Generative Urban Design: A Collaboration Between Autodesk Research and Van Wijnen

Kean Walmsley
Complex Systems, Autodesk Research

Lorenzo Villaggi
The Living, an Autodesk Studio

Learning Objectives

- Discover the core concepts behind Generative Design
- See how Dynamo can be used with an optimization engine to implement Generative Design workflows for AEC
- Learn best practices for implementing complex, large-scale Dynamo graphs, bringing the data into Revit
- Learn how Van Wijnen has implemented generative urban design in the real world

Description

Autodesk Research’s in-house architectural studio, The Living, is a pioneer in the application of generative-design technology. They have completed successful projects for Airbus—the famous bionic partition for the A320—and internal to Autodesk, including both the design of the new office in Toronto’s MaRS Discovery District and the exhibit hall layout for last year’s Autodesk University. In late 2016, The Living was approached by Van Wijnen, an innovative Dutch construction company interested in reinventing the way the firm built homes. Van Wijnen wanted to apply generative design at the urban scale, exploring possibilities around the layout of residential areas based on financial, sociological, or environmental metrics. Speakers from Autodesk Research will present the concept behind the project and details of the Dynamo-based tool that Van Wijnen uses for new layouts. Additionally, a speaker from Van Wijnen will talk about how the tool is used and has changed the way the firm approaches urban planning projects.

Speakers

Kean Walmsley is a Platform Architect and Evangelist working for Autodesk Research. Blogs and tweets about developing with Forge, AutoCAD and other Autodesk technology, especially with respect to IoT, VR and AR.
https://keanw.com
https://twitter.com/keanw

Lorenzo Villaggi is a designer and research scientist at The Living, an Autodesk studio. His work focuses on generative design, new materials and novel forms of visualization. More recently he has been exploring space subdivision techniques for space planning problems as well as the
quantification of spatial experience for generative workflows for architecture. Lorenzo graduated from the Columbia University Graduate School of Architecture, Planning and Preservation with a Masters of Architecture where he has taught a graduate seminar and received his Bachelor of Architecture from the Politecnico di Milano. He has worked on several architectural projects including the new Autodesk offices at MaRS and the Laboratory for Embodied Computation in Princeton campus. His work has been exhibited in internationally renowned venues including the Chicago Biennial, the World Leadership Forum in Davos, the Milan Design Week, the MoMA and the New Museum. Lorenzo lives and works in New York City.

http://lorenzovillaggi.com
https://twitter.com/LorenzoVillaggi
Generative Design

Generative design is a framework for combining digital computation and human creativity to achieve results that would not otherwise be possible. It involves the integration of a rule-based geometric system, a series of measurable goals, and a system for automatically generating, evaluating, and evolving a very large number of design options. This approach offers many benefits for designing buildings and cities – including managing complexity, optimizing for specific criteria, incorporating a large amount of input from past projects and current requests, navigating trade-offs based on real data, structuring discussion among stake-holders about design features and project objectives, offering transparency about project assumptions, and offering a “live model” for post-occupancy adaptation. The framework consists of three main components: 1. **generate** a wide design space of possible solutions through a bespoke geometry system; 2. **evaluate** each solution through measurable goals; 3. **evolve** generations of designs through evolutionary computation.

**The generative design workflow**

Generative Design is a flexible and scalable framework. It can be applied to a wide range of design problems and scales: from industrial components all the way to buildings and cities.
Autodesk Research into Generative Design

The Bionic Partition | Generative Design for Manufacturing

The Bionic Partition is a next generation airplane component designed for Airbus through the application of generative design. It involved creating a custom algorithm using bio-logical rules and two measurables goals: weight and maximum displacement. The result is a metal 3D printed component that is almost 50% lighter and almost 10% stronger than traditional partitions.

Final metal 3D printed partition

Macro and micro optimization and metal 3D printing process
Architecture can often become a more challenging problem than engineering ones. In fact, architecture, unlike engineering projects, involves qualitative aspects of the experience of space that are less tangible and more difficult to measure. In 2017 The Living pushed the boundaries of generative design and applied this framework to architecture for the design of the new Autodesk offices in the MaRS Discovery District in Toronto. The geometric system incorporated several levels of constraints including the size of the space, the number of amenities and meeting rooms and fixed locations for cores and mechanical rooms. The goals combined qualitative aspects of human experience (such as ‘workstyle preferences’ and ‘adjacency preferences’) with quantitative measures (such as ‘daylight’, ‘buzz’ and ‘productivity’). The process allowed the designers to go beyond the one-size-fits-all type of approach to workspace design and offer a space that was diverse and rich in features. Through ongoing monitoring of the space and survey-based data collection, generative design can be used to suggest new design options and the scoring algorithms can be improved.

Design goals: adjacency preferences; workstyle preferences; buzz; productivity; daylight; views to outside.

Geometric system: (0) incorporate constraints; (1) define generative and non-generative zones; (2-3) subdivide space into neighborhoods; (4) generate ‘amenity bars’; (5) generate ‘test fit’; (6) assign teams; (7) evaluate solution.
Van Wijnen is an innovative Dutch development and construction company that seeks to change the way buildings are designed and made. In 2017 they partnered with The Living to apply generative design at the scale of the city. The project involved the design of a geometric model that could meet the local building code constraints (such as number and location of access streets, setbacks, parking rules etc.), and satisfy the developer’s requirements (such as amount of two-story residential units and apartment buildings). Urban design problems generally present many stakeholders, often representing conflicting requirements and interests, thus intensifying the complexity of the design. Generative design is able to aid the management and structuring of such complexity through the definition of the goals. In this case the project involved seven distinct goals, including financial ones (revenue and construction cost), environmental ones (such as solar gain and views), as well as more architectural ones (such as variety).

For urban design problems, the generative design framework can aid the management and structuring of complexity through the definition of goals that can represent the interest of different stakeholders.
Geometric system: (1) create mesh from boundary; (2) generate streets; (3) subdivide into lots; (4) place housing units; (5) place apartment buildings.

Evolutionary process.
Useful references

Hingston P. F., Barone L. C., Michalewicz Z., “Design by Evolution”
Benjamin D., “Beyond Efficiency”, in “Digital Workflows in Architecture”
Frazer J., “An Evolutionary Architecture”
Terzidis K., “Permutation Design”
Autodesk’s Collaboration with Van Wijnen

The collaboration between Autodesk and Van Wijnen started after Autodesk University 2016, where Van Wijnen was inspired to pursue generative design as a mechanism for improving their planning process for residential neighborhoods.

While the first phase of the project was important to establish whether generative design could be applicable to the company, the second phase is of more relevance to this session, as it was during this period that Autodesk Research created a tool based on Autodesk products that Van Wijnen could use internally to solve new urban planning challenges.

The products in question are Dynamo and Refinery. Most people will already be familiar with Dynamo: it’s the node-based, visual programming tool that’s commonly used for computational design.

The graph Autodesk Research created for Van Wijnen was extremely complex:

Much of the complexity of the above graph is hidden from view: we made extensive use of custom nodes as well as functions defined using both DesignScript and Python.
This is likely to be a typical pattern for companies wanting to use generative design on non-trivial problems, so this section is focused on some of the best practices that Autodesk Research discovered while working on the Van Wijnen project.

Make different layers – with different colors – responsible for different activities

The specific coloring doesn’t matter, particularly, but here’s what we used.

- The top layer is for the input of data, and contains orange groups of “constant” input nodes as well as turquoise groups of input nodes that are controlled by Refinery.
- The middle, green layer is for processing-related activities.
- The blue layer is for display. Only nodes in this layer are allowed to have their “Preview” flag set. (More later.)
- The grey layer is for Revit geometry output. (More later.)
- The red layer is for metrics. Only nodes in this layer are allowed to report to Refinery.

Inputs don’t all have to be at the left in this model. This is often helpful, as not all inputs are needed in the first phase, and so it’s more logical to spread the inputs across the graph to the places they’re needed.
Keep spaghetti wiring to a minimum: use code-blocks connectors to allow data to flow through the graph without making a mess

This proved very effective for passing data from phase to phase. The boundary, for instance, needed to move through various phases. Having a number of connected code-blocks helped give the freedom to reposition the wiring, as well as giving a clue as to what the wire was carrying.

Group functions logically, using copious use of categories with labels at maximum font size

Try to place nodes in logical groups, ideally with clear inputs and outputs. This helps in many ways, but also because it greatly increases the chances that common processes can eventually be separated out to add tools to your company's generative design toolbox.

Giving these groups large labels really helped increase legibility and the high-level navigability of the graph. Dynamo often hides text as you zoom out, so having large fonts on the main groups was extremely beneficial.
Maintain a left-to-right flow of data wherever possible

This helped in a few ways: in general to make it clear where data was flowing, but specifically when we started using the Warnamo tool to locate and manage warnings and errors.

Here’s a view of the graph where we have very little hope of quickly spotting the first error:

Warnamo lists the warnings and errors reported by the last run of the graph in a dialog that helps you navigate to the selected warning or error.
Clicking on the first item in the list takes you to the leftmost error in the graph (which should be the first, of the discipline of data flowing left to right is maintained).

Warnamo is available via the Dynamo Package Manager.
Nodes marked as “Preview” can only belong in the display layer

We wasted a significant amount of time tracking down nodes that were contributing unwanted graphics. In the end we imposed a rule that no nodes would be set to “Preview” unless they were part of the blue display layer.

We eventually took this a step further and implemented a mechanism that meant we only had a single “Preview” node in the whole graph: a list created from the various visible geometries. The List.Join node takes wires from all the different geometries – that are fed through from the rest of the graph via a series of wires and command-block connectors – and gives the freedom for the user to remove any that aren't needed by replacing their wire with one coming from List.Empty.
Control Revit export using another layer

Our export to Revit is managed in very similar way to the graphical geometry display mentioned above.

There’s a little more complexity here than we saw before – as well as a scaling step that can be useful when managing unit conversions – and a single ImportInstance.ByGeometries node to bring basic geometry into Revit: for now we just wanted to bring solid geometry across that might be used for high-level visualization (such as in VR).

Use of Refinery in the Van Wijnen project

If you’re interested in how Refinery can be used along with a graph such as the one shown in this session, we recommend attending, watching or reading the handout from the following AU 2018 class:

SD229074 - Getting Started with Generative Design for AEC