

AS195942

Connecting FormIt and Dynamo for Predesign Formal Thinking

David R Beach
Drury University

Learning Objectives

- Learn how to generate a context model and contextual data for a design project in FormIt.
- Learn how to use Dynamo to develop formal design strategies to respond to both analytical data and visceral design thinking.
- Learn how to push the model to virtual reality for early design-process exploration and communication.
- Learn how to export the model for fabrication, including CNC cutting (site) and 3D printing (design and context).

Description

FormIt architectural modeling software connects 3 elements of the design process that are relevant to every project: contextual information, building performance simulation, and parametric modification. Learning to connect these pieces quickly and efficiently in the architectural, 3D sketch-style environment of FormIt enables quick iteration opening new modes of design thinking. This session will illustrate a process that we're teaching architecture design students: first, how to quickly build design context (both from data acquisition sites and the creation of custom models) of buildings and topography; second, how to develop predesign simulations to inform decisions and create a baseline for building performance; and, finally, how to develop formal strategies by using parametric processes with Dynamo software connecting design ideas and simulation data. In closing, the session will cover predesign visualization beyond FormIt, pushing the file to prototyping and virtual reality.

Speaker(s)

David R. Beach is an architect and Associate Professor of Architecture at the Hammons School of Architecture at Drury University. Specializing in digital design technology, David is an advocate of full digital immersion as part of a traditional design process leveraging technology to inform process, collaboration, and decision making through analysis. David teaches in the architectural design studios, the Center for Community Studies working at the urban design scale, and both the introductory and advanced digital design technology courses. The application elements of David's current research can be found at the Autodesk Design Academy, and on his blog and YouTube Channel: The Architect's Digital Design Guide - which combined have over 6 million minutes of viewership, and are outlets for work and information directly related to questions received from students and practitioners. David has given 8 presentations at AU, and has completed over 25 major conference presentations in the last 10 years.

Simulations Start Before Design Thinking Starts:

This is a topic that roughly revolves around the ideas of tying together of software and design thinking in a way that the AEC profession can connect Predesign Formality, Artificial Intelligence, Fabrication, Simulations, and Visual Programming. All cool topics, all on technological paths that are converging – and incredibly relevant to the AEC industry. Recently I gave a presentation similar to this at an AIA conference where I was deliberating on the relevance of simulation, VR, AI, and considerations that we should take to use this technology early in the design process. The talk had some fallout from my professional colleagues and friends. Specifically, as I discussed the reality that the robots are coming, the audience received the talk with the wide eyes, and not that “look of wonder.” This was more of a look of “what and why,” more terror than fascination. I stopped a bit short in the presentation to have a conversation, specifically, to begin asking the audience questions:

- Is faster better?
- Is more data better?
- Will generative design improve quality – or will it displace our role in the process?
- Will digital fabrication improve our connection to construction?

As I do in the academy on a regular basis, perhaps the best answers lie in precedents:

Specifically, let’s tackle the “auto indicator” of Detroit from the 1980’s through the early 2000’s.

The Robots Are Coming:

To use a sporting phrase: the numbers don’t lie. Winsun is one of the first full scale 3d printing companies. They are not just experimenting with construction, they have signed an agreement to do 1.5 billion dollars of printing in Dubai, including 1.5 million dedicated to affordable housing. All of this work is to be completed in 5 years. On their **first** pass at full scale 3d printing:

- 1,100 square meters or 11,840 square feet for a total cost of \$160,000.
- This equates to \$13.51 per square foot.
- Assuming a low average cost on current construction methods in the United States, \$200 per foot for 11,840 square feet would yield a cost of \$2,368,000.
- That is a cost reduction of 93.2%. Or, nearly free buildings.



LEFT: WINSUN MANSION PROJECT. RIGHT: CURRENT WINSUN PROJECT

These results are not anomalous. Another major full scale 3d printing company, Apis Cor. has recently shared one of their recent full scale 3d printing projects. Of note, their 3d system is also portable – meaning they deliver the printing to a site, and print on site (Winsun sets up a factory system within delivery distance of their site). The numbers on their first house:

- 38 square meters – 410 square feet (about the size of a hotel room) for \$10,134.
- That is \$24.72 per square foot.
- Or a cost savings from a conservative estimate on a small structure – perhaps \$150 per square foot – \$61,500.
- This is a cost reduction of 83.5%.



APIS COR. HOUSE PROJECT – IMAGE FROM [HTTP://APIS-COR.COM/](http://apis-cor.com/)

The numbers do not lie. This is a massive disruption coming toward the construction industry. I challenge anyone reading (or listening in on the presentation) to persuade their developers to not take a +75% cost construction on their next project and stay with standard practices of construction. The robots are coming.

Antithetical point: How many jobs will this process crush? What does percentage of cost of construction mean now as a commission? What does automation of fabrication do to a community of designers and workers? The precedent is Detroit and the auto industry.

The Auto Indicator:

Construction was a 1.18 trillion-dollar industry in 2016 (Washington Post). The potential then exists that we could reduce the cost of construction by the average between the two precedent projects from Winsun and Apis Cor., a reduction of 88.35% on the cost of construction. This number though, to be honest, is just too absurd, and we can assume that both companies are politely slanting their numbers for the sake of marketing, so if we aggressively cut the number in half to 40% cost reductions we can wrap ourselves around some economic numbers and their impact. It also lets us set up a nice parallel (though be it a conservative one) between the construction industry and the auto industry which has already moved through the process of labor intensive construction to digital fabrication.

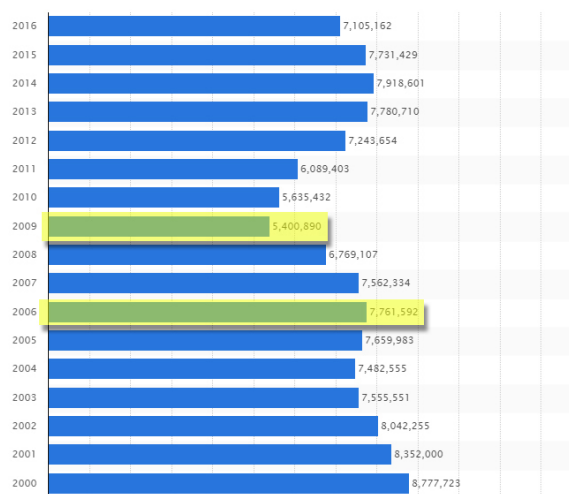
Current Ford CEO Mark Fields said of the 1980's:

"We (auto manufacturers) went through a huge transformation. . . . We had to say goodbye to almost 50 percent of the hourly workforce and almost 40 percent of the salaried workforce. The key thing about the transformation is that we did not miss a unit of

production during this time and quality went up.” (Cutcher-Gershenfeld, Brooks, and Mulloy 2015, 74)

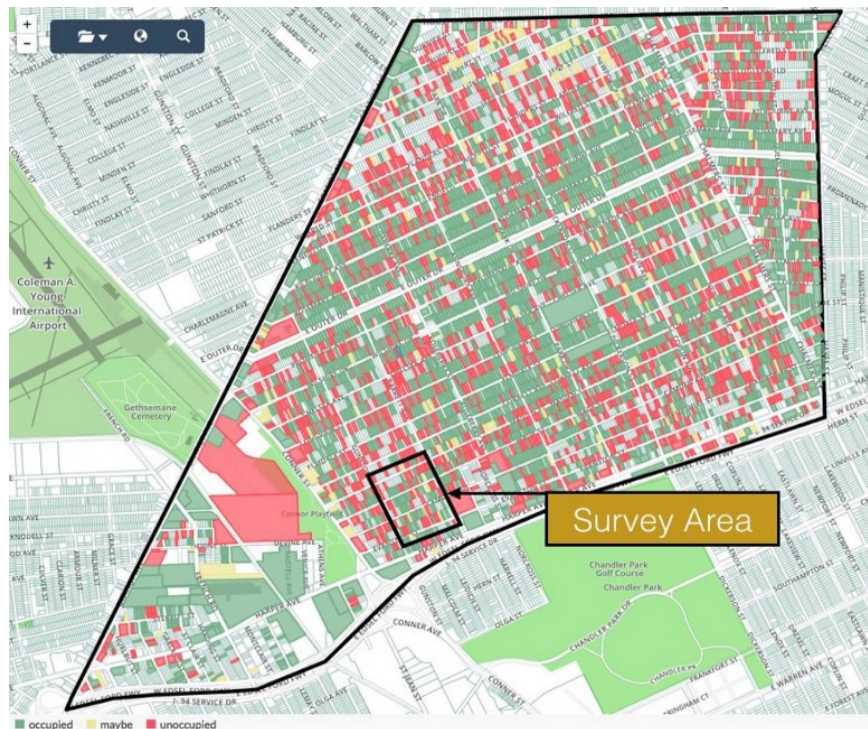
That is a wonderfully positive, corporate PC response to disruptive technology, but it is also clearly a terrifying spin for the labor force. And, it is also shockingly similar to the outline I just beginning to see in the construction industries future...reduction in labor while increase in quality and complexity, increase in production, reduction in costs. The auto industry recovered from the recession of the 80's as it implemented automated fabrication and moved forward, then saw the next in the early 2000's recession...which was a 40% dip in sales (between 2006 and 2009).

U.S. car sales from 1951 to 2016 (in units)



SOURCE: [HTTPS://WWW.STATISTA.COM/STATISTICS/199974/US-CAR-SALES-SINCE-1951/](https://www.statista.com/statistics/199974/us-car-sales-since-1951/)

In short, we have a reduction in the labor force, as well as a reduction of incoming finances at the rate of 40%. The urban result of this new socioeconomic paradigm shift is illustrated in the following map of a portion of Detroit. The red squares are unoccupied lots, many without a structure, some with abandoned residences. Around one fourth of the red lots are owned by banks, with almost no chance at any immediate return on their investment. These are neighborhoods not in trouble, but have had the very fabric of their community ripped apart. Without question there are more factors in the Detroit story than declining sales and automation to replace labor. The narrative is much wider: globalization, poor planning, race, and violence all played roles in the situation. Automation within the industry though, as indicated in the comments from CEO Mark Fields, is part of what has saved the auto industry, and has helped it to regrow from the latest recession, and it is also culpable in the reduction of half of the jobs.



SOURCE: LOVELAND DETROIT - [HTTPS://MAKELOVELAND.COM](https://make Loveland.com)

There are around 7 million workers specifically in construction.

(https://www.bls.gov/emp/ep_table_201.htm)

(<https://www.statista.com/statistics/192361/unadjusted-monthly-number-of-full-time-employees-in-the-us/>)

This relates to around 122 million workers in the US that rely on construction for the primary financial backing of their employment. That is 5.7% of our work force, which also influences countless other industries. The auto industry is 2.4 million employees, or right around 1.9% labor force, a number less than half of the construction industry.

Even if the reduction is less than half of the anticipated amount (reduction of incoming construction revenue could be as high as 80%, but this argument is staying in the more conservative 40% range), the result is reducing 1.18 trillion dollars down to 472 million dollars in a new construction economy.

We are talking about a massive reach into the economy. Continuing to parse numbers, just under 65% of the labor force in Detroit was tied directly to auto manufacturing. The resultant disruption has created a city where approximately one third of the is vacant (<http://www.michiganradio.org/post/detroit-has-tons-vacant-land-forty-square-miles>). This is a much higher number relative to a given location than construction as related to national distribution. Given that the construction industry is spread national, as 5.7% of our work force, the national map would only look 11.4% as bad. On a national scale, this means 1 in 20 lots become vacant (a higher percentage of residential than commercial lots). Not just available-vacant, but existing without **any** potential value for the immediate future. Imagine 1 in 20, gone,

distributed nationally, and that is using conservative numbers. This would be beyond unprecedented, beyond a disruption, this would shake an economy and further divide economic disparity to a potentially unrecoverable status.

Displace and Disrupt:

Do not take this out of context, because in reality I am a firm believer that we need this new reality. We can navigate past these points of depression on cover why the topics of why predesign thinking and simulation, digital fabrication, visual scripting, and AI are all relevant within this conversation. There are clear lessons from Detroit as the auto industry was centered around an economy that:

- Lacked of diversity in its market
- Lacked planning for the displacement of workers
- Lacked the education opportunities to prepare displaced workers for new/modified positions within their career
- Lacked leadership with sustained empathy to anticipate the socioeconomic impact of changes

I do believe the future of the construction industry is going to land in the hands of those I teach in our architecture program, and they are going to need to lead apply this precedent and improve their leadership with the AEC industry to:

- Diversify work and opportunity
- Plan for changing modalities of construction
- Help with the education of the construction community on the positive outcomes of digital fabrication
- Leadership within the AE and Construction community to begin navigating the disruptions coming

One of the questions we have to ask as a result of these changes is: If disruption is going to be so disruptive...why not pump the brakes a bit?

The base answers are that technology rarely works that way in business. Tech moves forward. Let's just look at the business of development – if the Return on Investment (ROI), is adjusted **massively** in the favor of reduced costs in construction for an improved product, developers are going to apply the tech. No question...finances along will drive large scale onsite digital fabrication. Financial reasoning alone is perhaps too simplistic and somewhat of a pessimistic view on how the technology will be applied. The reasons to apply the tech are likely much more profound as the industry of construction moves forward – some of which are arguably even more pessimistic, but let's wade through them anyway:

- We are gravitating to an open source world – the maker movement and mass customization are in line with the desire to build individual identities within converging globalization. Items that are expertly designed and capable of generating multiple authors are the key trend. This is showing up in fashion, gaming, product design and will certainly be part of both micro and macro components of architecture and construction.
- The need for mass customization is also approaching not just as a desire, but as a necessity. Given the recent outlook from the environmental studies conducted by the

EU: We are looking at a 1.8m rise in our oceans by the year 2100. As a result, we may/will be facing a massive refugee crisis due to climate change. The reality is we don't know exactly what this means other than we will need rapid, massive, highly customized, specialized solutions that must be generated through extremely cost effective, adaptable means. The map shows 1 million from Florida alone displaced, and another additional million refugees at minimum from the East coast alone. Now, this is not the population numbers of a displaced city, these are only the numbers of people who live on property that will be below sea level by 2100. The numbers could/will be much much higher.

- Finally, the design solutions we come up with won't last, we have already proven that most of our designs don't since new buildings have relatively very short life spans in the US...to meet changing needs, and the reduction of resources we will need to tear down and rebuild multiple times as new dwelling paradigms and new densities settle in. This is going to require rapid adaption through construction modification via digital fabrication.

So, these changes, disruptions, and displacements are coming. Perhaps they should. I don't particularly believe in absolute statements. However, I think it is safe to say that major change in both the way we construct, and the way we will inhabit designs will go through a dramatic shift over the next century, with the environment being one of many contributing factors. The exact path is a variable, but I do believe that ethical development, with an open source mentality, that is accomplished through rapid and adaptive of construction methods is our next reality.

So...how do we "teach it?"

Step 1 – Construction of Site Data:

Step by step video:

<https://youtu.be/rLYPMah0u6M>

Joseph Romm: "When just one percent of a project's upfront costs are spent...up to seventy percent of its life cycle costs may already be determined." Crucial to do sims at the beginning as designers! And this is why my favorite topic as an academic often revolve around predesign and early design thinking:

- Select site and data in CADMAPPER
- Import data into Revit as a dxf file
- Odd bits – but necessary to make things work:
 - Scale up by x1000
 - Delete all layers except topography
 - Export topography as dwg, reimport as dwg
 - Undo the "delete layers command" to get everything back (yes, we could use hide...I like this method better...not sure why to be perfectly honest)
 - Create topography file from dwg
 - Delete dxf layer of the imported topography mesh

- Delete the dwg layer of the imported topography mesh (we want to road lines, building outlines, and the Revit topography left)
- We now have a Revit topography, with building outlines and road centerlines as an underlay.
- Grab satellite photo from Google Maps if necessary, make sure you establish a distance measurement in Google Maps so you can scale the image appropriately.

*Note: we are building a file to run in parallel within multiple states – and this is always crucial to my workflow in a BIM world. I want this data to live in my physical/analogue world via fabrication, my BIM world (Revit), my predesign thinking world (FormIt), my VR world (Live and 3ds Max Interactive), my cloud (360), and be 100% available for collaboration via any of these avenues. This takes a bit of upfront work sometimes...not much, but some. It is more about attention to what you are getting/building than time. For instance, I want to build most of my context models rather than download them so they work in every instance, and have the level of detail in relation to the data I want to use as a designer. That is more time early on, but less trouble shooting for why something, perhaps, is not fabricating, or rendering correctly in VR, later.

*Note: Centering the site is more of an art than a science to right now this early in the design process. You will have to make assumptions, but you have to start somewhere, and starting with known quantities is better than beginning without context. I know that I want to work in Revit with my entry level floor at 0' on the elevation, so I always move the topography to the building site. I do not move the building to the topography. The Revit world works best (my opinion) when the building is at the center of the Revit world (the X,Y,Z, Cartesian Coordinate System). This will help my files line up across my multiple platforms, so as I move files from Revit to FormIt, FormIt to Revit, Revit to Live, and so on, everything should line up without additional modifications. I loathe reworking data...

Step 2 – Building Context in FormIt:

Step by step video:

<https://youtu.be/0aw-DM1h3V8>

1. Export from the 3d view as a .dwg file.
2. Import the .dwg file into FormIt.
3. In FormIt: Before I get too carried away, I want to make the 3d model in FormIt water tight – this will help ensure accurate 3d printing, and may make a difference in the CNC process depending on the software used to create a toolpath (we use PartWorks for our ShopBot, and to be honest, it doesn't care if the object is water tight or not, it just looks for a surface and builds a tool path).
4. Double click the topo surface in FormIt to edit the topo group
5. Double click the outer edge from a view below the topo to edit to geometry that form the four vertical sides under the topography – this will let you select the bottom lines of each piece.

6. Activate the Cover tool, select all four lines that make up the bottom edges, and then click the check box for the cover tool, this will close out the bottom edge of the topography geometry and create a water tight(ish) model.
7. There may be a few additional pieces to delete that get created from this process – notably above the edges of the top of the topo surface, just double click to edit the family, and delete the extra faces.
8. Create a new material and apply it to the topo model – this will allow you to actively adjust the transparency and opacity to see the satellite photo for reference below the topography model.
9. Use the location button in FormIt to establish your location, and import a satellite photo.
10. Move the topography model to line up with the imported satellite photo.

*Note: The satellite photo gives us that extra bit of information that doesn't exist in the Cad Mapper drawings – width of roads, foliage and additional site features, and it is an important step to verify that the data is concise and correct.

11. We are ready to start modeling existing buildings and conditions. Woot! The steps become less elaborate from here – it is always crucial to set things up correctly at the beginning of a project!

Context modeling is always a bit of a tricky task. There are routines to auto model, sometimes you can find models of existing building (InfraWorks360 is a great resource for that), and you can always use a variety of methods to build point-clouds and objects. All that said, I still prefer teaching a process of modeling the geometry. This reinforces modeling skills to my students, but more importantly, it builds a model that you know. I can be reasonably certain that everything I model will have minimal geometry (to run fast in VR), will be water tight (to do 3d printing), and will be modeled with the data needed in relation to my design process.

12. Save file. Delete the imported model so that we can work on a clean base. Save as.

*Note: FormIt, like many conceptual modelers, like to autosnap everything. This keeps your tasks moving quickly by making assumptions in a 3d drawing space. However, we need to turn off the topography for a moment to ensure we are modeling existing buildings with a flat base, on the ground plane, then we will move them vertically to the topography.

13. Trace out 3336 Washington Avenue.
14. Check Google Earth to establish height.
15. Do math (front edge and base heights).
16. Extrude base file in FormIt.
17. Google Maps – grab street view image.
18. Fix perspective in Photoshop
19. Set this as a texture.
20. Apply texture in FormIt and add detail.

*Note: No, this process is not precise to the inch. It is precise within a few feet. Where and when fractions of an inch matters, rely on your professionally created survey. For predesign work, and a basic understanding of context, everything done thus far in the steps is within acceptable tolerances.

*Note: I typically name recommend a naming convention on existing buildings that utilizes the address of the building. Using this method, combined with naming the geometry by the address

as well, will keep things reasonably organized and capable of being dissected by other team members working with the file.

*Note: Creating new geometry will remap the material to the new surface, making a bit of a mess of the elevation. Hold off on creating new geometry until you do not need the material for reference. To draw the guide lines correctly, the best practice is to start at on edge looking for the “on edge” snap. Draw your line across the surface and look for the “on face” or “perpendicular” snap on the second click of your line. Rogue lines (lines that are not coplanar with the elevation surface) will not behave in a predictable way as you modify the elevation surface.

Step 3 – Fabricating Context Models From FormIt:

Step by step video:

<https://youtu.be/2JapG0qEvYk>

1. In FormIt – check for water tight model.
2. Export as *.stl
3. Import into slicing software (Cura is used in this example)
4. Verify and modify scale
5. Verify layers and tool path.

Step 4 – Building PreDesign Simulations in FormIt:

Step by step video:

<https://youtu.be/F9OJX8gWWYs>

1. Establish site and weather stations.
2. Take particular note of wind directions.
3. Create massing based on square footage and potential programming.
4. Set levels – this will let FormIt and Insight know what volume to run the simulations on.
5. Run sun simulations to establish BTU's on surfaces.

It is never too early to start creating building data. At this point in the process you have a site, constraints, physical context, and something close to a building program. Given that data, we can begin to build some baseline studies on how a building will perform within the context. While very little has been done to validate the necessity of predesign simulation, climate expert Joseph Romm has observed through his research that “when just 1 percent of a project’s upfront costs are spent...up to 70 percent of its life cycle costs may already be determined.”

1. Turn off the topography layer. We will need to draw on the surface to make sure we are drawing a level base.

2. Sketch out the base (keep it simple, rectangles are good) of your building footprint paying special attention to square footage based off of your building program.
3. Double click the form, and create a group.
4. Under the object's properties (instance properties), give it a name and check the "use levels" box.
5. Under the Levels panel, set level heights, and names. Verify both the floor level and total square footage numbers in comparison to your program document for the building design.

*Note: Setting levels for an object in FormIt denotes the object as a design element to run simulations on. When there are context models within the workspace, having levels selected helps the software understand where to look for surfaces and interior spaces to form the building simulation.

6. Starting with an initial pass on Solar Analysis, double click any surface on your building instance and run the solar analysis.
7. For a location in the Midwest, observing the Month Peak setting for the initial reading is probably the most beneficial.

*Note: Temperate climates like St. Louis have temperatures that swing drastically from summer to winter, resulting in the need to block the sun through the extreme temperatures in the summer (July and August), while leveraging solar radiation in the coldest months (January and February).

8. Export several of the views as images for reference. We will revisit these surfaces and numbers later on in the process. These numbers will also be useful for building calculations on solar panel potentials (which is outside the scope of this class).

*Note: A BTU, or British Thermal Unit is the amount of heat required to raise the temperature of one pound of water by one degree Fahrenheit. An imprecise but simple, and relevant example that I use with my students to quantify this number is the rule of a light giving off 3.4 BTU's per watt. So, a 100 watt incandescent light bulb (not sure you can even find them anymore, but everyone can understand they are a "burn your fingers" kind of hot) gives off 340 BTU's. Looking at the example project in August, when temperatures in Kansas City are often at their most extreme, the South façade has a reading of 151 BTU/sq ft. You could imagine this as a total of 543,600 BTU's across the surface (120' long x 30' tall = 3600 sq ft. $3,600 \times 151 = 543,600$ BTU's) amounting to 1,599 100 watt incandescent light bulbs even spaced across the surface of a wall in August. The interior of the wall is trying to maintain a comfortable 75 degrees Fahrenheit while the exterior ambient temperature is in the mid 90's, with 1600 100 watt lightbulbs spread across the surface. This is a wall really battling the conditions, and a wall requiring design attention as it is also primary street front property with excellent views in and out. Compromise is a good word, and part of what a designer must consider in similar conditions.

9. Turn off the solar analysis tool, and be sure to open the file in the online version of FormIt.
10. Save your FormIt file to your A360 drive, and open the project in the web browser version of FormIt.
11. Select the location button and click on the adjacent weather station on the location map. This will provide you with both an annual wind rose, as well as monthly (seasonal) wind

roses where we can look at the direction and severity of wind. In a location like Kansas City, the winter winds are of a completely different quality than the summer wind. Do a screen capture of the key wind roses and save them to your project directory.

12. To start the energy analysis, hide all of the layers outside of your proposed design.
13. Click the Generate Insight button, and continue working on other things (perhaps an espresso) while waiting what is typically only a few minutes for the results.
14. Once completed, click View Insight to study the results and to begin working through values that might shape your design thinking.
15. Pay special attention to the following tiles:
 - a. Windows
 - b. Roof and wall constructions
 - c. Photovoltaics

*Note: Leveraging information from the simulation and data acquisition steps should help inform both you, and your client's decisions regarding several schematic decisions. On sites that are more open, several passes at the initial data should be done with the building in different configurations including rotation and varying length, width and height proportions.

Step 5 – DYNAMO!:

Step by step video:

<https://youtu.be/QEWp6mEtCyE>

1. *Dynamo!* Launch Dynamo.

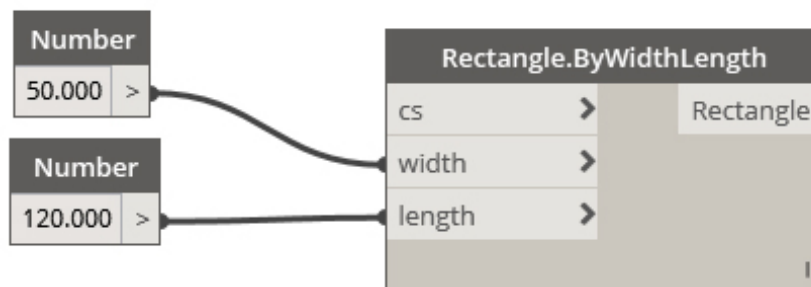
*Note 1: There are several versions (flavors) of Dynamo, including versions specifically for Revit, Web, and the standalone version Dynamo Studio. To leverage Dynamo within the FormIt environment your only current choice is Dynamo Studio. The studio version contains the process that allows us to upload a script to the cloud, and then utilize that script from the cloud in the FormIt workspace.

*Note 2: It should come as no surprise that diving into a visual programming environment is intimidating. I am someone that is on a similar learning curve, acknowledging that this mode of design, process, and computation is an extremely relevant part of moving forward with technology. One of the keys to learning Dynamo is developing a strong understanding of what the potential uses are by applying examples from relevant scripts, and how to repurpose what is already available. The following segments hope to both show how to create custom scripts, and work with packages published by others and available for use. For most users, you will be making minor modifications to relevant scripts that already exist. Having a basic understanding of how the scripts are functioning will make it relatively easy to pick apart an existing script and reapply it for your specific task. One key hurdle to get over on the quest to learn Dynamo is learning the nodes (a node is one of the predefined boxes in the Dynamo workspace with input and output ports). As you develop what you would like a script to do, finding the appropriate nodes takes practice, and the patience to build a new language of nodes with cause and effect. Finally, while this is speculation on my part, there is plenty of evidence available to suggest that visual scripting programs like Dynamo will be the bridge connecting designers to the effective use of artificial intelligence in design. Scripts will be able to set up the routines that select,

parse, and actively run routines versus a set of predefined variables to generate, or guide, a designer through options leveraging AI. This is again a bit of speculation on my part, but there is also, again, plenty of evidence to suggest that AI will be the technological disruption...not just a disruption.

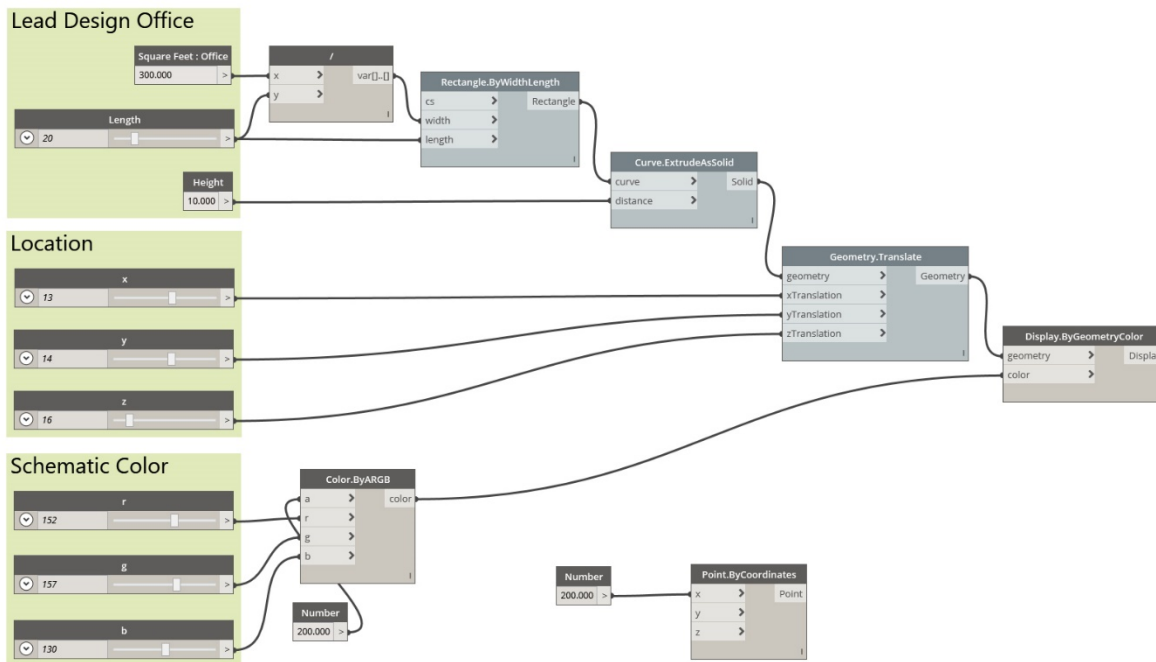
*Note 3: Especially for the designer that is learning to develop spatial programming, building a better diagram of spaces and their relationships is key in making successful designs. For me, the necessity of teaching this mode of thinking (building programming), with the immersing necessity of learning visual programming (Dynamo), creates a perfect opportunity. The following script was developed to introduce the two topics together with a programming/programming assignment:

2. Create the site base (site context to work within) by building the following script:



*Note: This script creates the basic rectangular space that all of the programmatic elements need to fit within. Dynamo Studio does not have specific units, but importing a script from Dynamo Studio to FormIt will convert your units to feet. So, 50x120 can safely be assumed as 50' x 120'. You also need to make an assumption regarding polar coordinates. I typically assume that positive X is north, and positive Y is west. A more complex site can be made from a series of points using the "Point.ByCoordinates" node and providing X and Y numerical inputs for each point.

3. Build the following script. Groups were created for space name and data, its location, and its color. The groups don't really follow any function other than providing visual clarity, and important part of visual programming.



4. Select all of the nodes, copy and paste, and overwrite the properties for the grouping title, and the square feet.

Note: This script is pretty simple. Take a number representing a required amount of square feet. Plug that number into a division node, and divide by an amount generated by a numerical slider. This creates a variable that is one half of the definition of a rectangle. The other half of the numerical definition is the slider (length) that was also the denominator in the division node. This creates an equation where no matter the length number in the slider, you always have the same square footage represented in the rectangle. The remaining nodes exist to help represent the required square footage as a three dimensional form with a specific color.

5. Use the location sliders to move the programmatic spaces along the X,Y,Z coordinate system. Add additional spaces as required, paying special attention to connective pieces like vertical circulation, appropriate adjacencies, and consider the type of space according to its location within the context and building design. The next step will be to push the Dynamo package to the cloud, and import the package into FormIt for testing within the context.
6. The last step within Dynamo is to save select File-Save to Web. This will send the file to your 360 account, and make the script available within FormIt.
7. In FormIt, open up the Dynamo window and update available scripts by selecting the Refresh button. Select your script and place it by clicking in the FormIt work space.
8. To edit the available values in FormIt, double click the geometry created by the script, and make sure that the properties window opens. All of the available values from the script will be available to edit in FormIt.

*Note: Make major modifications within Dynamo! The process is a bit slower in FormIt, and the variables are not as easy to navigate if you have a large number of options to manage.

Step 5a More Dynamos: NOTE: This was not included in my talk this year, but I thought it might be nice to include in the handout....

Watch the video on Dynamo and FormIt here:

<https://youtu.be/1X-WNda3H7E>

1. Load the Column Grid script by Josh Goldstein onto your 360 storage by opening the script in Dynamo Studio, and using the “save to web” function.

*Note: you can simply load a script from any shared location online. However, if you didn’t write that script, you may not be in control of it. Any changes made in the script will be transmitted to your FormIt file the next time you insert the Dynamo script into your workspace.

2. Adjust the attributes to begin working with the building form.
3. Continue adding column grid geometry to your FormIt workspace. Each one is an independent instance with its own variables.

*Note: you can work while the script generates modifications to the geometry. For working with something like a column grid script, I will collect a measurement that the structure needs to adapt to, modify a variable within the script, and while the geometry is updating I will gather the next measurement needed. Creating a flexible, adaptable structural system into a predesign process builds an absolute sense of joy for me. It is excellent to quickly iterate possible options to build an understanding of how structure can be implicitly related to, and yet independently expressed with enclosure and space.

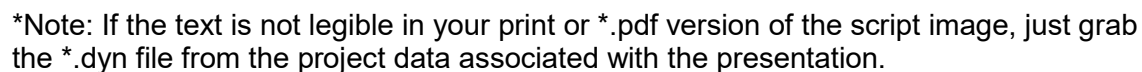
Step 6 – Fractal!

Step by step:

<https://youtu.be/d2LbaSPfDCA>

*Note: Building design often happens from two different ends of a spectrum – form follows function, and form elevates function. Teaching has taught me to be more of a valuest, specifically that I am happy when I can track a logical process of design thinking. So to approach this process from a different angle with a similar toolset, we can begin to study the enclosure system through a set of variables to create forms for evaluation. A key that should be noted with visual programming and Dynamo, the additional upfront work builds the opportunity for additional variables and iteration on the back end.

1. Build the following script in Dynamo Studio:



2. Select the nodes, and copy paste to add additional enclosure elements. Vary the height and width and length inputs. Verify that the sliders do not push the geometry out of the building site. Try to avoid conflicts that will cause the geometry to “fold in” on itself which will throw out a warning, and void your form.
3. Save the file to the web, and open up project fractal at:

4. Inside of Project Fractal, develop iterations of the basic form using the “random 10” options.
5. Peruse the options, download the models that warrant further exploration, and import them into FormIt for evaluation using File – Import 3d model tool.

Page 16

Step by step video:

<https://youtu.be/KND916pGKw8>

1. Push the Revit site file to Go Live
2. Open the *.lvsc file in 3ds Max Interactive
3. Push files from FormIt to 3ds Max using the *.obj format. (I love .fbx, but I have not had materials and scale carry over as consistently as the *.obj).
4. Once in Max, you may need to rotate (unless you always remember to adjust your settings), and scale up by 1200%.
5. Link the 3ds Max scene with the Interactive scene and use send all to build the *.fbx file and push the geometry into your Interactive scene.
6. Position, place, press play, click on the VR button. Gold!

*Note: As designers, we have been using the term scale as it applies to representation techniques wrong for decades. I did not realize this until we began working with Dev Kit 1 on the Oculus Rift. A perspective drawing or rendering provides proportion, VR provides an actual sense of scale displaying specifically how your body will relate to a design.