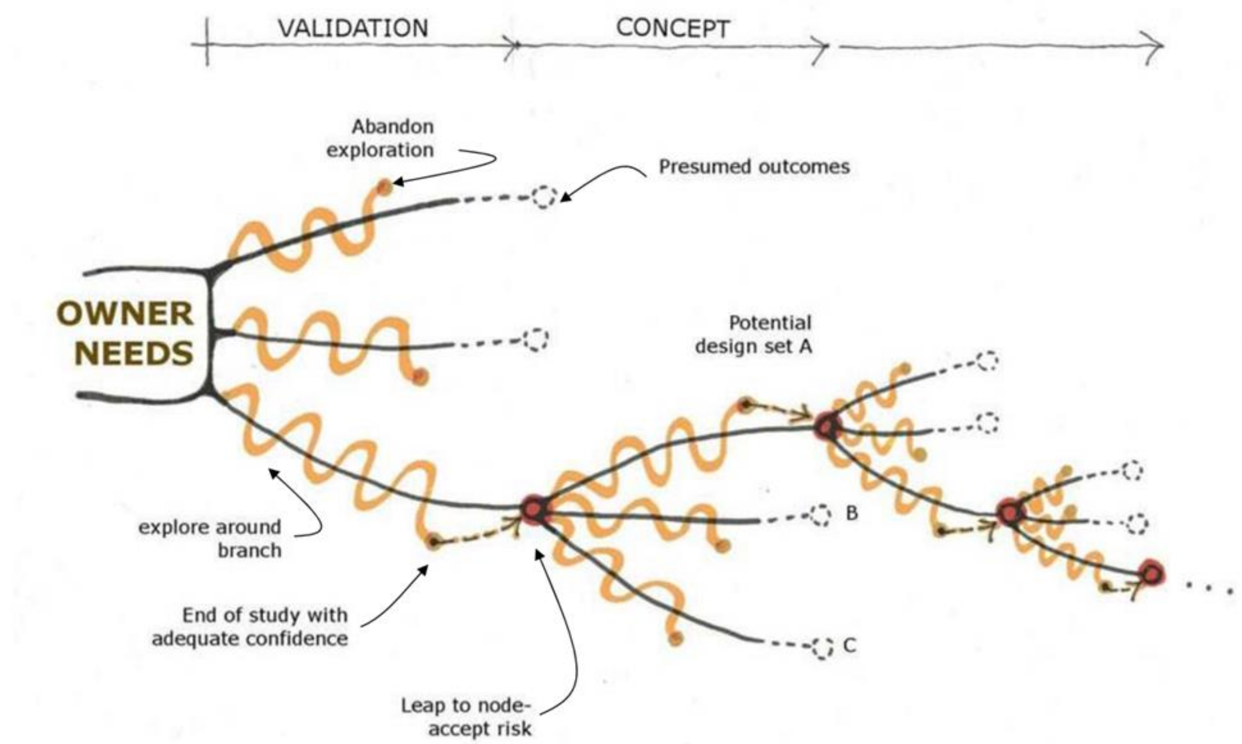

Figure 2: A graphic representation of set-based design from Lean Construction Institute presentation "Explorations In Set-Based Design" presented by Tipping | Mar on July 11, 2012. https://leanconstruction.org/uploads/wp/media/docs/chapterpdf/nor-cal/2012-07-11-lci-nor-cal-meeting.pdf

Set-based design is all about making the right decisions at the right time in a way that leads to the next decision that needs to be made. It is not the classic linear decision making where iteration is a series of repeats, revisits, and backward steps. It is not only putting decisions in the right order but making the magnitude and size of those decisions one that can be managed effectively.

If you research set-based design, you will find many resources referencing a Toyota method of designing. These describe how this method helps to control cost[1], increase and share knowledge between projects, and reduce backward iterations by leveraging stakeholder involvement.

Is It a Candidate for Dynamo and Generative Design?

Look at those 25,000 swidgets and the other information that goes with them. Compile them and look at them again. What do you know other than you have a lot of stuff? Are there items that seem to naturally group together? Even if you cannot locate them in space, can you develop the relationships between them? Consider exploring a set of relationships as an early branch in set-based design.

At what point does Dynamo and Generative Design fit into this? How do you know whether those tools are right for this project, or a portion of this project? If you can answer yes to most of the questions below, then Dynamo and Generative Design are good tools for your project.

- The number of reasonable answers is greater than one. When you eliminate all the yes-no decisions and all the single answer linear decisions (i.e., this, then this, then this, or basic algebraic logic), what is left? Are there multiple answers with no obvious best answer?

- Success can be measured by the results of a series of formulas. Can you conceive, with a bit of creativity, a way to measure the relative level of success? Are there multiple considerations that you are trying to optimize? Do the items you are trying to measure have dissimilar data types?

- The client is expecting options or a series of pros and cons. The client wants to be engaged in the decision-making process. You want the client to understand the effects of the decisions being made. Does a fact-based, or numerical option presentation sound like a useful tool?

- Manually creating viable solutions would take a crazy amount of time. If the computer can make one thousand good solutions in less than the time it would take you to create one, why not leverage the computing power of the computer?

- And this assumes you can make or program a Generative Design study to work for you.

[1] If you have a specific interest in cost control, widen your research to include target value design or delivery (TVD).

Generative Design has a feature that assists with comparing nonlinear relationships, dissimilar data types, and can engage your client in the options discussion. It is the Autodesk stylized optimization graph that is built into the Generative Design feature. While every project generates a different graph, it looks a little like this:

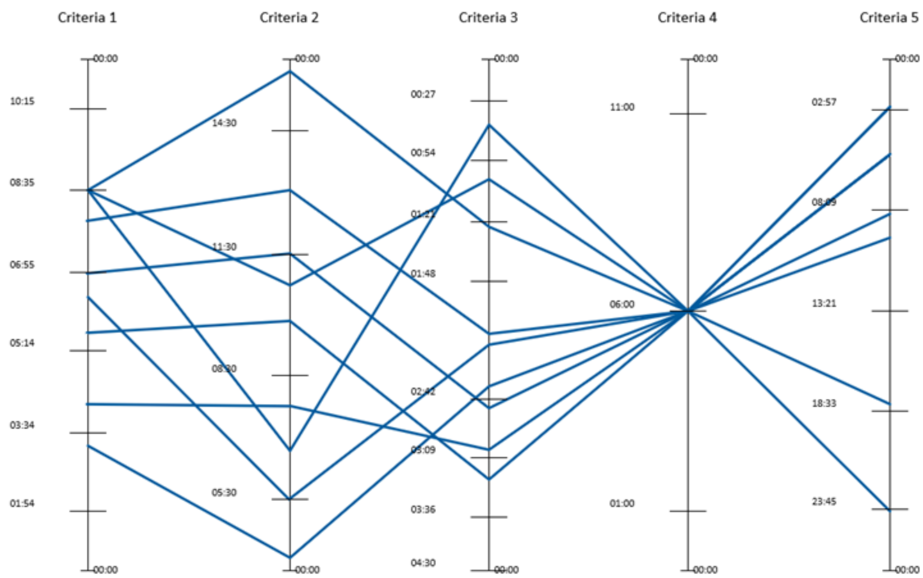

Figure 3: General idea of what an optimization graph looks like. The vertical line graduation will match the grading scale applied to the criterion represented by that line. Each lateral line represents the performance of an option relative to each criterion. The number of lines represented match the number of options generated and can be reduced by refining acceptable range of grading in each criteria line, a function exercised by dragging on a criteria line. If a criterion is not being measured, e.g., a Dynamo node is blocked, frozen, or disconnected, or the program creates no variation, the graph will pass through a single point (refer to Criteria 4 in the Figure).

Each horizontal path is a Generative Design option. You can have as many options as you are willing to wait for Generative Design to generate. Once they are generated you can use the optimization graph to identify the combination of attributes that form the set of best, most desirable options. This can be done during a conversation with your client, allowing you to interactively refine the decision criteria and to give a greater prioritization to certain options or criteria.

The interesting thing about having to organize 25,000 swidgets in three dimensions is most relationships between objects cease to be linear. If the “may not be adjacent” rule is met, the potential number of successful organizations is shockingly close to infinite. Optimization, then becomes a balance of multiple and diverse other measurements, or parametric computations if you prefer. This is where the Autodesk optimization graph becomes a powerful tool.

There is a wealth of Dynamo and Generative Design information, blogs, and presentations on the Internet, naturally most of which are hosted by Autodesk. While looking at these you might notice most of the other AEC building related Generative Design presentations are for two-dimensional solutions, i.e., 2D objects on a plane or 3D shapes on a plane. Here are two of this type we found particularly interesting and inspiring when considering how to approach this project:

> Space planning in Dynamo with DynaSpace. The Autodesk hosted DynamoBIM.org has the DynaSpace package for bubble diagramming developed by Long Nguyen (Institute for Computational Design and Construction) and Mohammad Rahmani Asl (Autodesk Generative Design Group for AEC). You can also see it as part of an Autodesk webinar, Generative Design in Revit: Three workflow examples to jumpstart your practice.
>
> Generative Design at Hogwarts: Using Tech Instead of Magic. This 2020 Autodesk University presentation by Jacob Small and Alexandra Nelson shows three different strategies for organizing objects in space.

Our research turned up truly little about additively organizing objects in three-dimensional space. Most of the work done in three dimensions is about subtracting extraneous material.

Gain insights that will help inform your path to automation and design optimization.

So where are we? We have a strategy to use set-based design to help us control the decision-making process. We think using automation tools has promise. Can we develop the Generative Design skillsets to make this work? This is what we learned:

> First you must be open to trying new tools and ways of doing things. This does not mean to trash the old ways of doing things. Those old ways are powerful informers of the new. For example, what you know from manual design processes becomes a good basis for grading the performance of computer-generated options. The things you know about good placement may not fit globally in the option generating process, but it can have a significant bearing on the option grading process.
>
> Second, you do not need to do it alone. Part of our relationship with Autodesk is having access to their Implementation Services. We are actively developing our in-house Dynamo skillsets but are not ready to effectively tackle the backbone programming it takes to develop a new Generative Design concept. Here are a few things I learned along the way about using the service:
>
> > Autodesk Implementation Services cannot be tied to production deliverables. They are wonderful people, but your client is not their client. You have contacted them for a new thing that is still being defined and the process is uncertain and unrelated to your client project timeline.

A Weekly Check-in is essential. You want to understand what your Autodesk consultants are thinking and whether it continues to fit your goals. It provides you with a seat in the development decision-making process. You help to shift criteria to or from option development and success grading. It also helps you understand how the input data needs to be structured.

Then you need to make some important adjustments to your traditional approach to architectural programming.

Respect the Factorial

This advice is why it is important to control the number of variables being manipulated or being considered at any one time. Looking at the set-based design graphic, Figure *2*, consider this a key consideration for a single node and branch.

A factorial, written as n!, is a mathematical way to describe all possible linear combinations when each combination and order count as unique. The integer used represents the number of items in a linear decision of arrangement. The factorial is the product of that integer and all the positive integers less than it, for example[2]:

1! = 1
2! = 2 x 1 = 2
3! = 3 x 2 x 1 = 6
4! = 4 x 3 x 2 x 1 = 24

Have the idea?

Notice how fast the number grows:

8! = 40,320
12! = 479 million (That is 6 zeros.)
15! = 1.31 trillion (That is 12 zeros.)
18! = 6.40 quadrillion (That is 15 zeros.)

Now think about it in terms of computer time. Even if each iteration is 1/1,000,000th of a second, 18! adds up to 203 years of processing time. 15! shortens the time to 15 days, and 12! shortens time to 8 minutes.

Recall this describes linear combinations which is technically one dimensional, which we typically think of as length.

[2] For those that familiar with permutations: The swidgets are all in. Every order is unique. The denominator of the permutation becomes 0!, which equals 1. Conceptually all we need is the factorial.

Considering a two-dimensional organization, e.g., one that has length and width, changes the number of combinations to factorial times factorial (n! x n!). The time estimates jump to 1.30 quintillion years (18! x 18!, a number with eighteen zeros), 19.8 trillion days (15! x 15!), and 3.82 billion minutes (12! x 12!).

Considering a three-dimensional organization, e.g., one that has length, width, and height like our swidgets, changes the number of combinations to factorial times factorial times factorial (n! x n! x n!). Let us just say the time estimates jump to unreal, crazy numbers. Following the same logic even the time to calculate through five variables is daunting; 5! x 5! x 5! factors to 20 days of processing time.

The big lesson is employing a strategy other than applying brute force is essential.

If you are inspired to get deeper into statistical methods and the design of experiments, consider taking a look at an e-text hosted by National Institute of Standards and Technology, the NIST/SEMATECH e Handbook of Statistical Methods, http://www.itl.nist.gov/div898/handbook/.

Adjusting Architectural Programming Tools

Architectural programming traditionally uses several tools for describing and comparing orders of preference and desirability, the most common of these is a matrix. In manual practice, these matrixes describe design objectives or evaluation criteria that can be applied in schematic design and beyond.

One type is an adjacency matrix. See Figure *4* for a generic spreadsheet example where both axes are the same length.

This is the type of information we had as input for how those 100,000 things needed to be organized. That is, locate these things “near”, “close to”, “not close to”, or “may not be near” to each other.[3] We found with Dynamo and Generative Design that we only had the capacity of generating to one of these criteria, either a hard “yes” or “no”. The other criteria either needed to be ignored or a creative way invented to turn it into a measurement that would show up on the optimization graph.

[3] This is an example of Fuzzy Logic. Dynamo does not seem ready for Fuzzy Logic, i.e., degrees of yes or no; it currently works well with Boolean Logic, i.e., a precise yes or no, currently.

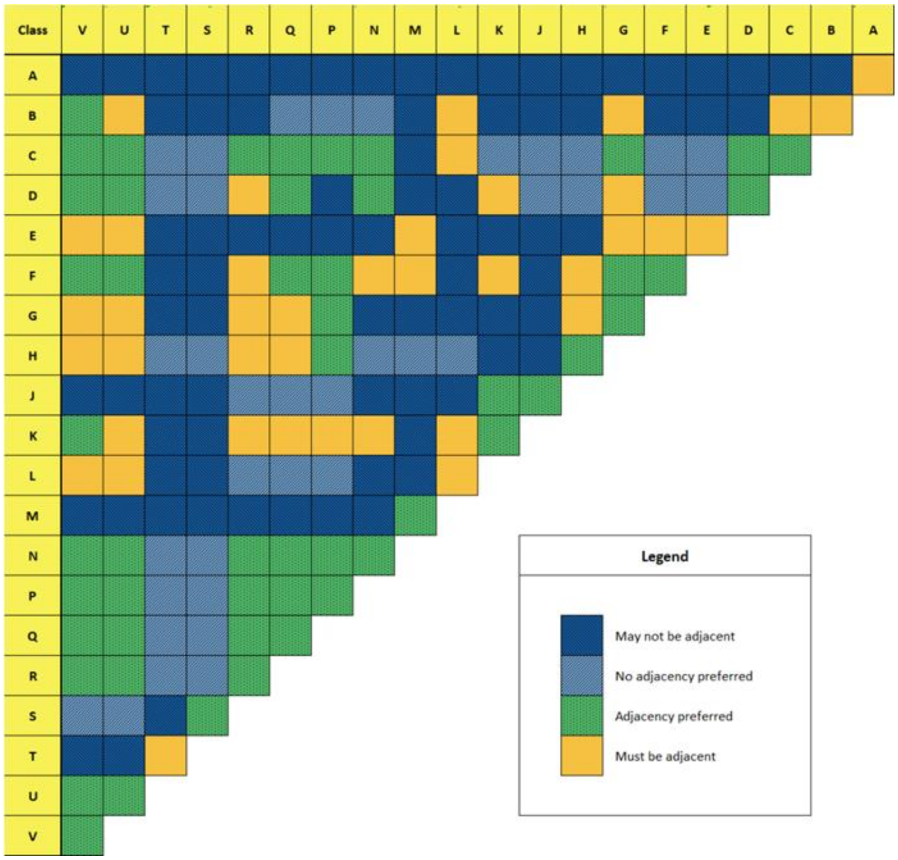

Figure 4: Sample adjacency matrix from architectural programming which shows “yes”, “no”, “maybe yes”, and “maybe no”.

There was another type of information in the project that formed a different type of criteria. Traditionally captured in a matrix, the relationships are not based on proximity, but functionality or other more esoteric criteria. See Figure *5* for an example. This matrix looks easier, right? Even assuming all the data types expressed in the matrix matches, this is not so. It seems although the information is well described as a matrix, the usual way a computer looks at a matrix is as an array of arrays. This seems to defeat or inordinately complicate the lookup strategy one thinks a matrix represents. The lesson is two-dimensional arrays need to be reduced to lookup lists.

To be able to use Generative Design as it exists today, architects need to adapt architectural programming to use optimization practices. A particularly useful one of these is Pareto[4] thinking, or its common name the 80:20 Rule. Consider manual manipulation or using Dynamo to automate the 20% that drives 80% of the results, then use the optimization tools of Generative Design to sort through the other 80% to refine the last 20%.

[4] The Pareto Principle says that 80% of consequences come from 20% of the causes. To use this information, one makes a list of topics or consequences and performs and exercise to find their root causes.

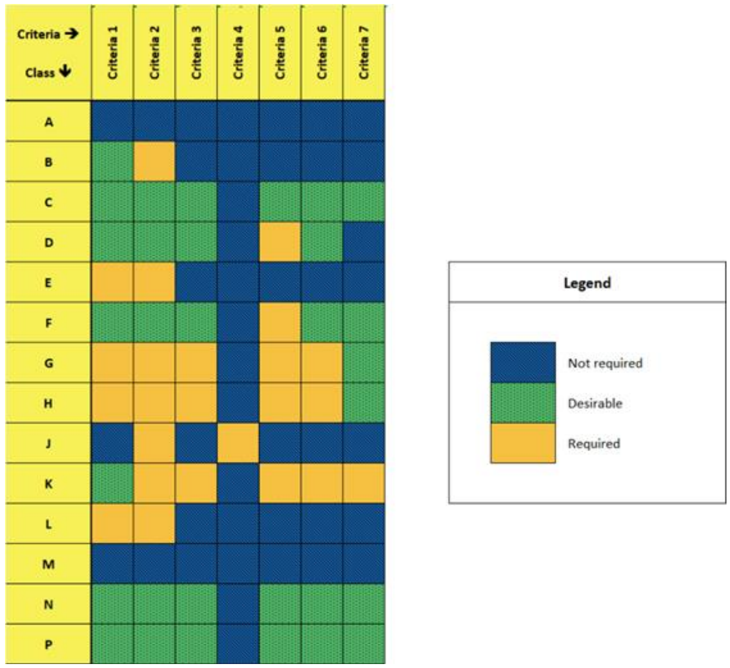

Figure 5: Sample criteria matrix from architectural programming.

For example, using Figure 5 above, this may mean you create a solution of those items that have more than two required criteria and develop a grading scheme that can help address the items with two or less required criteria.

You may find it useful or necessary to combine optimization graphs with other optimization strategies. One of these is to introduce drift as a mechanism for adjusting the process to a desired level of refinement. In Dynamo this frequently means employing sliders. A slider allows the user to adjust input criteria on successive runs with the intent towards moving the outcome closer to the optimum. This takes understanding of the problem and the script created to solve it. It takes time to make successive runs and patience to make “just the right amount” of variation to the right slider. I found a slider useful for creating aesthetic variations, and problematic for purposely moving options toward an optimum.

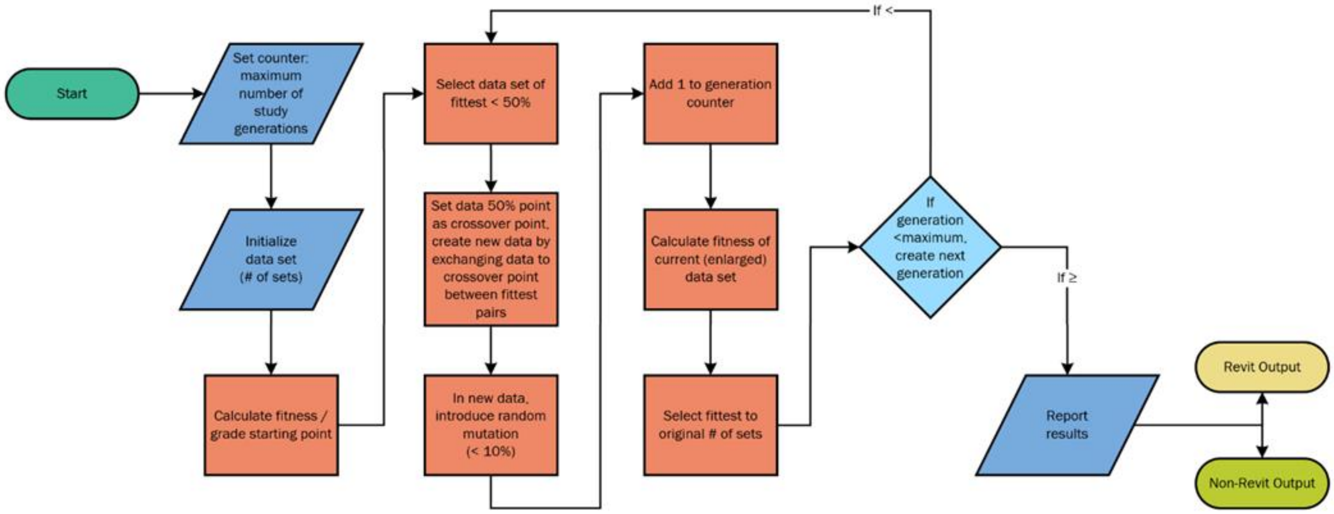

Figure 6: One strategy for evolving data to the optimum. Based on Genetic Algorithm.

A more automated option would be to program an algorithm that steps intermediate outcomes towards a specific goal. This is a strategy one might find in artificial intelligence circles or in estimating genetic migration. See Figure *6* for the basic flow of this type of algorithm. Alternately, your programmer for custom nodes may have an alternate or favorite technique for balancing randomness and motion toward an optimized solution.

Caution, while this removes randomness from the sliders, you may be giving up result randomness. The calculations drive options to a specific goal or optimum. You must consider how this does or does not fit your project and the structure of your variables.

Build upon your understanding of what working with Dynamo and Generative Design can do.

These are some of the things I learned about working with Dynamo and Generative Design:

- These are new features added to Revit. They are not single click commands. They take a certain amount of development. They are rapidly changing; as a consumer you are helping to develop their capabilities.
- Dynamo is a visual programming system built on nodes. Limitations of the existing programming is often supplemented by creating what will appear as a new node using another programming language such as C-sharp, Python, Grasshopper, or another programming language.
- Lists and dictionaries are the most powerful tools in Dynamo. Magic seems to happen when you can reduce data to a list and then manipulate the list.

- Dynamo does not perform its work in Revit. Revit information must be pushed into the Dynamo application and as quickly as possible converted to an appropriate programming data type before it can be used. In programming, data types can be unforgiving, pick the right one and use it consistently.

- Not the Generative Design output, nor the Dynamo output, is going to be automatically found in Revit. Purposeful steps must be taken to get it there. Each Generative Design option opens a virtual version of Dynamo; you need a way to capture the desired information, it may not be simple to repeat or find the one you want again.

- This is a computer application; the adage “garbage in equals garbage out” still applies.

- To an architect there are other subtle things that are different when using Dynamo, such as a line is programmed as a curve. (I know some of you are saying “duh!”, but it took me a couple of conversations before this seemed normal.)

- Measuring distances between two objects is typically as the crow flies. These tend to be accomplished as the radius of circles, or spheres, around a point. There is no (simple) way to account for barriers on that path, like walls, or changes in elevation.

- Leveraging all the tools at your disposal for simplifying complex problems is critical to your success. Throwing the whole problem at Dynamo and Generative Design is going to be fraught with angst and heartbreak. The maturity level of Dynamo is in the parts and pieces; consider it a tool, not a solution.

- We are storing project data in cloud-based applications like Autodesk Construction Cloud, and where ACC does not host it, in other third-party applications like Orkestra. How your user interface interacts with these cloud-based storage locations will affect your ability to run Generative Design and Dynamo. When creating your script, be sure to allow time to sort through the idiosyncrasies of your setup. For example, we could not run Generative Design until we resolved the meaning of the message “server not found”.

Compare your experience with solving complex problems with this case study.

This project used Dynamo and Generative Design as tools for managing the organization of 100,000 things reduced to 25,000 swidgets in three-dimensional space. Structurally they were placed either on the floor, in a pile, or on a support structure. The spaces in which they were placed had all the right environmental attributes and building code compliances; Dynamo used but did not help with these.

This is the column structure of the Excel data used as input for this case study:

Index (for tracking, identifies the Excel row)
Object name (this is for my convenience)
Class ID (a classification used to determine acceptable adjacencies)
Purpose (in architectural program this separates objects to be arrayed and spaces)
Quantity (how many units, in this case swidgets, were represented by the Excel row)
Each swidget length (clearance factors were built into this dimension)
Each swidget width (clearance factors were built into this dimension)
Each swidget height (clearance factors were built into this dimension)
Unit description (this was used as a check)
Unit criteria 1 (separates required placement criteria, some items must be at finished floor)
Unit criteria 2 (sets maximum height an object can be placed)
Unit criteria 3 (sets environmental requirements)
Cohabitation (string identifying which Class IDs can/cannot be together)
Color (for visualization convenience)
Unit criteria 4 (this criterion came and changed as the script developed)

The Revit model had rooms designated as being suitable for these objects in a manner that Dynamo could identify as where it should perform its operation. This involved adding a couple of parameters to the Room.

We learned as we were refining the programming, it is never too late to start thinking about how to go about the Generative Design scoring and what it might tell you.

Alas, none of us knows everything. The image of the master architect who knows everything is unrealistic at best. Most problems take a team to resolve, or as set-base design would refer to them, stakeholders. This project was no exception. Our contractual relationship with Autodesk includes access to Autodesk Implementation Services. We did not have the depth of programming expertise to do this alone. Similarly, our implementation consultant did not have the depth of architectural programming knowledge to do this alone either. Regular touch points were essential for keeping us working towards a common goal.

This project did expand the genre from two-dimensional organization to a three-dimensional one. There are articles about designing a three-dimensional network for 3D printing, but this project was not a fit for building upon those. This project had discrete objects that were being organized, where network strategies were about removing unnecessary material from what would otherwise be a solid. The objects in this project were orthogonal and their organization needed to follow orthogonal rules. The process was additive, where the 3D network strategies were subtractive. This project required a novel approach.

The temptation to continually move to the newest and shiniest will become a consideration. There were times when our Implementation Services representative fussed at the Dynamo and Generative Design versions we were using, saying the next version of Revit fixes this or seems to fix that. While new stuff is exciting and fun, I work in a design firm. Design firms are always engaged in revenue-generating work. A new piece of software means testing for compatibility with the network, employing rollout strategies, training users and support personnel, and coordinating anything found to be necessary to make the new software work. Changing versions of Revit mid-project needed to be a “no”.

Software instability periodically comes up. What is it? Me? I am an architect. I do not have The Answer. My understanding thus far: This is new. Dynamo nodes and Generative Design components come from many sources. It is possible to have two nodes that appear to do the same thing but are as different as two people can be. Putting nodes together is like forming a team, if you put the right people together you have a wonderful team. Putting the wrong people together can result in unintended consequences. This is a programming issue. Is it an example of instability? I do not know.

Not everything gets solved perfectly, be vigilant. I found it challenging that our script periodically forgot the difference between available room height and how high things can be stacked. Conventional wisdom seems to suggest that things cannot be stacked higher that the space available. However, we had areas in the project that were new, and the height was not set. The script did not seem to appreciate the difference, yet.

When you take this avenue for project development, quality control is as important in programming as it is in building design. Punctuation and spelling errors can stop or skew your outcomes. If you do not have the skillset, consider looking for a resource that can provide QA/QC input.

Excel files are a wonderful way to organize information, however if you want to optimize the outcome, you want to start from randomized data. It seems starting with ordered data can lead to a false optimization, particularly if you constrain the number of runs either by total number or by increment. The solution looks good, but it is just giving back to you what you fed it. Early in the process the script simply stacked the objects in the quantities given. Which meant everything fit and a couple of the Generative Design measures all returned the same number. This was not what we were after.

Sliders were proposed to randomize the data. When working in three dimensions, this proved to be uncontrolled and lacking practical reasoning. One slider was needed for every type of data. If this script were repurposed for other projects, the number of sliders would need to adjust to match the number of data types. Sliders are also lacking as a persistent User Interface. Another way to randomize data was required.

We learned that Python may be a great programming language, but not the best for developing User Interfaces. To develop a script for wider consumption, people need a way to interact with it. As of this writing, this is not yet resolved.

There are several things related to this Dynamo and Generative Design project that are not resolved. However, the revenue-generating project on which it was modelled is complete. The things we learned from doing this project did not solve the revenue-generating one, but it did inform it in a manner that improved its result and helped to make the client happy.

Next Steps

What do we see as the next steps?

Develop and refine a User Interface that might fit with an internal deployment of this script.

Further develop a non-slider means for randomizing data.

Explore whether other volume types can be included, with the intent these would be used in Generative Design grading.

The volumes we used had the things and clear space as one. Explore whether we can be smarter and orient the clear spaces.

Develop a deployment model.

The Usual and Customary Resources

The Dynamo site hosted by Autodesk: https://dynamobim.org/
The Dynamo Primer (web book): https://primer.dynamobim.org/
The Generative Design (web book) copyright by Autodesk: https://primer.dynamobim.org/

Interesting Set-Based Design Resources from a Google Search

What is Set-Based Design? By Singer, Doerry, and Buckley:
https://www.researchgate.net/publication/227761149_What_Is_Set-Based_Design or
http://www.doerry.org/norbert/papers/SBDFinal.pdf

Published Patterns on Set-Based Design: https://sites.google.com/a/scrumplop.org/published-patterns/value-stream/set-based-design

Annotations and Acknowledgements

How Terms are Used here

Architecture, Architectural, Architect: These terms are used with their meanings relating to designing and constructing buildings. (It does not reference computer systems.)

Owner: The entity designated in construction contracts as such. They are thought of as the entity who holds the title and fiscal responsibility for the completed building.

AEC: An acronym used to describe the collaborative nature of building construction and the parties engaged in it. Stands for Architectural Engineering and Construction.

2D: Two dimensional; having length and width.

3D: Three dimensional; having length, width, and height.

Disclosure

1. SSOE accepts the predictions saying the current methods and means will not be able to support future building construction needs. Therefore, SSOE has an interest in exploring and developing Generative Design methodologies.

2. SSOE has found, for us and our corporate culture, having a revenue-generating project as a model to mirror/support/parallel works best when developing new skillsets and applications.

3. The author and presenter, Louise Schlatter identified the project on which this presentation is based in the proposal phase and was engaged in it architecturally throughout the project.

4. The original project is confidential, but the techniques, lessons and outcomes apply widely to the work we are asked to do.