AS226214

Custom Computational Workflows for BIM Design Implementation

Eckart Schwerdtfeger
BIM Associate | Zaha Hadid Architects

Learning Objectives

- Understand how ZHA uses innovative custom workflows to implement BIM models for their signature fluid and parametric architecture
- Effectively automate cross-application workflows using custom tools, visual scripting and the cloud
- Create parametric freeform elements inside Revit + Dynamo using different methods like forms, adaptive components and T-Splines
- Develop efficient facade workflows utilising computational design, intelligent adaptive BIM elements and meta data analysis

Description

Zaha Hadid Architects are known for their signature fluid architecture and computational design. The transformation of these projects into Revit BIM models represents a special challenge and requires custom workflows.

During the last years, we have created BIM models for a number of iconic global projects and have used a variety of efficient and innovative methods that could enhance and inspire your work as well.

This presentation will explore practical examples that deal with the efficient cross-application transfer of geometry and metadata using custom software, visual scripting and the cloud. We will show how to utilise the different types of Revit families to generate parametric, rule-based geometry and freeform elements.

How can visual scripting in Dynamo enhance our computational design and how could an efficient façade workflow utilising computational design, intelligent BIM elements, and metadata analysis look like?

What challenges did we face and how did our methods evolve?

Speaker

Eckart Schwerdtfeger is an architect with 12 years of experience in all project phases and in implementing large-scale, high-profile international projects in Europe, Russia, the Middle East and China.

During his time at Behnisch Architects, Coop Himmelb(l)au and LAVA he parametrically designed, optimised and implemented several remarkable facades. His main fields of work were computational design, BIM management and programming.

Currently, Eckart is BIM Associate at Zaha Hadid Architects and responsible for managing the office's BIM team. Besides leading and implementing several large-scale BIM projects, he focuses on developing the ZHA global BIM workflows.

He develops and writes proprietary software, plugins and scripts to effectively facilitate, optimise and automate BIM and cross-application tasks and to enable the design team to contribute to the BIM models more directly and fluently.

Index

Introduction Aims

During the course of an architectural project, from concept design to documentation and implementation, a lot of different software is used. This e.g. depends on the type of project, the architect's scope, requirements and deliverables, even the skills and preferences of the individual team members.

Additionally, the design will change regularly, either due to testing of various options, to optimise and value engineer the initial design or to implement coordination changes and details in later phases.

To translate a design model into a proper BIM model and to coordinate both is a particular challenge. The design team should be enabled to contribute to the BIM model in a very direct and fluent way to save time and prevent issues.

Our main aims are:

Effective cross-application workflows from design to BIM

- Be able to use content created in 3rd party software
- Prevent duplicated work

Parametrise and automate as much as possible

- Enable + accelerate creation of design iterations and implementation of changes
- Save time spent on repetitive tasks

To facilitate this we use

Custom...

- BIM Elements
- Visual Scripts / Scripts / Macros
- Software / Plugins

Introduction Customisation Options

Every software provides a set of general tools that can be used to e.g. create a BIM model. However, this might not be enough for every office, certain projects, workflows or tasks that require special or custom solutions.

Fortunately, most software provides various possibilities to customise and extend it's base functionality, e.g. a visual scripting environment, macros or application programming interface [API].

Introduction Customisation Options

Customisation options for Revit + Dynamo:
[sorted by difficulty]

Revit

Family	Modelling Parametric geometry + conditions + formulas Revit family editor
Macro	Scripting [e.g. C#, VB.Net, Python, Ruby] Access to Revit programming interface Integrated editor, SharpDevelop
Plugin	Programming [e.g. C#, VB.Net] Standalone library with advanced capabilities External editor, e.g. Visual Studio

Dynamo

Visual Script Custom Node	Visual Scripting Graphical access to the programming interface Dynamo
Script	Scripting [e.g. DesignScript, Python] Access to Dynamo + Revit programming interface Integrated script nodes
	Programming [e.g. C#, VB.Net] Standalone library with advanced capabilities External editor, e.g. Visual Studio
Zero Touch Node Custom Node	Simple user-defined functions Custom user interface possible
Extension	Global access and extension of Dynamo possible

Forge

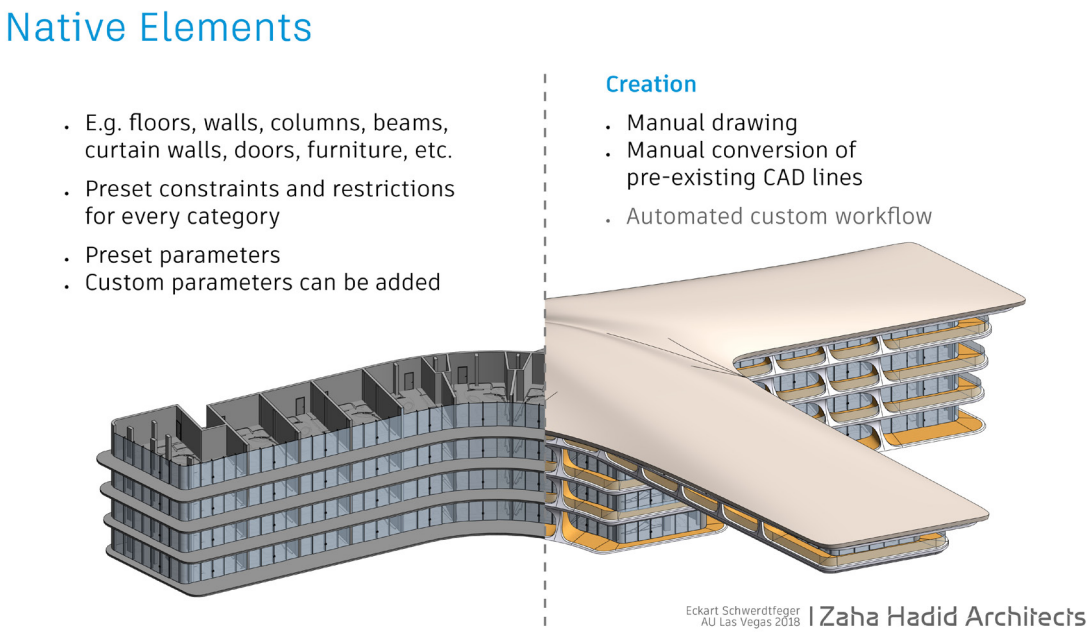
The Autodesk Forge cloud platform can be used for design automation as well. It currently operates on AutoCAD / DWG files but in 2019 Revit BIM models will be supported and can be automated, too.

Aims

- Parameterisation and automation of the design process
- Effective use of freeform geometry in BIM

Native Elements

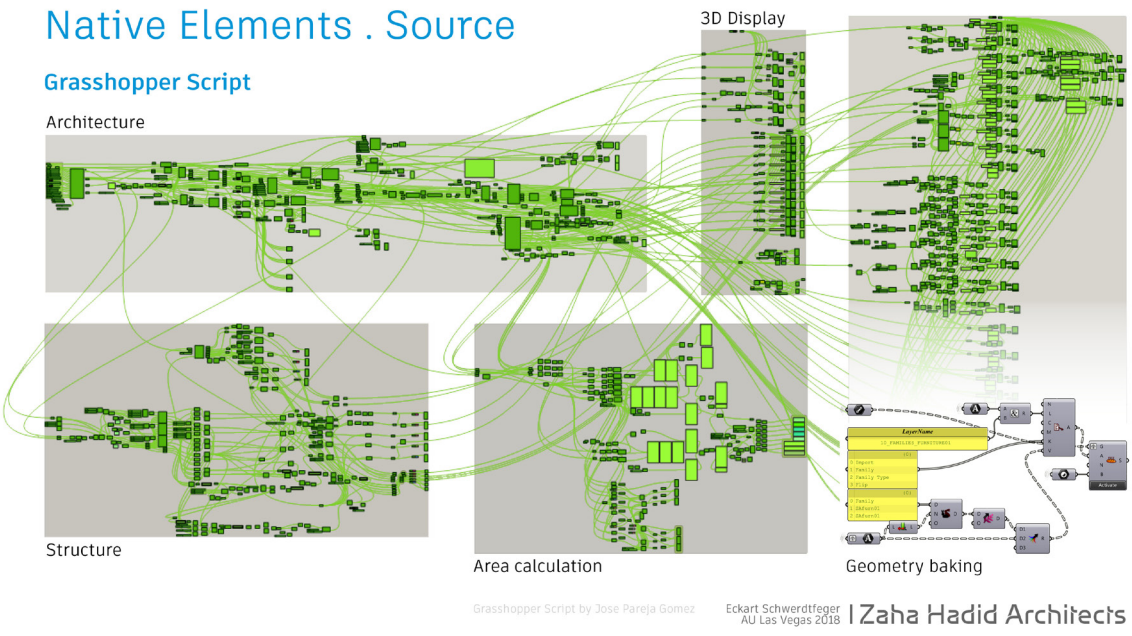
Revit provides a number of native elements that can be used for the BIM model out of the box and easily:



Native Elements

For some projects the manual drawing or conversion process might not be efficient enough to create a BIM model, e.g. due to scale of building, number of elements or design in general.

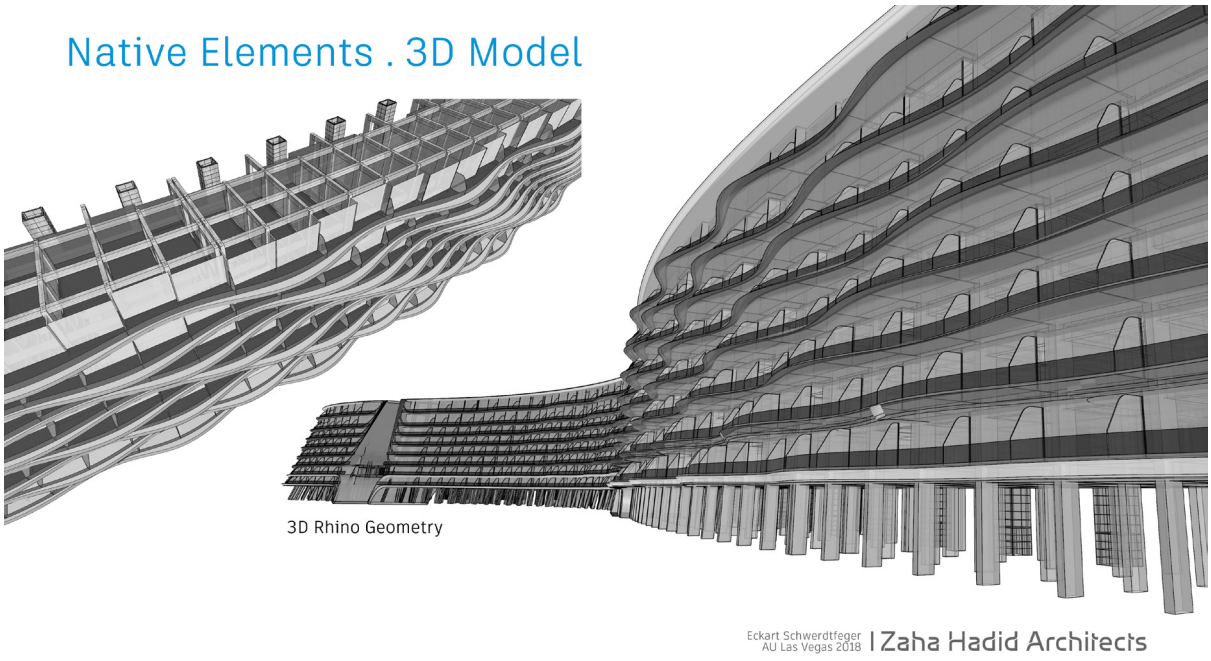
An automated process could be a much better or only solution, especially when the building elements are already available in some format or software and the design changes regularly and in quick succession, e.g. the following apartment buildings:

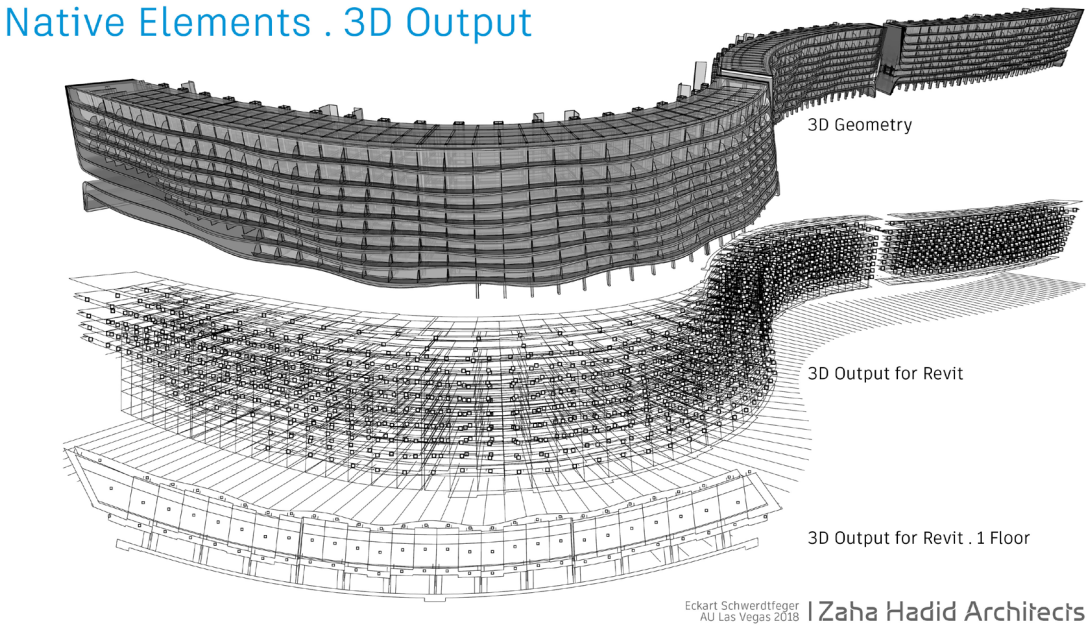


Native Elements

A comprehensive Grasshopper script already existed and was used for generating the detailed Rhino model. Hence, the most efficient way for creating the BIM model was to re-use this information and convert it into a format that can be transferred easily.

Only simple geometries like polylines [floor slab] or lines [wall] are needed to store the geometric information. Additional meta data is added to store element types, additional parameters and the information required by Revit.





ZHA BIM

In our office various applications [e.g. Maya, Grasshopper, Rhino, Catia] are used in the design process. Those different models are all collated in a Rhino Design Model which also acts as base for generating the Revit BIM + Documentation Model.

ZHA BIM was developed to efficiently translate geometry and meta data between the Design and BIM model and is comprised of two C# plugins for Rhino [to attach meta data to geometry] + Revit [to automate the creation of BIM elements].

ZHA BIM . Introduction

To Transfer an element cross-application we need

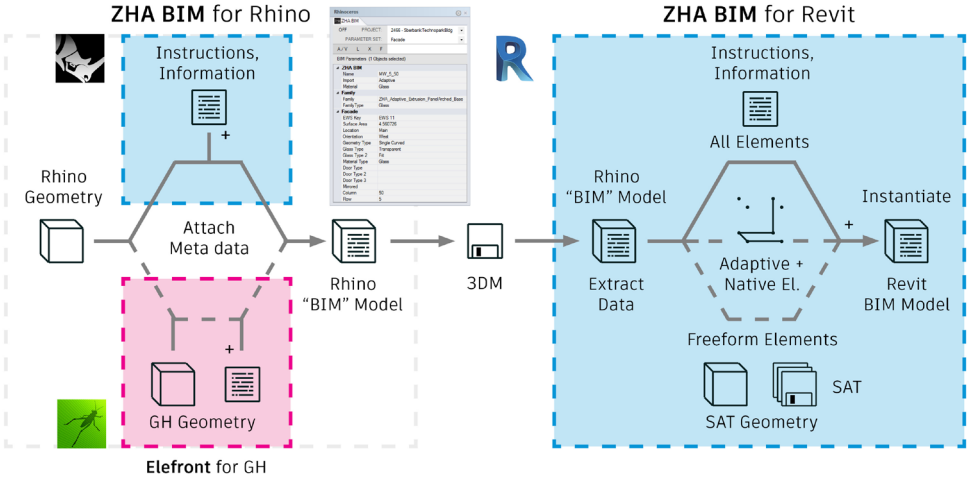
Geometry
of element or it's underlying geometry

Instructions
for transfer process and targeted software

Building element information
Parameters, meta data [optional]

Eckart Schwerdtfeger AU Las Vegas 2018 | Zaha Hadid Architects

ZHA BIM . Process

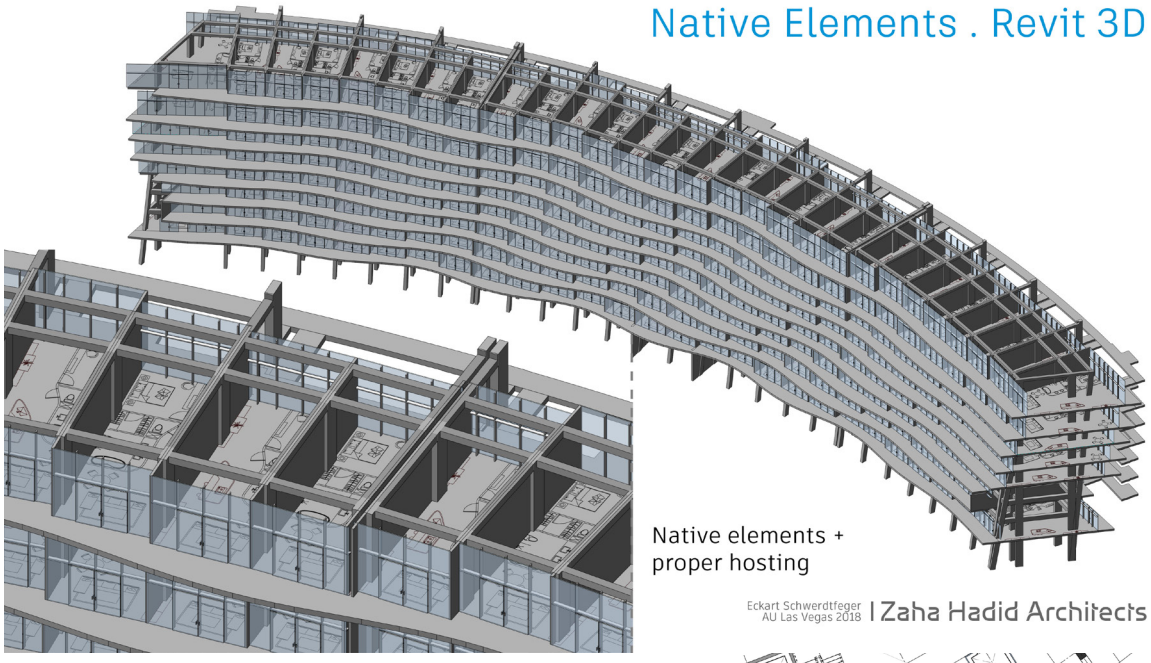


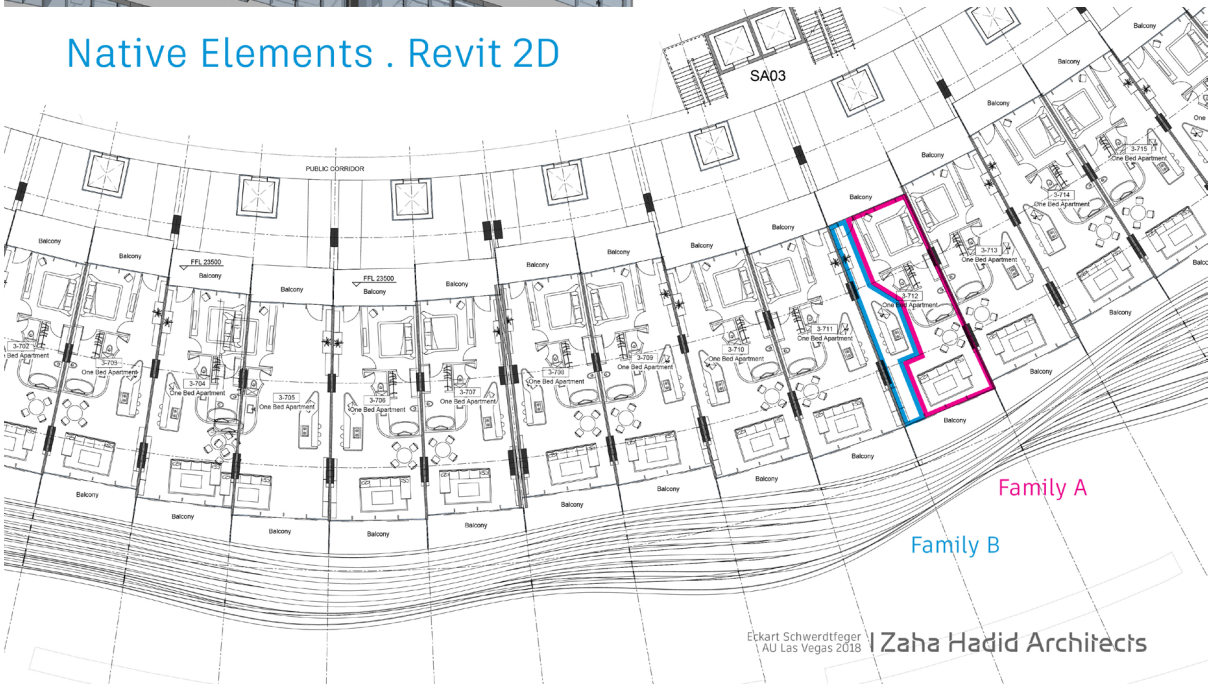
Eckart Schwerdtfeger AU Las Vegas 2018 | Zaha Hadid Architects

Native Elements

The ZHA BIM plugin takes the geometry and meta data that was generated by Grasshopper and simulates the work of a user in a sequential order and predefined steps so all BIM elements can be created and hosted properly.

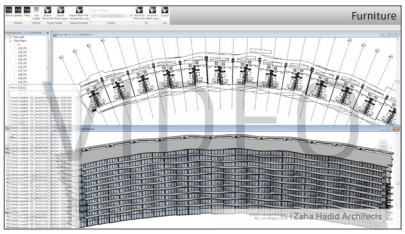
The automated process takes about 15 minutes for the below example and afterwards a proper Revit model and drawings have been generated. Only some custom situations that were not included in the Grasshopper script are left to be adjusted.





The video shows how the Revit plugin simulates the work of a user in a specific sequence required for proper hosting of elements and based on the information read from the Rhino file that was generated by the above Grasshopper script:

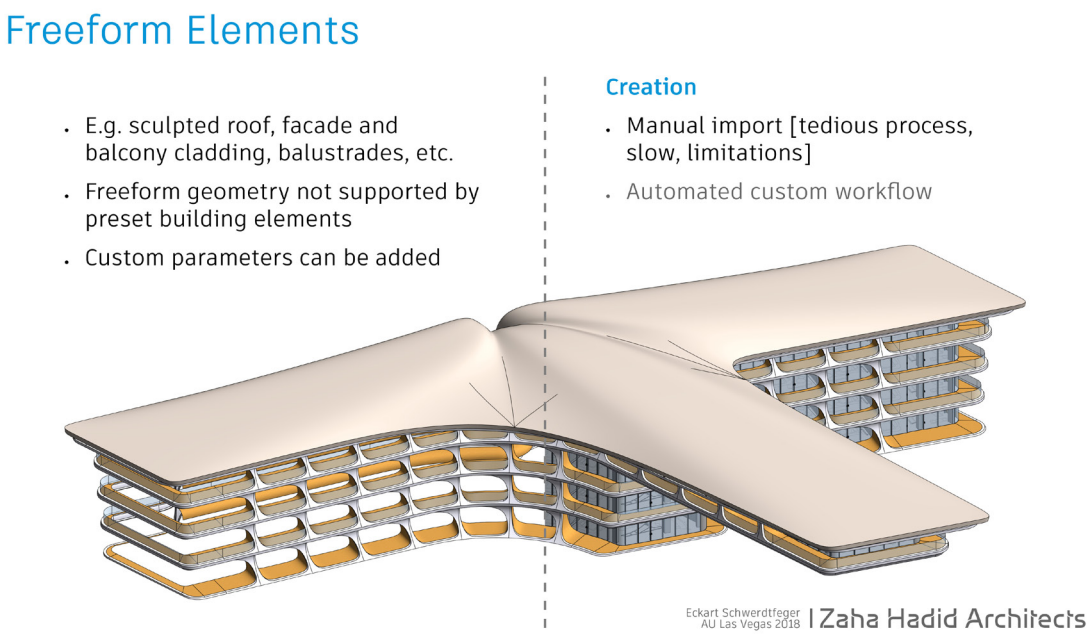
https://vimeo.com/eckart/aulasvegas2018-01



Freeform Elements

Standard Revit building elements do not support freeform geometry. This kind of elements need to be loaded into generic model families and processed afterwards to properly enable most of the existing Revit features.

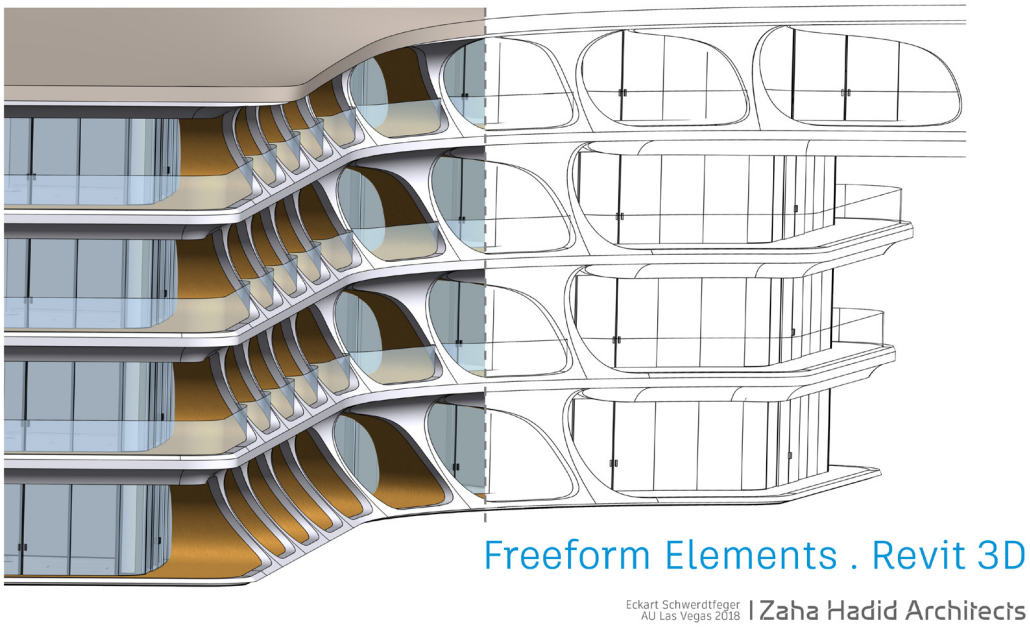
The buildings contained a lot of sculpted facades, roofs and balustrades, hence an efficient workflow was required to create the BIM model:



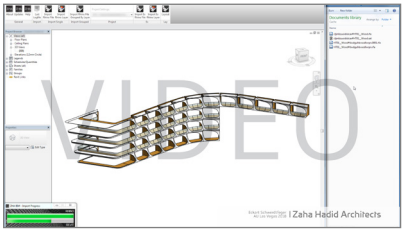
Translating freeform elements into Revit families is a rather work-intensive task that requires up to a dozen successive steps a user would need to undertake for each element, e.g. creating and placing the families, dealing with SAT files, setting materials and parameters, etc.

A task like this gets even more difficult when hundreds or thousands of elements, e.g. facade panels, need to be created or transferred.

All required steps to create freeform elements in Revit can rather easily be programmed into a C# Revit addin like ZHA BIM or Dynamo visual script to automate and accelerate this repetitive and tedious task.



The following video shows the used process. In Rhino the required and custom parameters are set, attached to the geometry and saved. Then the Revit plugin is used to simulate the different steps a user would need to undertake for each element and transfers them into proper Revit families:



https://vimeo.com/eckart/aulasvegas2018-02

Aims

- Parametric Revit elements instead of importing “dead” geometry
- Intensified use of visual scripting / Dynamo

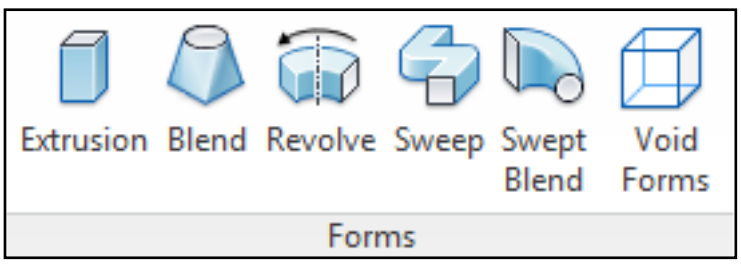
Freeform Elements In Revit

Importing freeform geometry with the ZHA BIM plugin is very fast and simple. However, the resulting elements are “dead” and cannot be edited or changed inside the BIM model later.

In the beginning of a project imported geometry is very helpful to quickly create a BIM model and start the coordination process with the consultants. But in the course of the project as many elements as possible should be done natively and using intelligent Revit families.

Revit is able to create certain rule-based geometry natively and fully parametrically. It takes a bit longer to set this up but afterwards changes can be implemented
directly and very efficiently.

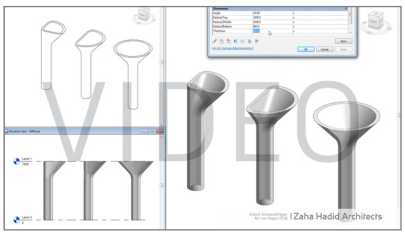
The following methods for creating forms are provided:



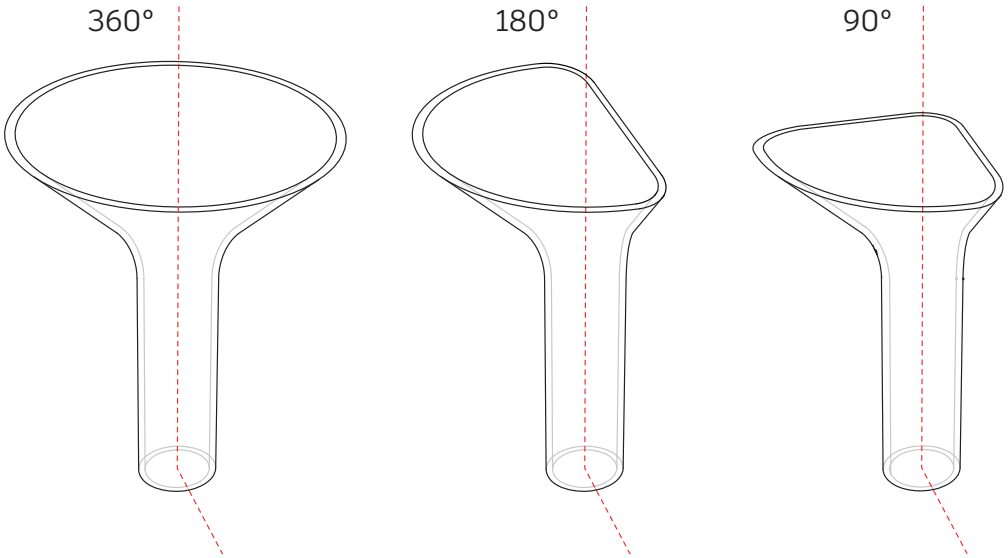
Freeform Elements In Revit Forms

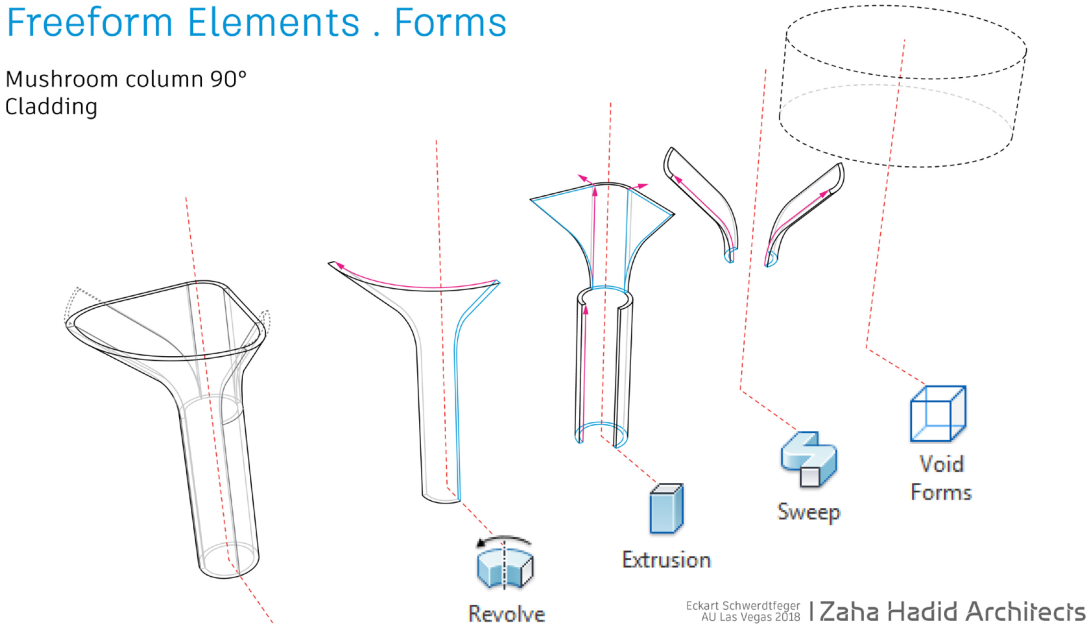
Even rather elaborate forms can be created inside Revit in a fully parametric way in case they can be broken down into individual pieces that follow rules that Revit understands.

In a final step the individual pieces can be joined together into a consistent form and Revit family that can easily be adjusted by parameter values.



Example - Mushroom column cladding:
https://vimeo.com/eckart/aulasvegas2018-04

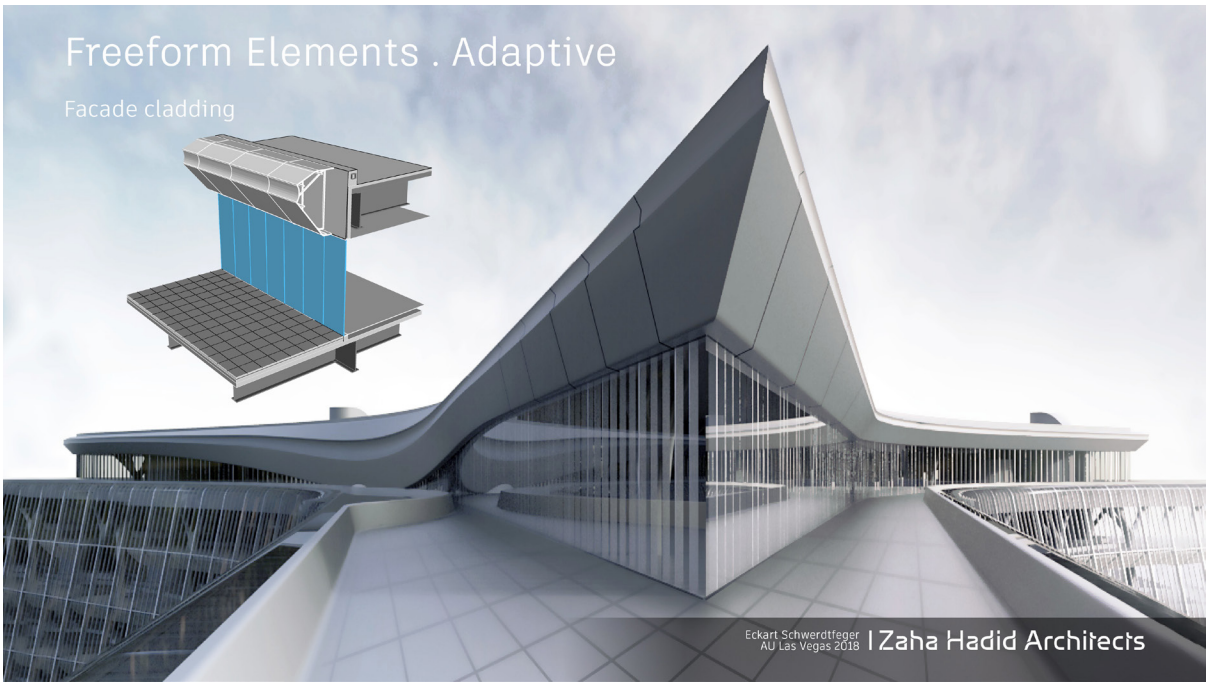


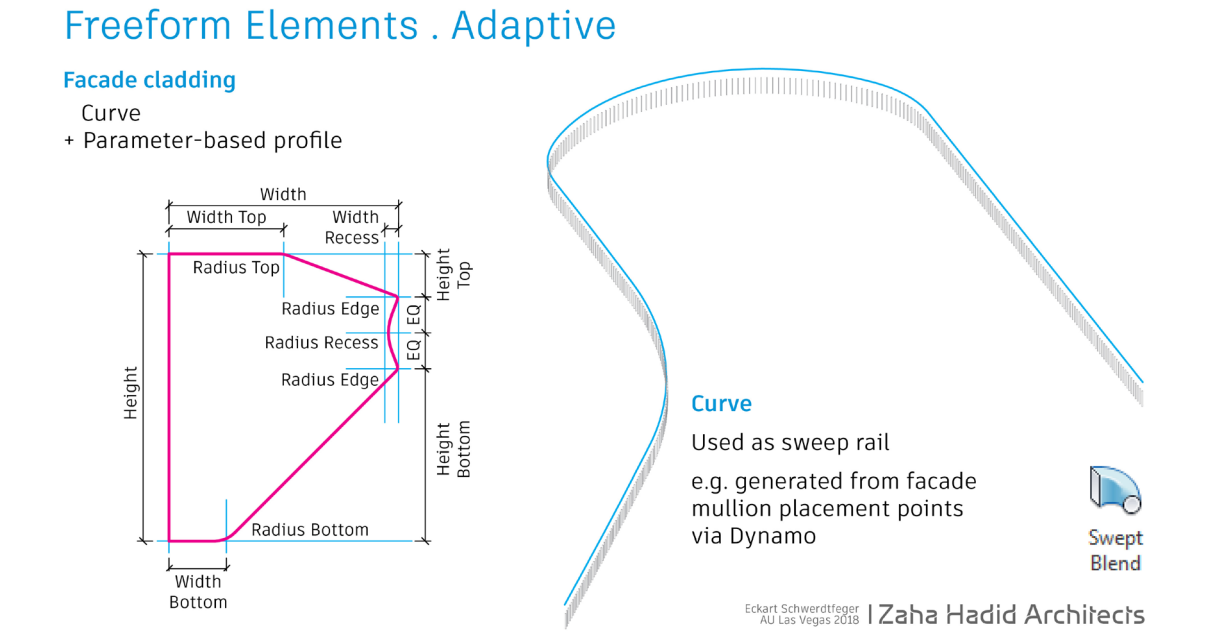


Freeform Elements In Revit Adaptive

Adaptive families prove to be very useful for generating rule-based freeform elements as well. They use points as input and with the help of parameters, formulas, constraints, auxiliary elements and nested families additional geometry can be created in a fully parametric way and fed into the Revit form generation tools from above.

Example - Freeform facade cladding:





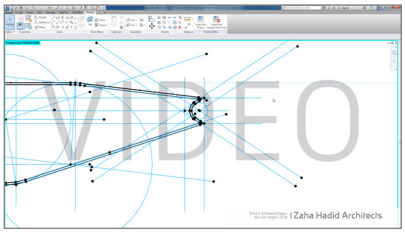
Freeform Elements In Revit Adaptive

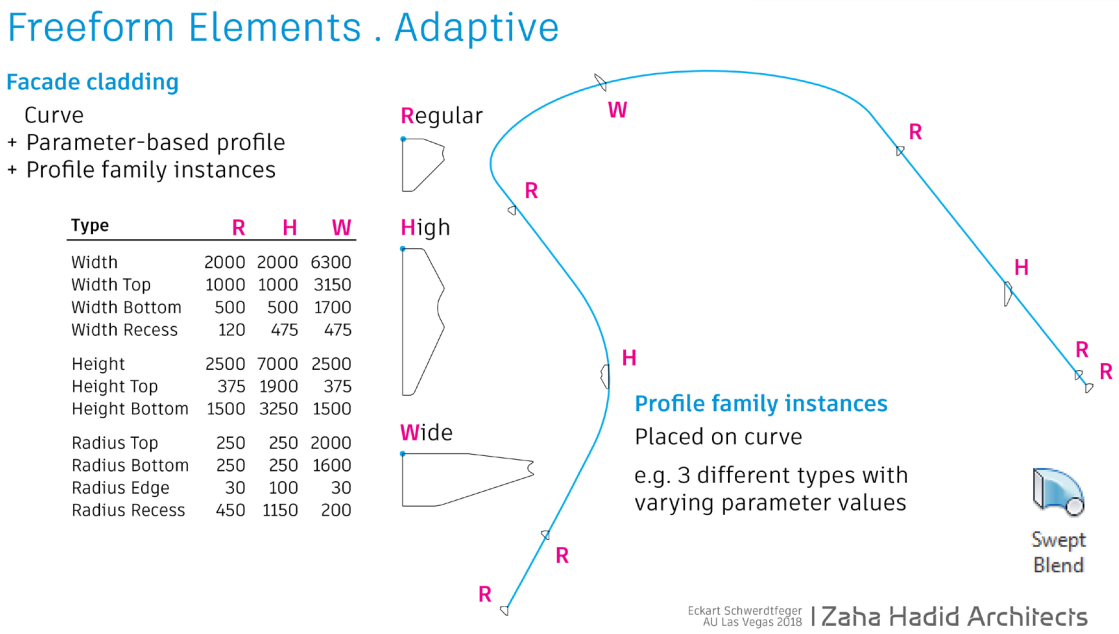
As usual, the first question is how to split the element into smaller parts or steps and to find the underlying rules.

A swept blend should be suitable to generate a form like this. The shape of the roof continously changes but it always follows a certain geometry which can be turned into a fully parametric profile in Revit. Even the parametric fillets between the flat sections can be constructed in an adaptive family.

This again requires a bit of setting up and auxilliary work first but will be very useful in the course of the project:

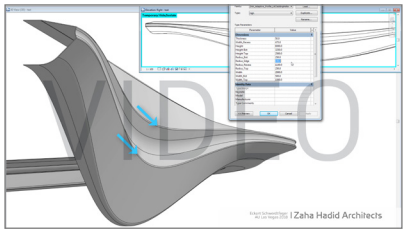
https://vimeo.com/eckart/aulasvegas2018-05





Three different profile types are defined [matching the Rhino design model] and placed on a curve that was extracted from the facade mullions of the BIM model via Dynamo.

That's all the input required to generate a swept blend in Revit and the best part is that changing the underlying profiles immediately changes the facade cladding as well. Hence the family can be used to test design options inside Revit or to easily adjust the shape to design changes later:



https://vimeo.com/eckart/aulasvegas2018-06

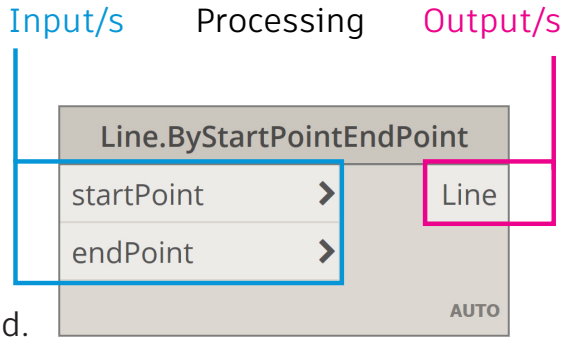
Dynamo Visual Scripting

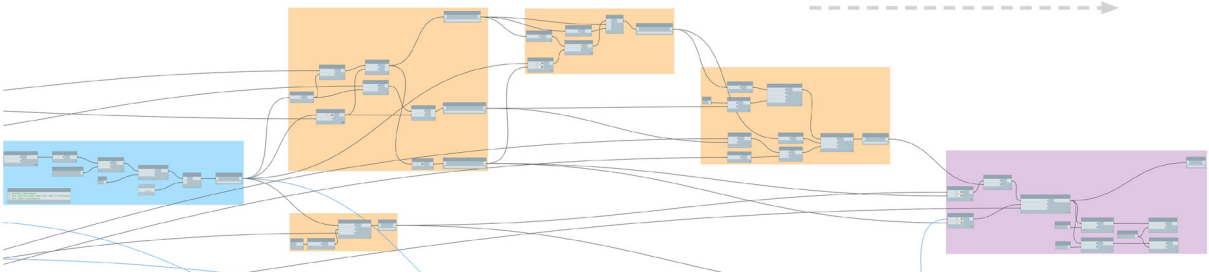
Dynamo is a graphical user interface for scripts based on the Dynamo + Revit application programming interface [API].

With “visual scripting” custom workflows and processes can be created rather easily and without having to learn a programming language first.

Each script represents a linear sequence of nodes. A node can receive geometry or data via one or multiple input/s, processes them in a predefined way and generates one or multiple output/s.

Inputs and outputs are connected using wires and this defines the sequence of commands that is executed.





Tree Column Structure

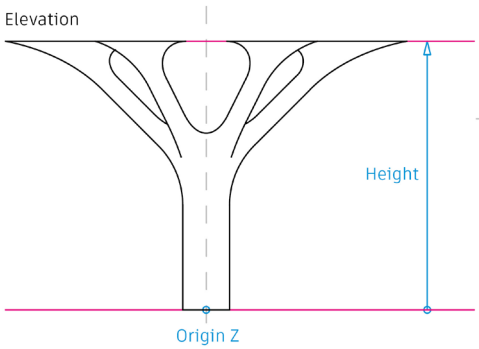
Dynamo is a Revit plugin, hence the open building model is fully accessible, can be analysed and processed in a Dynamo script.

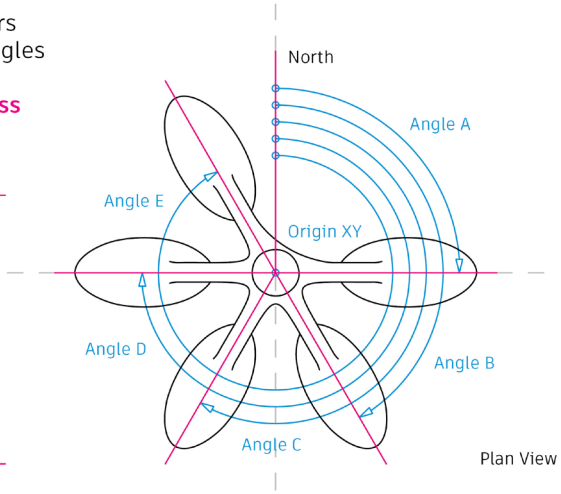
E.g. the tree column structure of the project can be generated based on the live building model and is automatically correct. This is difficult to achieve when multiple 3D models exist and have to be coordinated manually.

Dynamo . Tree Columns

Process

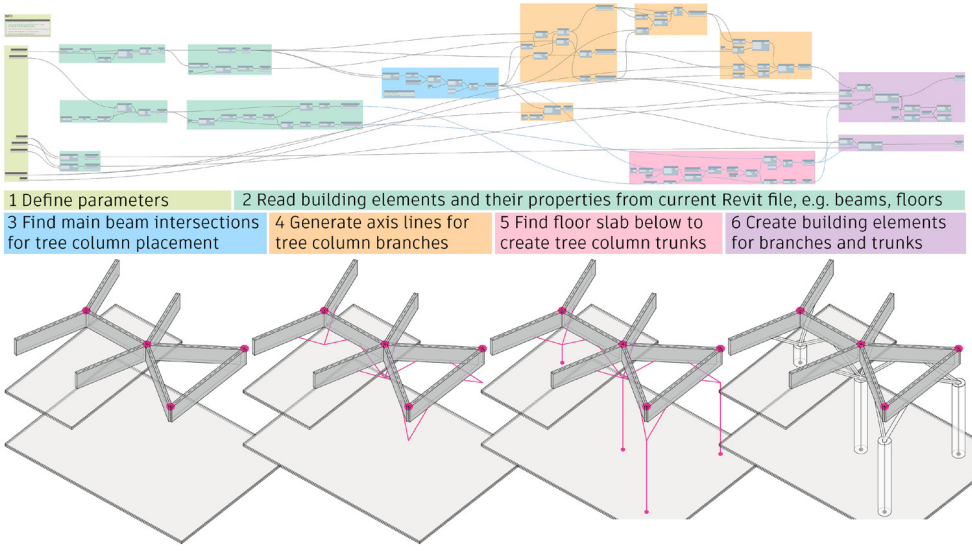
- Analyse **live** building model - main beams, floors
- Collect + sort meta data - origin XYZ, height, angles
- Store meta data - csv spreadsheet file
- **Generate structure - automated Dynamo process**
- **...**



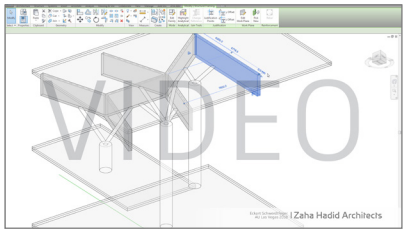


Eckart Schwerdtfeger AU Las Vegas 2018 | Zaha Hadid Architects

Dynamo . Tree Column Structure



Eckart Schwerdtfeger AU Las Vegas 2018 | Zaha Hadid Architects

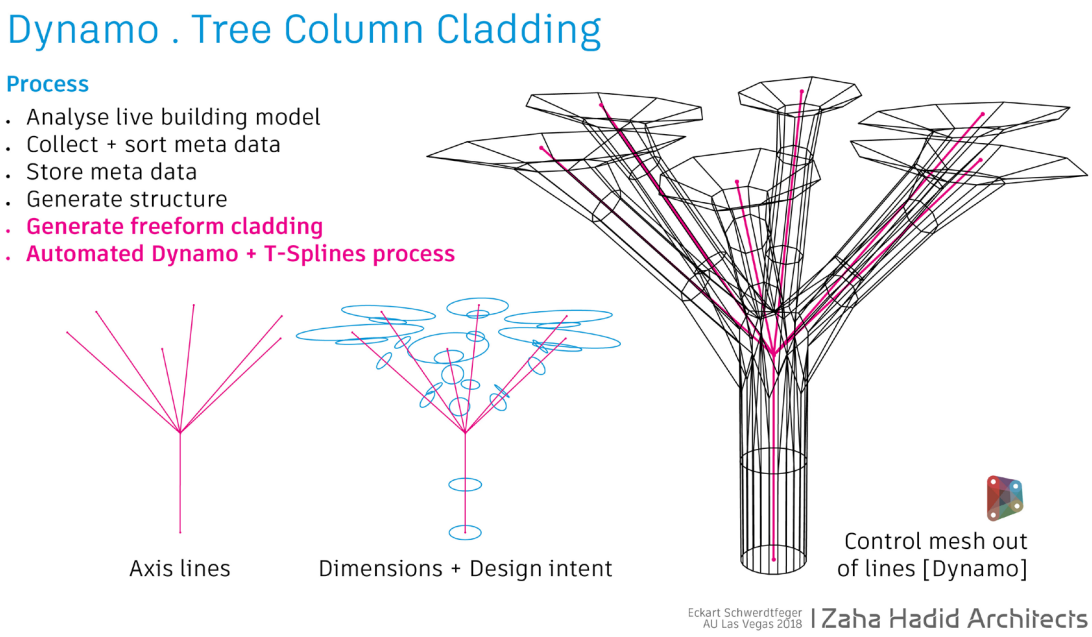


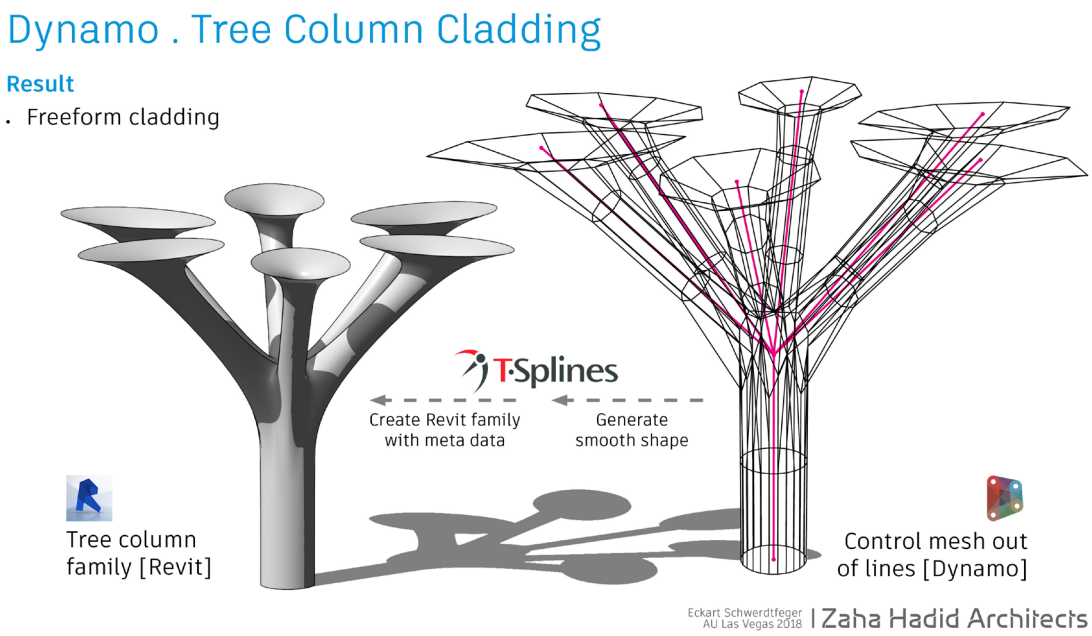
https://vimeo.com/eckart/aulasvegas2018-07

Dynamo Visual Scripting T-Splines

Using the T-Splines functionality, Dynamo has the ability to translate low poly meshes into smooth freeform geometry and "bake" them into Revit BIM elements. The geometry within this families cannot be edited directly but the tedious process of importing 3rd party geometry is no longer required and the result is usually much cleaner, too.

This process was used for the 400 custom tree columns in the building. Each individual tree configuration and cladding was calculated based on the location and direction of structural beams in their vicinity.





Dynamo Visual Scripting T-Splines

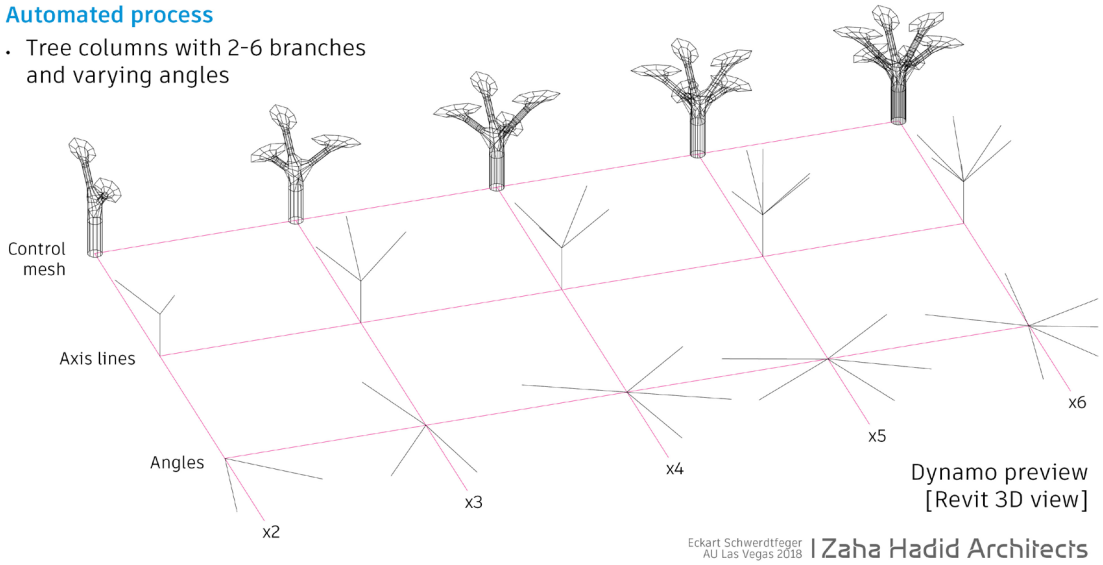
The Dynamo script generates control meshes for the T-Splines plugin individually and based on the geometry and meta data collected from the live building model.

A custom Dynamo node makes sure the calculation happens in a sequential order so each column is generated in a different pass and the software does not get overwhelmed with data or runs out of memory.

Dynamo . Tree Column Cladding

Automated process

- Tree columns with 2-6 branches and varying angles



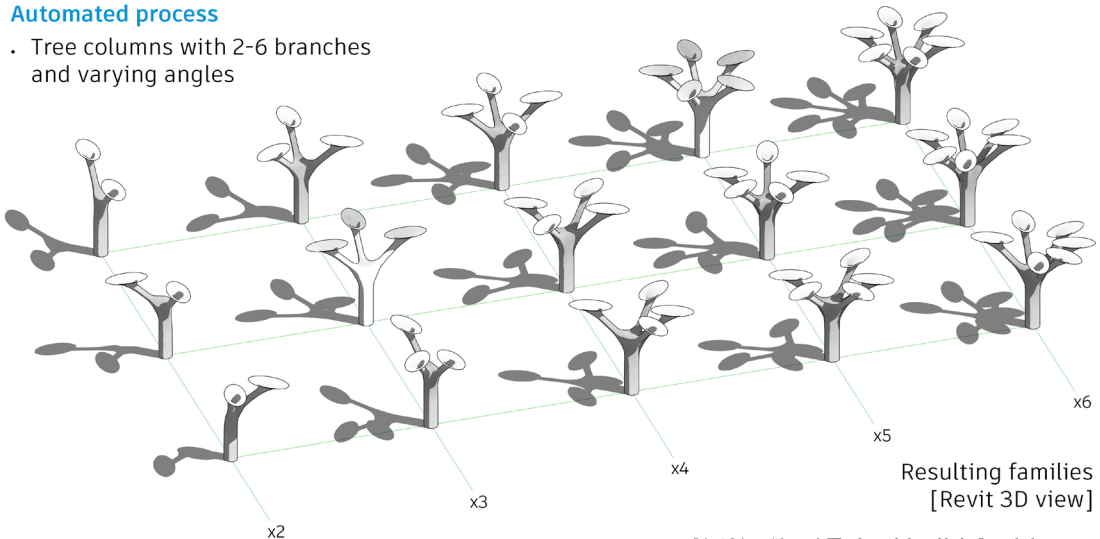
Dynamo preview
[Revit 3D view]

Eckart Schwerdtfeger
AU Las Vegas 2018 | Zaha Hadid Architects

Dynamo . Tree Column Cladding

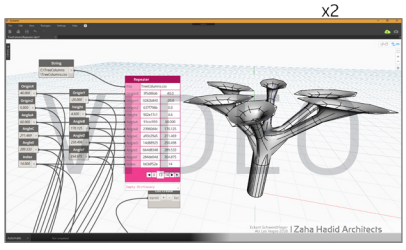
Automated process

- Tree columns with 2-6 branches and varying angles



Resulting families
[Revit 3D view]

Eckart Schwerdtfeger
AU Las Vegas 2018 | Zaha Hadid Architects

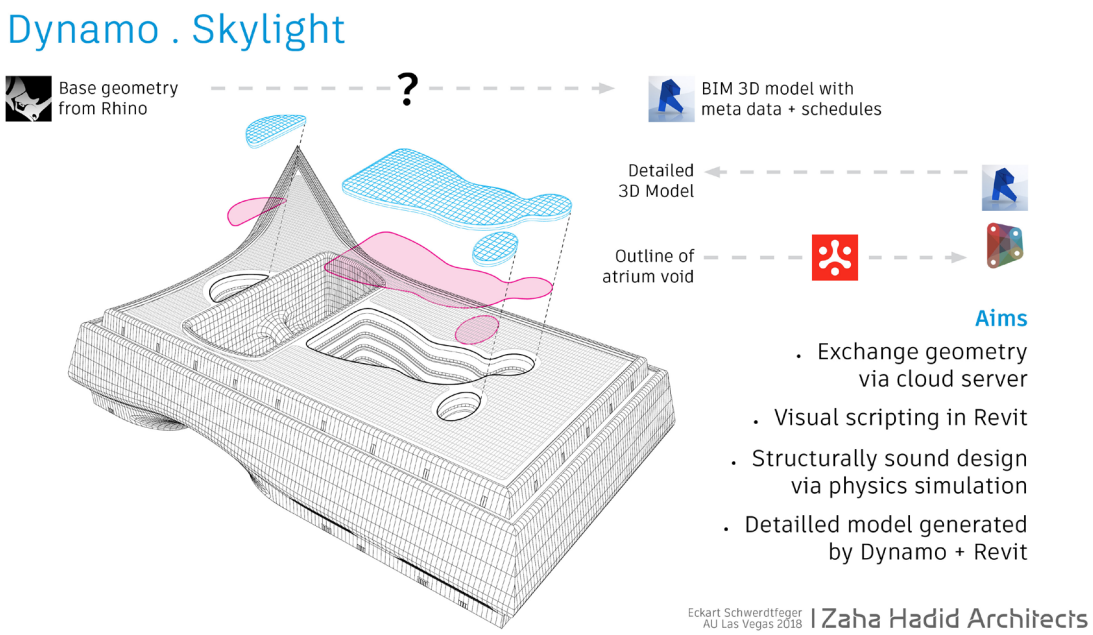


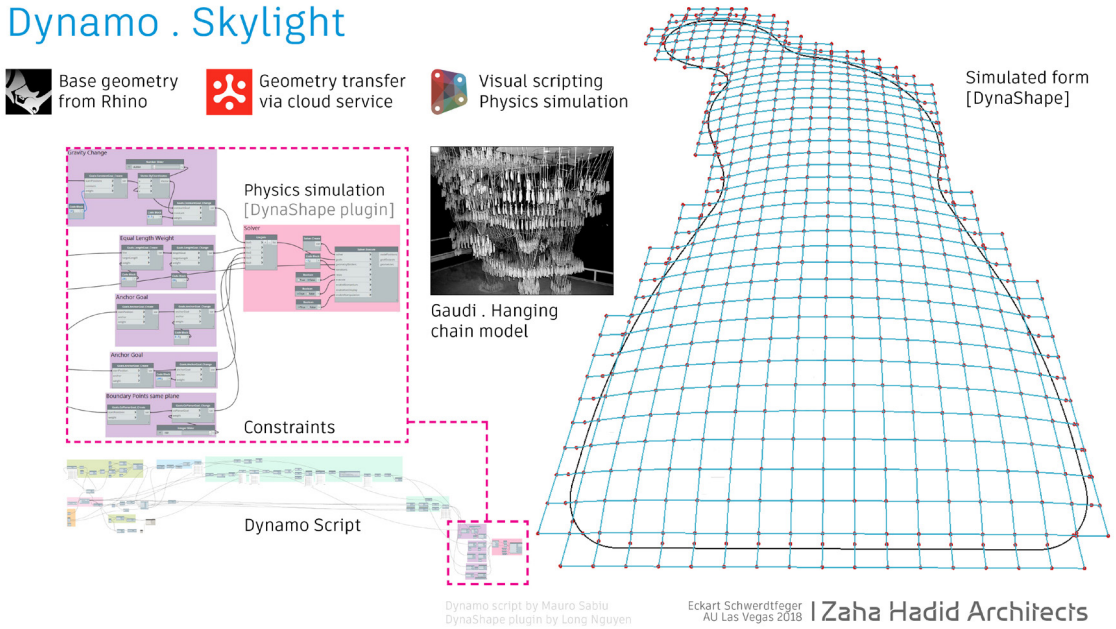
https://vimeo.com/eckart/aulasvegas2018-08

Dynamo Visual Scripting DynaShape

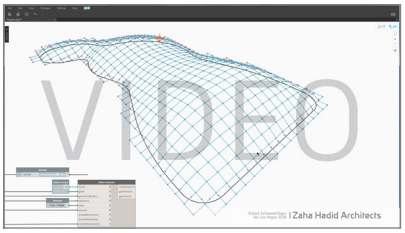
DynaShape is a Dynamo package for constraint-based form finding, optimisation and physics simulation by Long Nguyen. It allows a user to model complex curved forms that are based on certain physical or structural types (e.g. catenary shells, tensile structures) or that meet certain geometric properties (e.g. planar quad panels, unrollable surface stripes, as-uniform-as-possible edge length, etc.).

DynaShape was used for the atrium skylights to generate a structurally sound design in a process reminiscent of Gaudi's hanging chain models:





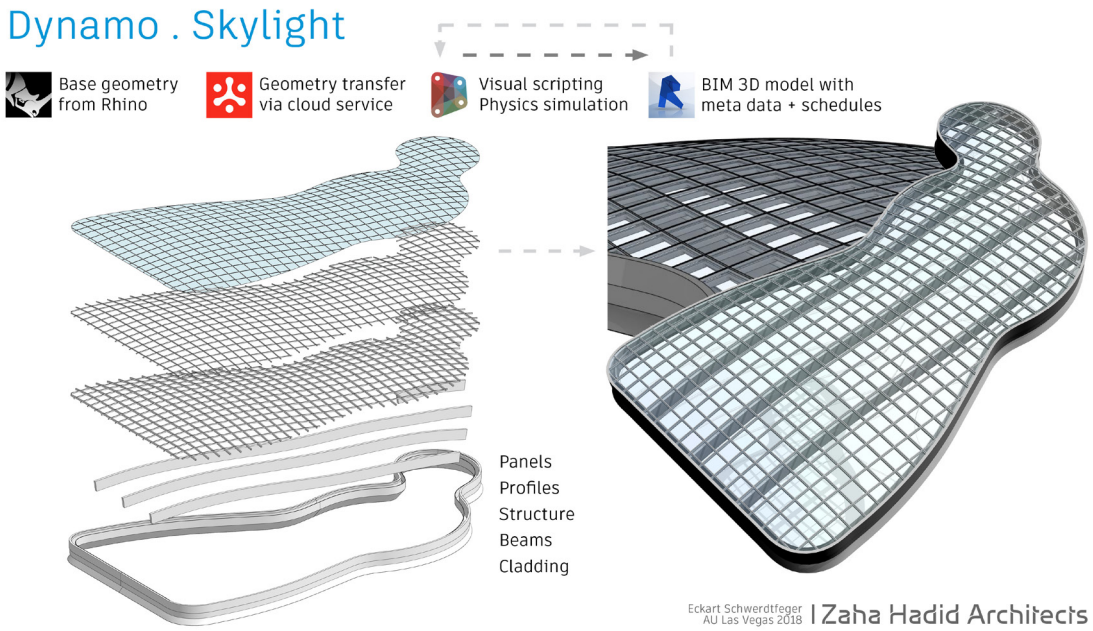
Dynamo Visual Scripting DynaShape



The DynaShape plugin simulates the given constraints of the Dynamo script and negotiates an optimised result:

https://vimeo.com/eckart/aulasvegas2018-09

The user can freeze this result at any time and then use it as input for other parts of the Dynamo script which e.g. generate the geometry and native building elements for the Skylight.



Dynamo Visual Scripting Custom Nodes

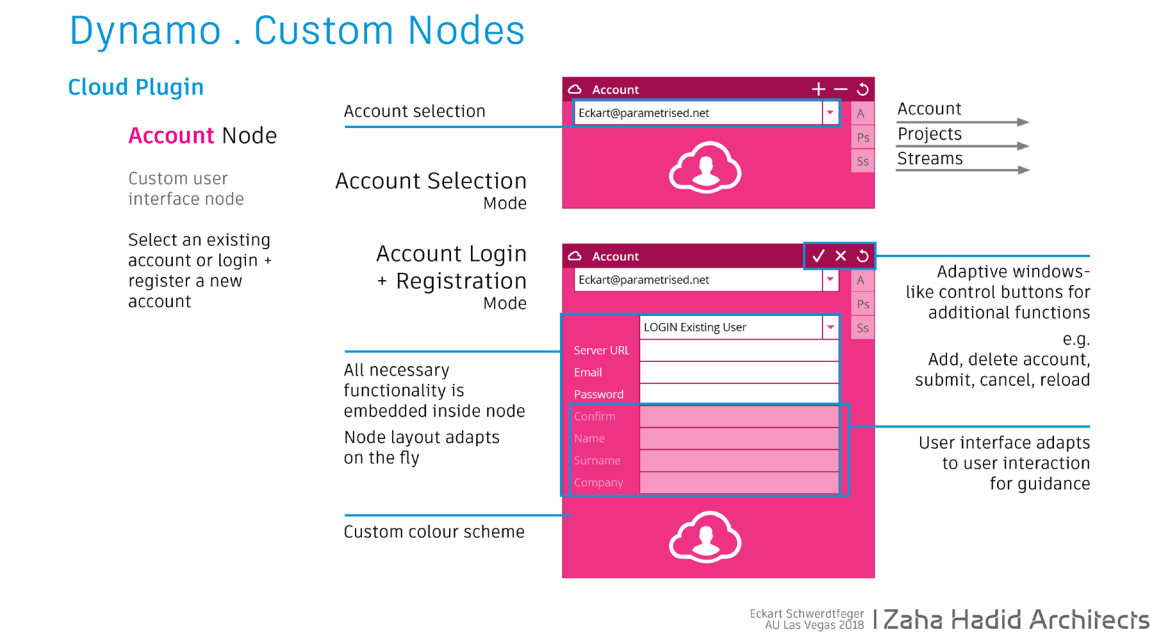
In some cases out-of-the-box Dynamo nodes might not be suitable or enough for dealing with custom tasks. Fortunately, Dynamo is highly flexible and can be extended using many different methods and matching the very different skills the users bring along.

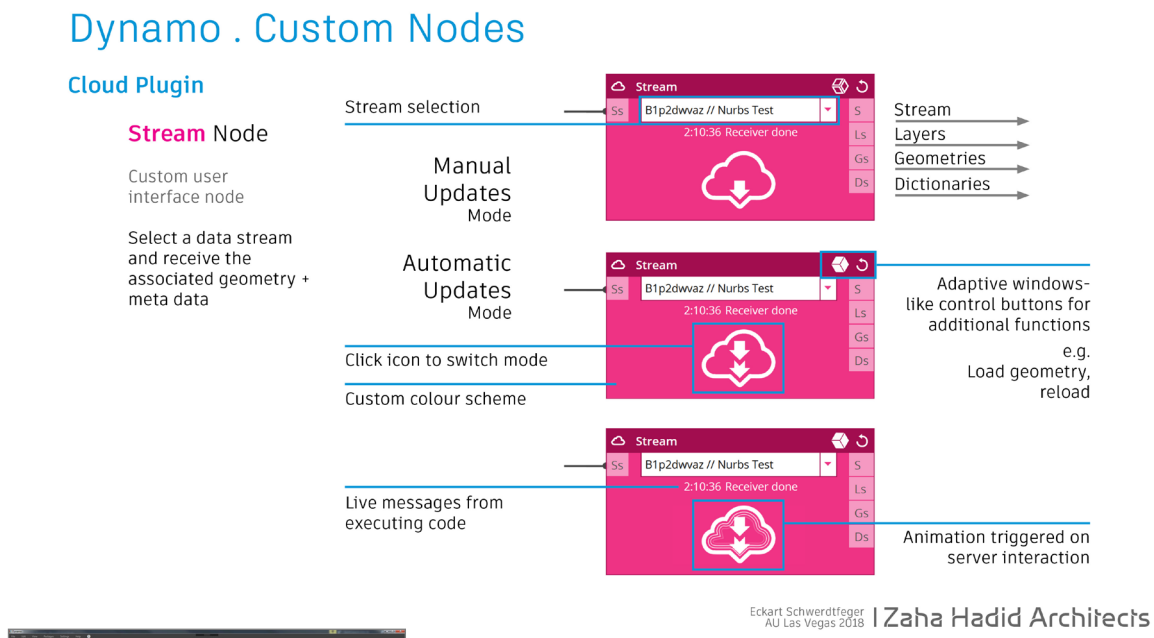
Visual scripts can be packaged into custom nodes right inside Dynamo. If advanced capabilities are required a standalone editor like Visual Studio and a programming language like C# can be used.

Zero Touch Nodes are the easiest to program and implement, while custom nodes can contain an individual user interface as well. Additional knowledge of the Windows Presentation Foundation [WPF] graphical subsystem is required to use this capabilities properly.

Dynamo Visual Scripting Custom Nodes

A number of custom user interface nodes were developed as part of a research. The aim was to try out what is possible, to innovate and create more efficient and adaptable nodes, e.g. with putting all the necessary functionality and features into the nodes itself and have the node layout and content adapt to the users input or requirements as needed.





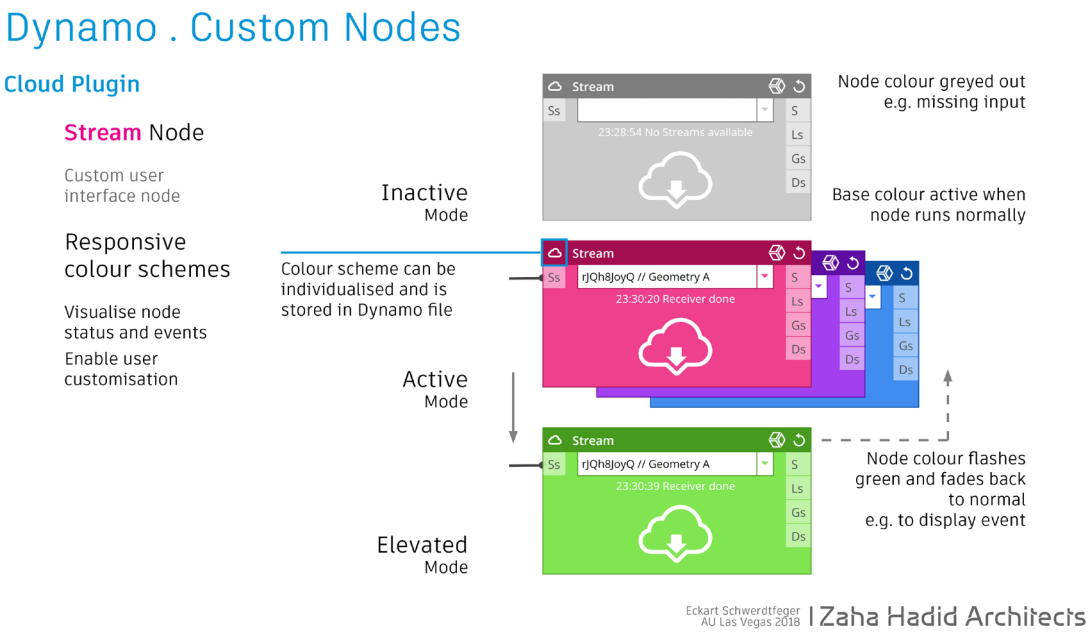
https://vimeo.com/eckart/aulasvegas2018-10
https://vimeo.com/eckart/aulasvegas2018-11

Dynamo Visual Scripting Custom Nodes

In general, Dynamo scripts and nodes are displayed using the Windows Presentation Foundation [WPF] graphical subsystem that is responsible for rendering Windows user interfaces.

Therefore, programmers can use the features of WPF and its XML-based language to create custom styles and interfaces. A programming software like Visual Studio can be used as a graphical editor and to layout this custom forms. However, some coding and setting up of the XML files and resources [e.g. images] will be necessary as well.

Dynamo allows the programmers to place this individual WPF elements in the central area of a custom node. However, with advanced knowledge of the WPF subsystem even global adjustments like changing the colour of the node can be implemented.



https://vimeo.com/eckart/aulasvegas2018-12

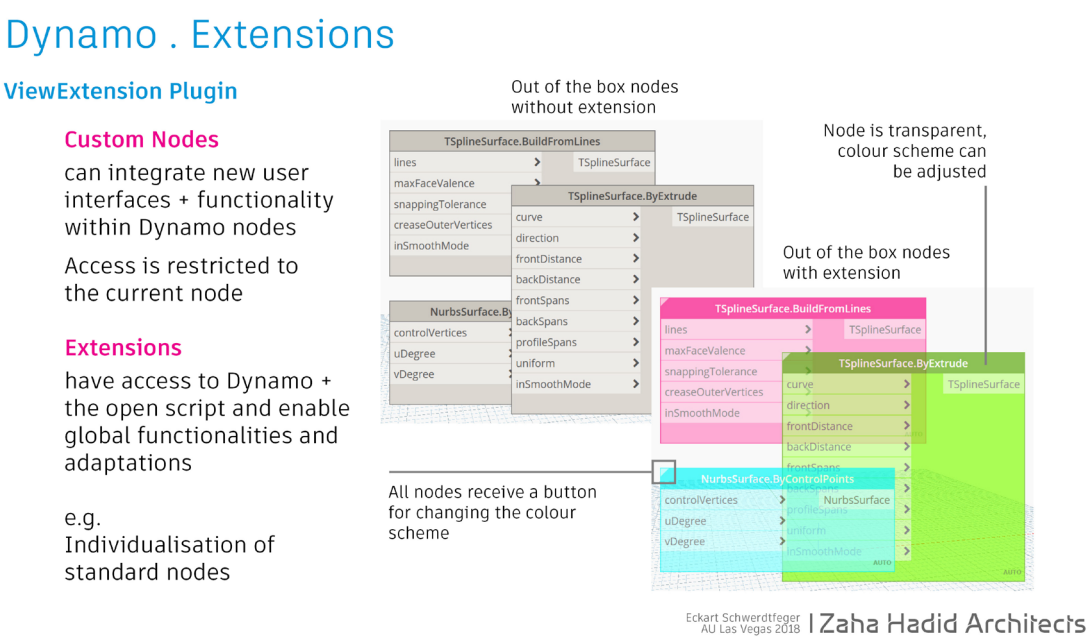
In general, custom nodes can integrate new functionalities and user interfaces within Dynamo. Since they are part of a script and run in a self-contained way, their access is usually limited to their own context and restricted for the rest of the Dynamo graph and the other nodes used.

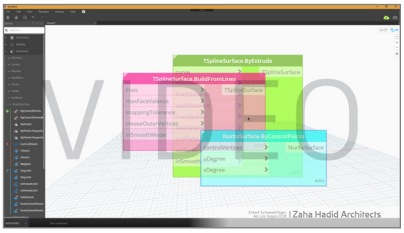
Dynamo Visual Scripting Extensions

Recently, extensions were introduced as another possibility to add custom features to Dynamo and to overcome the limitations of custom node development.

Extensions are plugins that gain access to the entire Dynamo environment, e.g. the Dynamo user interface, display and open visual scripts. They can be used to add to or control Dynamo on a more global level.

Using a ViewExtension plugin the previous research and developments regarding custom coloured nodes can now be applied to all out-of-the-box Dynamo nodes as well.



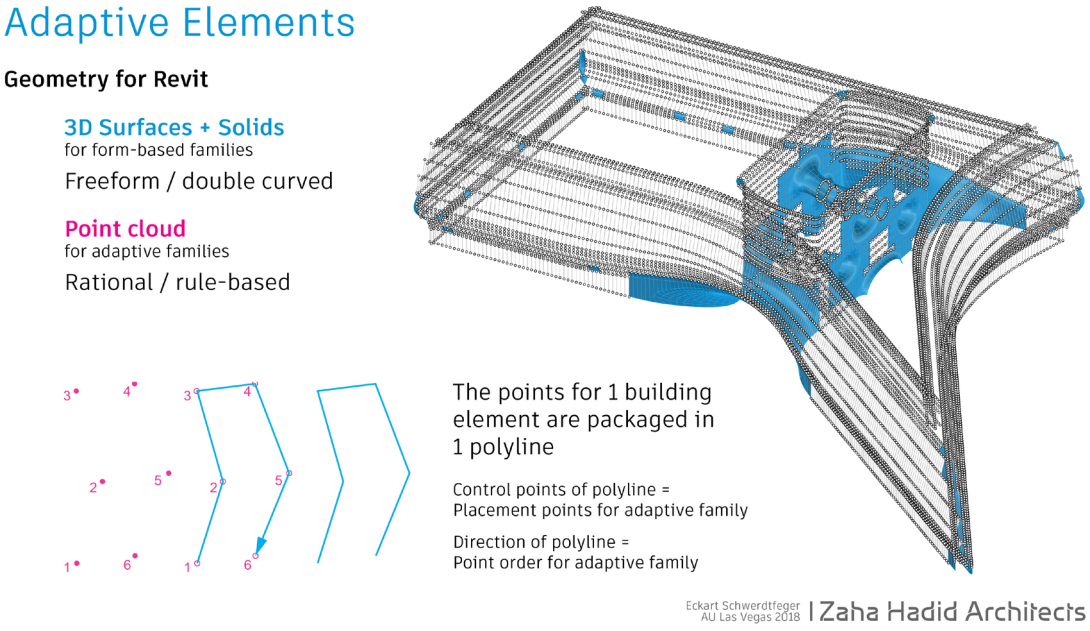


https://vimeo.com/eckart/aulasvegas2018-13

Adaptive Elements

Adaptive elements are parametric families that can be placed and controlled by just points and meta data which makes them very lightweight, easy to transfer and efficient for implementing flat or single curved facade or roof panels within a three-dimensional space.

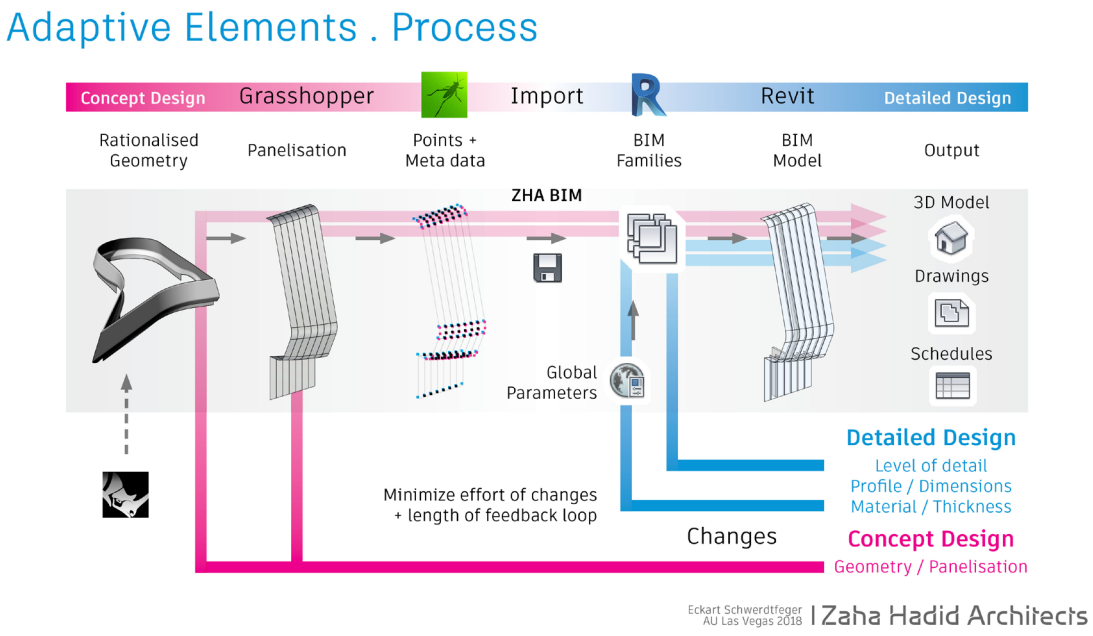
The facade design and panelisation was generated by a Grasshopper script. For the BIM model the panels were divided into form-based and adaptive families based on their geometry type and most suitable Revit workflow.



Adaptive Elements

The aim was an efficient transfer process from Grasshopper to Revit that enables regular design changes and fast adaptations of the BIM model. The ZHA BIM plugin was used to transfer the required point coordinates and meta data from Rhino to Revit and to instantiate the panels.

During concept design phase the base geometry and panelisation changed regularly and the translation process needed to be run completely. Due to our clearly defined and predetermined workflow this was highly efficient and swift.



During detailed design phase updates were limited to different types of changes, e.g. level of detail, profiles, dimensions, etc. and those were implemented inside Revit, e.g. via parametric families, global parameters.

This minimises the effort of changes and the feedback loop length. Rhino and Grasshopper are taken out of the process and the focus fully shifts to the parametric BIM model.

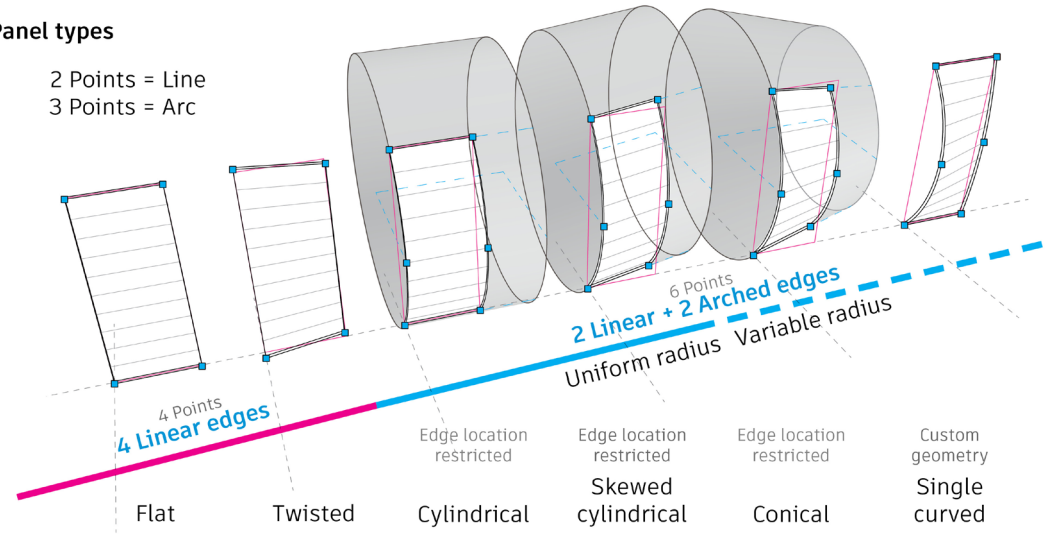
Families

Adaptive Revit families were created to turn the input points into fully parametric facade panels and mullions. The family parameters were linked to global parameters so that changes could be implemented with just changing global dimensions and values. Reporting parameters provide live information about the panel geometry and measurements for scheduling.

Adaptive Elements . Facade Panels

Panel types

2 Points = Line
3 Points = Arc



Eckart Schwerdtfeger AU Las Vegas 2018 | Zaha Hadid Architects

Adaptive Elements . Facade Panels

Adaptive Family

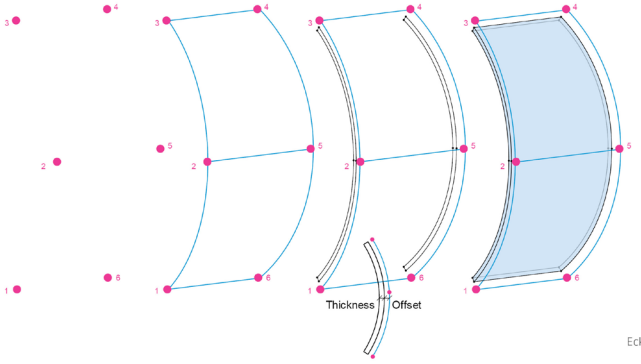
1 Set up adaptive points, order and placement

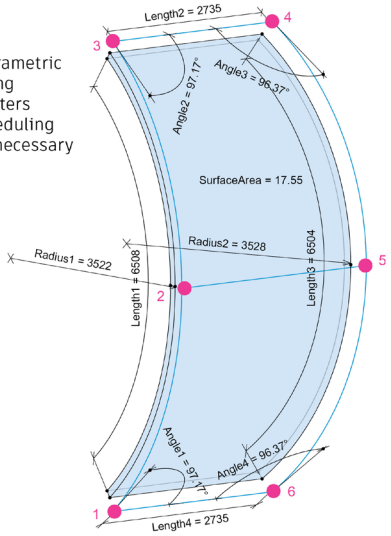
2 Create frame out of lines [2 points] and arcs [3 points] that allows for panel edge offsets

3 Place edge profiles using nested family that deals with panel offset and thickness

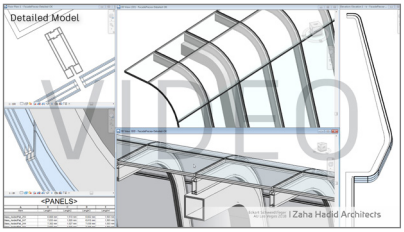
4 Blend both edge profiles into a solid object

5 Add parametric reporting parameters for scheduling where necessary





Eckart Schwerdtfeger AU Las Vegas 2018 | Zaha Hadid Architects



https://vimeo.com/eckart/aulasvegas2018-14

As a result, the entire facade can be controlled with points and a couple parameter values only, is efficient and easy to change as well as very lightweight in regards to the file size.

Adaptive Elements

Due to the angled facades, most panels are cut at the corners. This results in a large number of individually shaped panels that would have required many custom families and intricate changes in the Grasshopper script.

A much more simple and efficient solution was to keep the 2 standardised families [flat + arched panels] everywhere and to use void elements at the corners to subtract the overhanging pieces. After every facade instantiation a C# macro is run and automatically adjust the building corners as required.

Facade . Perimeter

Custom panels

Use standardised families only + cut corners with voids

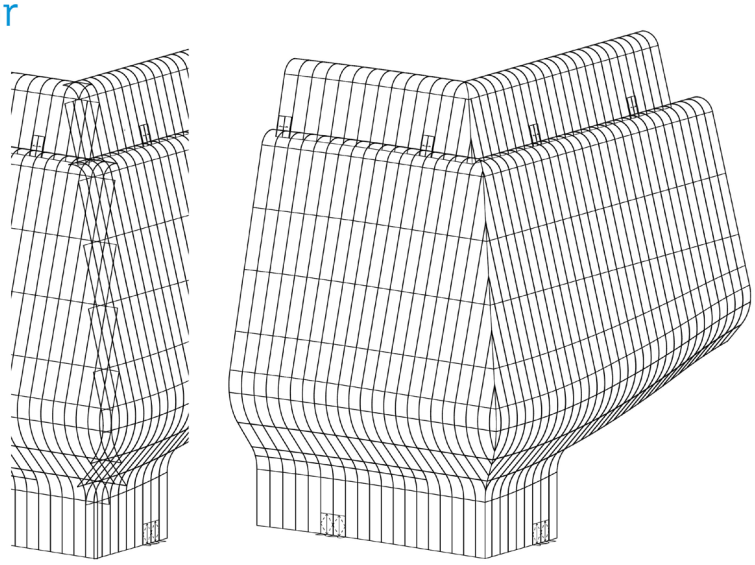
C# Macro

For every panel...

- Read parameter 'Orientation'
- Check intersection of panel with void of same 'Orientation'
- Upon intersection...

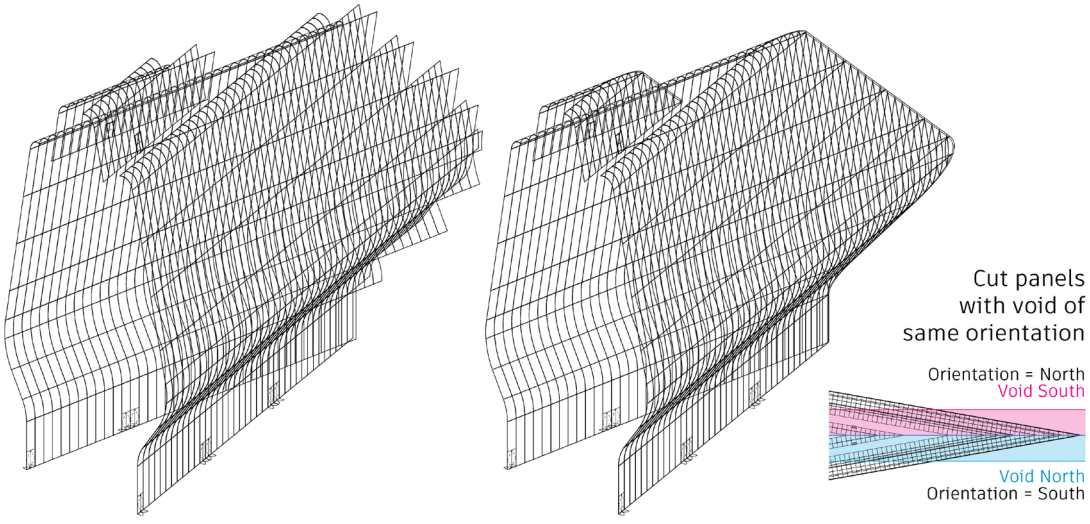
  Subtract void from element

  Update parameters 'Cut' + 'Area'



Eckart Schwerdtfeger AU Las Vegas 2018 | Zaha Hadid Architects

Facade . Perimeter

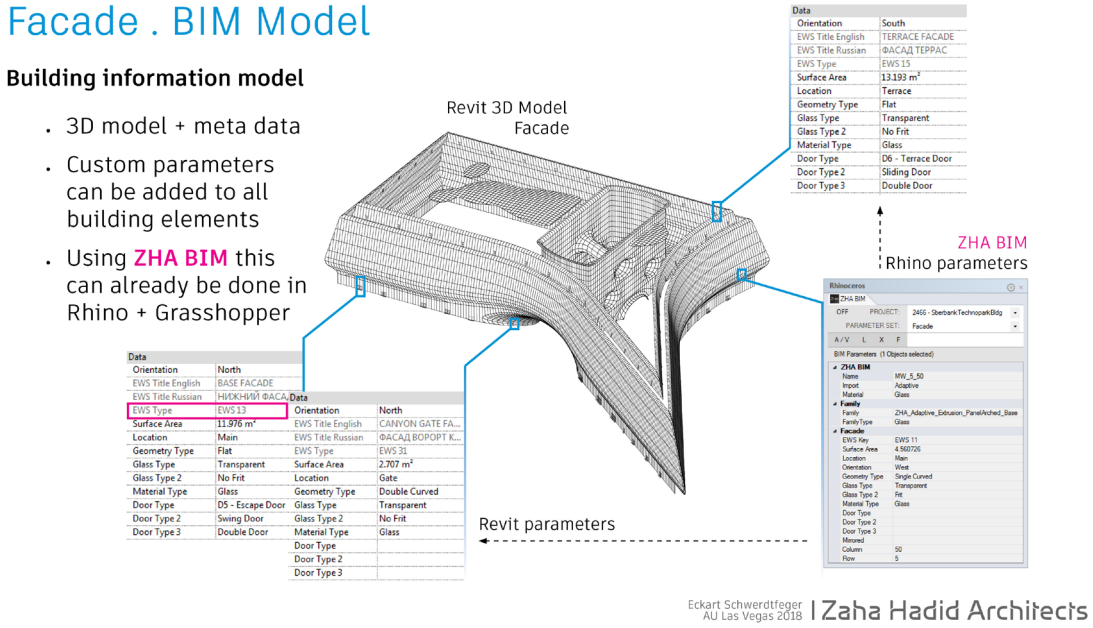


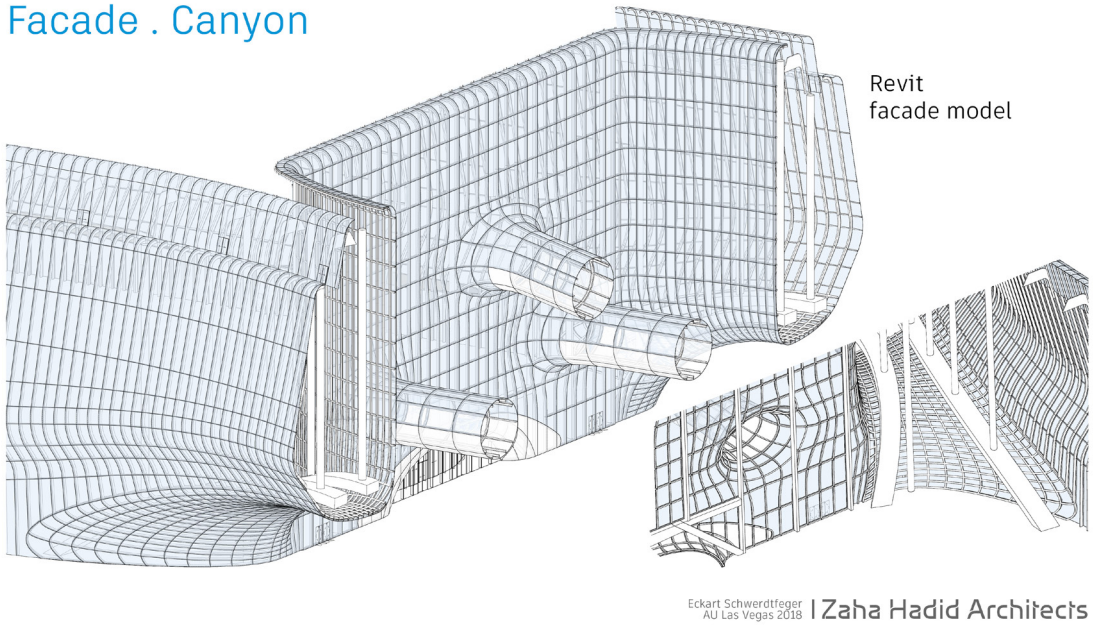
Eckart Schwerdtfeger AU Las Vegas 2018 | Zaha Hadid Architects

Adaptive Elements

After the transfer from Rhino, the facade model contains a mixture of adaptive families and form-based families as well as reporting parameters that are read from the geometry and custom parameters that were set by the initial Grasshopper script and brought into Revit by the ZHA BIM plugin.

The best suited and most efficient workflows are used for each BIM element and in the end they form one coherent facade model with the same appearances and parameters that can be scheduled and analysed altogether.

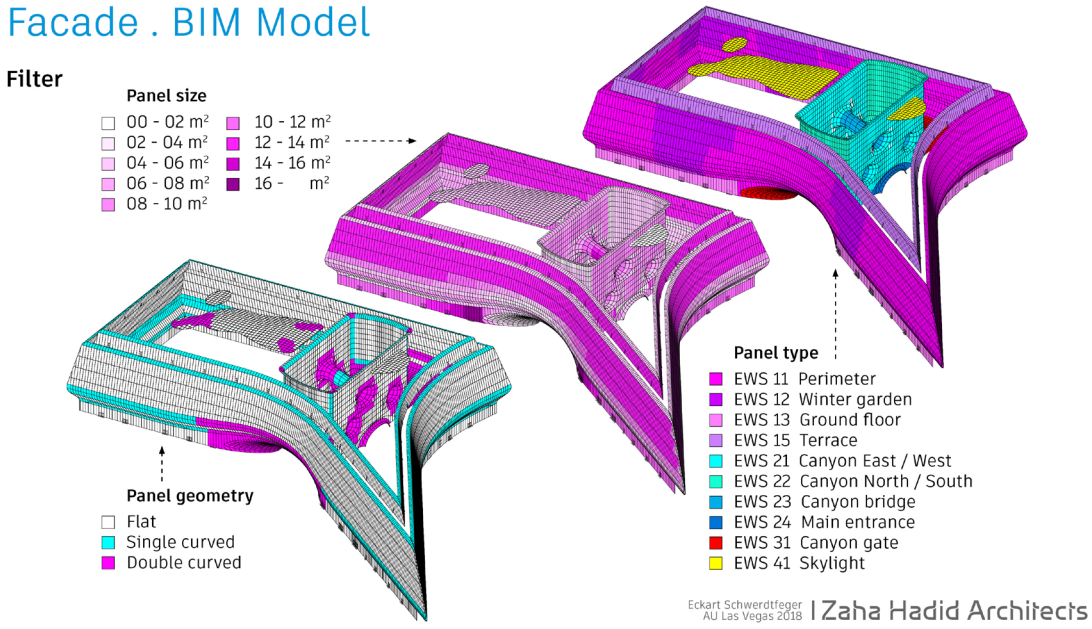


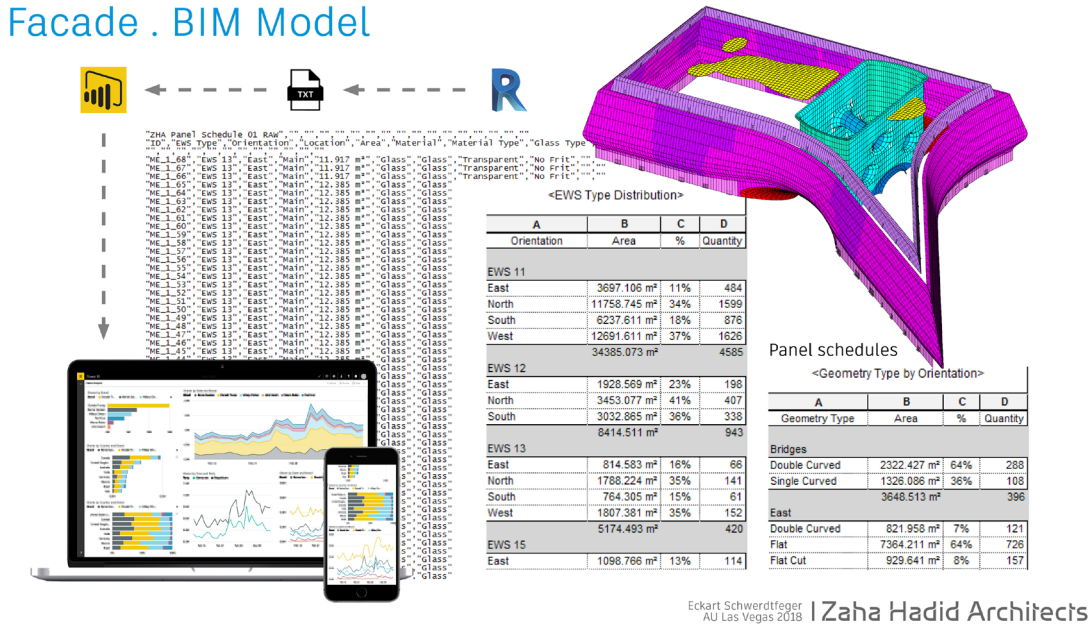


Adaptive Elements

View templates that filter for the various custom parameters are set up to quickly and easily display facade characteristics and to generate isometric drawings for visualisation.

Furthermore, CSV files including all facade parameters are exported for processing in a dashboard software like Power BI to generate a detailled analysis of key characteristics and facade options for evaluation and as overview for project management.





Links

Revit . Plugins + Macros

- https://mva.microsoft.com/en-US/training-courses/c-fundamentals-for-absolute-beginners-16169
- https://www.autodesk.com/developer-network/platform-technologies/revit
- http://www.autodesk.com/myfirstrevitplugin
- https://www.autodesk.com/autodesk-university/class/Pushing-Revit-Next-Level-Intro-Revit-Plugins-C-2018
- https://knowledge.autodesk.com/guidref/RVT/2019/learn-explore/GUID-4DFDA8CD-B0FD-492E-8EDE-A28C29B1E316/REVITPRODUCTS
- http://www.revitapidocs.com/
- https://thebuildingcoder.typepad.com/

Forge . Revit Automation

- https://forge.autodesk.com/
- https://thebuildingcoder.typepad.com/blog/2018/10/forge-design-automation-for-revit.html
- https://thebuildingcoder.typepad.com/blog/2018/11/forge-design-automation-for-revit-at-au-and-in-public.html
- https://www.autodesk.com/autodesk-university/class/Revit-Data-Forge-How-Can-Design-Automation-Revit-API-Help-Me-2018

Dynamo

- http://dynamobim.org/
- http://dynamobim.com/learn/
- http://primer.dynamobim.org/
- https://forum.dynamobim.com/

Dynamo . T-Splines

- http://dynamobim.org/a-high-level-introduction-to-t-splines-in-dynamo/

Dynamo . DynaShape

- https://forum.dynamobim.com/t/dynashape/11666
- https://www.autodesk.com/autodesk-university/class/Designing-Complex-Forms-Dynamo-DynaShape-2018

Dynamo . Custom Nodes

- https://github.com/DynamoDS/Dynamo/wiki/How-To-Create-Your-Own-Nodes
- https://github.com/DynamoDS/DynamoSamples
- https://en.wikipedia.org/wiki/Windows_Presentation_Foundation
- https://docs.microsoft.com/en-gb/visualstudio/designers/getting-started-with-wpf
- https://www.wpftutorial.net

Dynamo . Extensions

- http://dynamobim.org/extensions-workshop-materials-now-available/
- http://dynamobim.org/extensions-now-supported-in-package-manager/

ZAHA HADID ARCHITECTS

Studio London
10 Bowling Green Lane
London EC1R 0BQ
T +44 20 7253 5147
F +44 20 7251 8322

mail@zaha-hadid.com

www.zaha-hadid.com
www.twitter.com/ZHA_News
www.linkedin.com/company/zaha-hadid-architects/

Eckart Schwerdtfeger
BIM Associate

Eckart.Schwerdtfeger@gmx.com
Eckart.Schwerdtfeger@zaha-hadid.com

www.parametrised.net
www.twitter.com/Eckart_S
www.linkedin.com/in/eckart-schwerdtfeger/