

Pushing the Envelope

Stretching the limits of Generative Design

António Menezes Leitão
INESC-ID/Instituto Superior Técnico, Portugal
antonio.menezes.leitao@ist.utl.pt

Rita Fernandes
Instituto Superior Técnico, Portugal
rita.serra.fernandes@gmail.com

Luís Santos
University of California, Berkeley, USA
luis.sds82@berkeley.edu

SIGraDI 2013

Stretching the Limits

*Geometry may still serve as the output of the design, but the design representation shifts, **at least partially**, into the realm of programming.*

Kilian, 2006

Stretching the Limits

*Geometry may still serve as the output of the design, but the design representation shifts, **at least partially**, into the realm of programming.*

Kilian, 2006

*Geometry may still serve as the output of the design, but the design representation shifts, **totally**, into the realm of programming.*

Leitão, 2013

Context

Teaching Computer Science (in just one semester)

- ▶ Emphasis on Mathematics
- ▶ Translated to *expressive* programming languages

Context

Teaching Computer Science (in just one semester)

- ▶ Emphasis on Mathematics
- ▶ Translated to *expressive* programming languages

Volume of frustum of square pyramid - Egyptian Math



Context

Teaching Computer Science (in just one semester)

- ▶ Emphasis on Mathematics
- ▶ Translated to *expressive* programming languages

Volume of frustum of square pyramid - Egyptian Math

If you are told: a truncated pyramid of h for the vertical height by b on the base by t on the top:

1. You are to square the b ; result r_1 .
2. You are to multiply b and t ; result r_2 .
3. You are to square the t ; result r_3 .
4. You are to add the r_1 and the r_2 and the r_3 ; result r_4 .
5. You are to take $1/3$ of h ; result r_5 .
6. You are to take r_4 times r_5 ; result r_6 .

Context

Teaching Computer Science (in just one semester)

- ▶ Emphasis on Mathematics
- ▶ Translated to *expressive* programming languages

Volume of frustum of square pyramid - Modern Math

$$\text{Volume}(h, b, t) = \frac{h}{3}(b^2 + bt + t^2)$$

Context

Teaching Computer Science (in just one semester)

- ▶ Emphasis on Mathematics
- ▶ Translated to *expressive* programming languages

Volume of frustum of square pyramid - Python

```
def volume (h,b,t):  
    return h/3.0*(pow(b,2)+b*t+pow(t,2))
```

Context

Teaching Computer Science (in just one semester)

- ▶ Emphasis on Mathematics
- ▶ Translated to *expressive* programming languages

Volume of frustum of square pyramid - JavaScript

```
function volume (h,b,t)
  return h/3.0*(Math.pow(b,2)+b*t+Math.pow(t,2));
```

Context

Teaching Computer Science (in just one semester)

- ▶ Emphasis on Mathematics
- ▶ Translated to *expressive* programming languages

Volume of frustum of square pyramid - Lisp

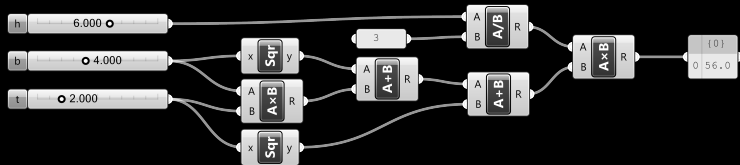
```
(defun volume (h b t)
  (* (/ h 3) (+ (pow b 2) (* b t) (pow t 2))))
```


Context

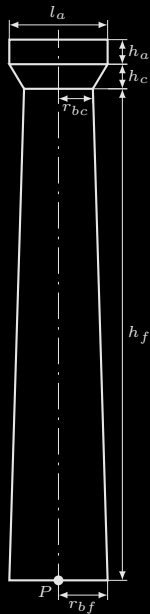
Teaching Computer Science (in just one semester)

- ▶ Emphasis on Mathematics
- ▶ Translated to *expressive* programming languages

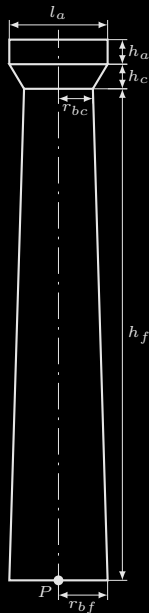
Volume of frustum of square pyramid - Grasshopper



Primitives & Abstractions

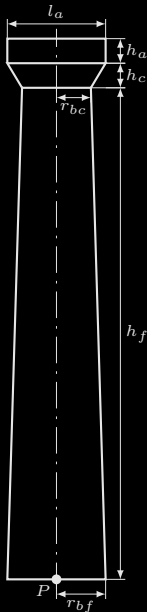


Primitives & Abstractions



$$\begin{aligned} \text{column}(p, h_f, r_{bf}, h_c, r_{bc}, h_a, l_a) = & \\ & \text{frustum}(p, h_f, r_{bf}, r_{bc}) \cup \\ & \text{frustum}(+_z(p, h_f), h_c, r_{bc}, l_a/2) \cup \\ & \text{box}(+_z(p, h_f + h_c), h_a, l_a) \end{aligned}$$

Primitives & Abstractions

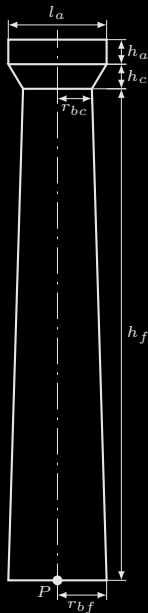


$$\begin{aligned} \text{column}(p, h_f, r_{bf}, h_c, r_{bc}, h_a, l_a) = & \\ & \text{frustum}(p, h_f, r_{bf}, r_{bc}) \cup \\ & \text{frustum}(+_z(p, h_f), h_c, r_{bc}, l_a/2) \cup \\ & \text{box}(+_z(p, h_f + h_c), h_a, l_a) \end{aligned}$$

The diameter of the columns will be two modules, the height including the capital 14, the height of the capital is one module, the width two modules and a sixth. The height of the capital is to be divided into three parts.

Vitruvius

Primitives & Abstractions

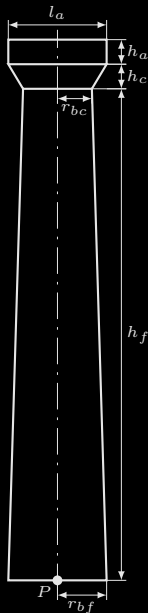


$$\begin{aligned} \text{column}(p, h_f, r_{bf}, h_c, r_{bc}, h_a, l_a) = & \\ & \text{frustum}(p, h_f, r_{bf}, r_{bc}) \cup \\ & \text{frustum}(+_z(p, h_f), h_c, r_{bc}, l_a/2) \cup \\ & \text{box}(+_z(p, h_f + h_c), h_a, l_a) \end{aligned}$$

$$h_f = 13r_{bf} \quad h_c = \frac{2}{3}r_{bf}$$

$$h_a = \frac{1}{3}r_{bf} \quad l_a = \frac{13}{6}r_{bf}$$

Primitives & Abstractions



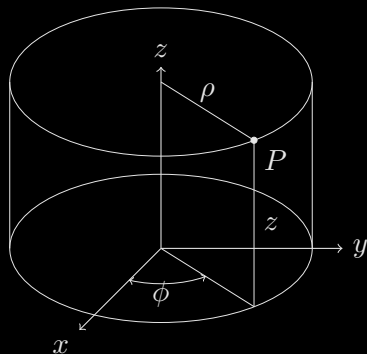
$$\begin{aligned} \text{column}(p, h_f, r_{bf}, h_c, r_{bc}, h_a, l_a) = & \\ \text{frustum}(p, h_f, r_{bf}, r_{bc}) \cup & \\ \text{frustum}(+_z(p, h_f), h_c, r_{bc}, l_a/2) \cup & \\ \text{box}(+_z(p, h_f + h_c), h_a, l_a) & \end{aligned}$$

$$h_f = 13r_{bf} \quad h_c = \frac{2}{3}r_{bf}$$

$$h_a = \frac{1}{3}r_{bf} \quad l_a = \frac{13}{6}r_{bf}$$

$$\begin{aligned} \text{doricColumn}(p, r_{bf}, r_{bc}) = & \\ \text{column}(p, 13r_{bf}, r_{bf}, \frac{2}{3}r_{bf}, r_{bc}, \frac{1}{3}r_{bf}, \frac{13}{6}r_{bf}) & \end{aligned}$$

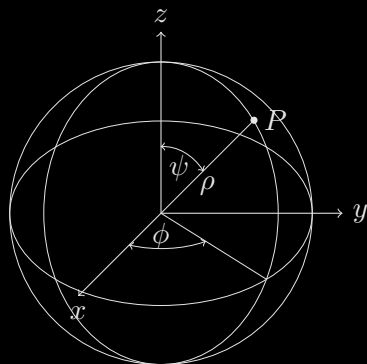
Coordinate Systems



$$\forall_s \in [0, \dots, n]$$

$$\text{step}(\text{cyl}(r_i, \Delta_\omega \cdot s, \Delta_h \cdot s), \\ \text{cyl}(r_e, \Delta_\omega \cdot s, \Delta_h \cdot s))$$

Coordinate Systems



$$\forall \psi \in \{0, \frac{\pi}{10}, \dots, \pi\}$$

$$\forall \phi \in \{\frac{\pi}{10}, 2\frac{\pi}{10}, \dots, 2\pi\}$$

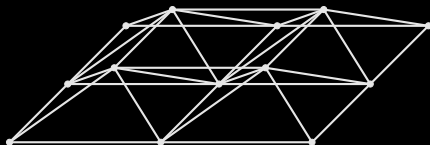
$$\text{cone}(\text{sph}(1, \phi, \psi), \frac{\sin \psi}{10}, \text{xyz}(0, 0, 0))$$

Randomness

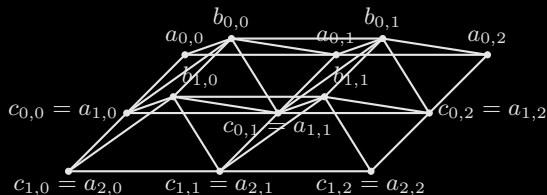
Gaussian Distribution

$$f(x, y) = e^{-\left(\left(\frac{x-x_0}{\sigma_x}\right)^2 + \left(\frac{y-y_0}{\sigma_y}\right)^2\right)}$$

Data Structures



Data Structures



$$\begin{pmatrix} a_{0,0} & a_{0,1} & a_{0,2} & \dots & a_{0,n-1} & a_{0,n} \\ b_{0,0} & b_{0,1} & b_{0,2} & \dots & b_{0,n-1} \\ a_{1,0} & a_{1,1} & a_{1,2} & \dots & a_{1,n-1} & a_{1,n} \\ \dots \\ c_{m,0} & c_{m,1} & c_{m,2} & \dots & c_{m,n-1} & c_{m,n} \end{pmatrix}$$

Data Structures

Solid Geometry

Operations

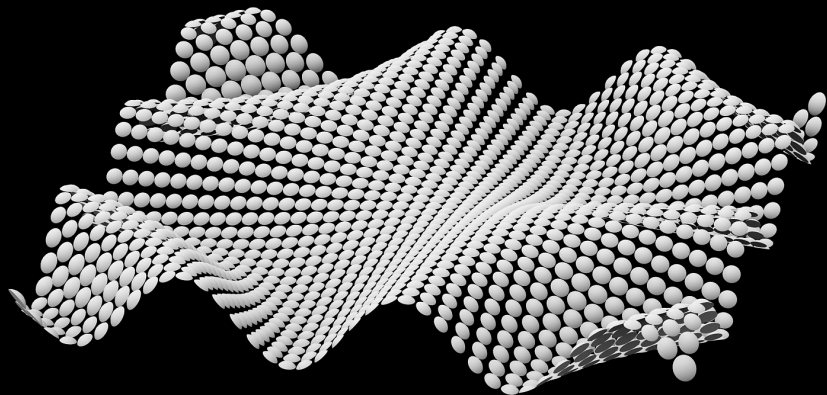
- ▶ Regular polygon
- ▶ Region
- ▶ Sweep
- ▶ Rotation
- ▶ Scale
- ▶ Union
- ▶ Intersection

Parametric Descriptions

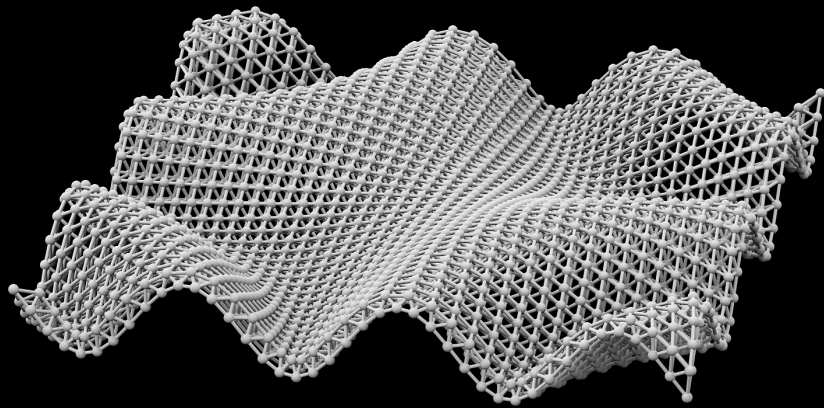
$$\begin{cases} \rho(u, v) = 1 + v \cos \frac{u}{2} \\ \theta(u, v) = u \\ z(u, v) = v \sin \frac{u}{2} \end{cases}$$

Parametric Composition

Tesselations



Tesselations



Generative Design

How far can we go?

- ▶ Case study
- ▶ Formalize an actual design using GD
- ▶ Simulate changes
- ▶ Evaluate their impact in time and effort
- ▶ Compare with conventional approach

Compare the effort

- ▶ Complex models
- ▶ Handling changes

Case Study



©MVRDV

Case Study

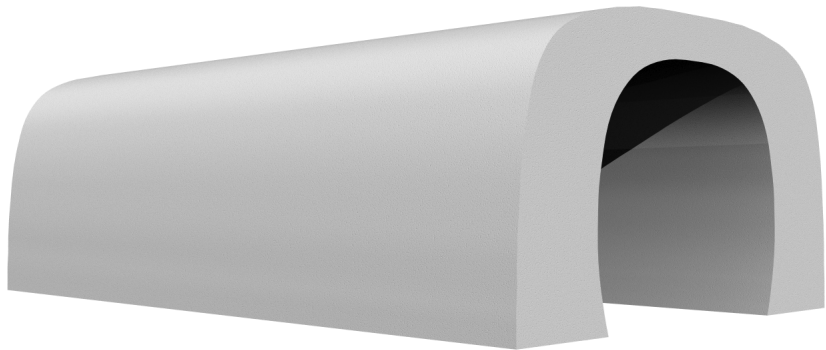
Modeling Process

1. Overall shape
2. Building elements (walls, slabs, etc.)
3. Openings and frames
4. Detailed elements (posts, glass facade, etc.)

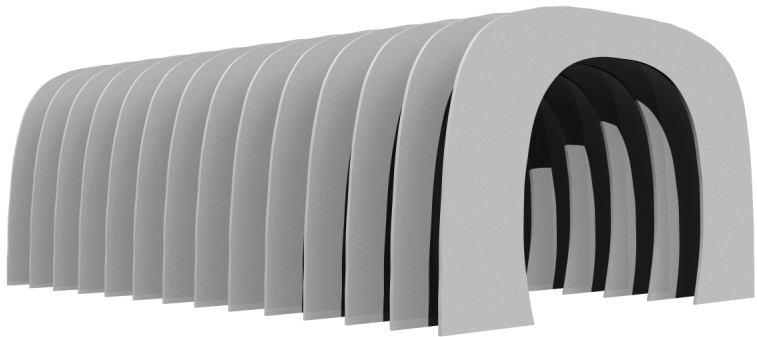
Consequences

- ▶ The output of each phase is the input of the next one
- ▶ Each phase increases the detail of the model
- ▶ Each phase creates dependencies between elements
- ▶ Changes in previous phases are propagated to the next phases

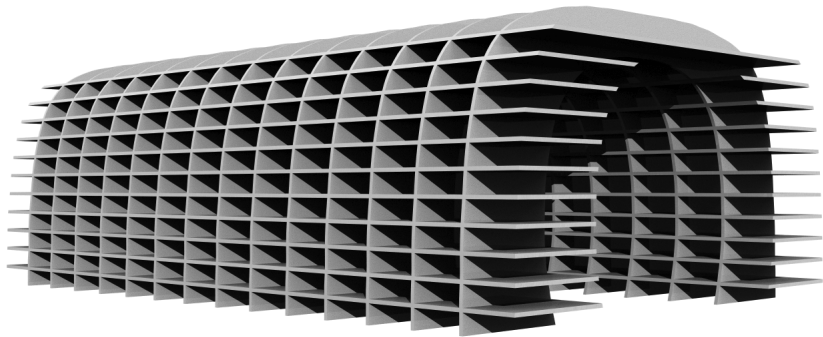
Complex Model Production



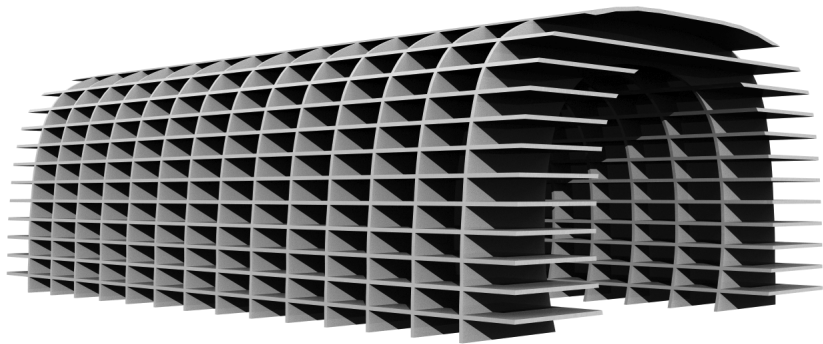
Complex Model Production



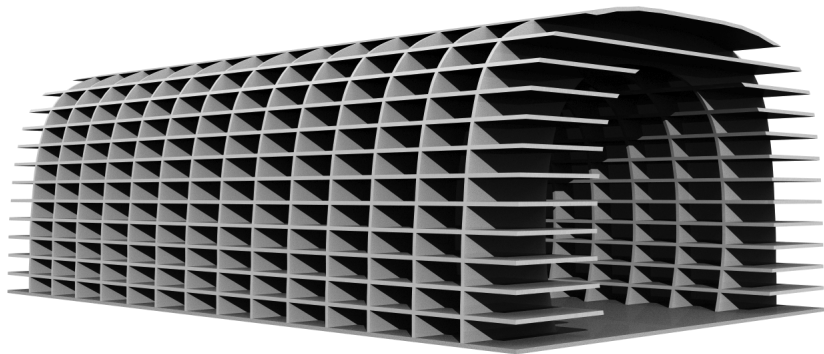
Complex Model Production



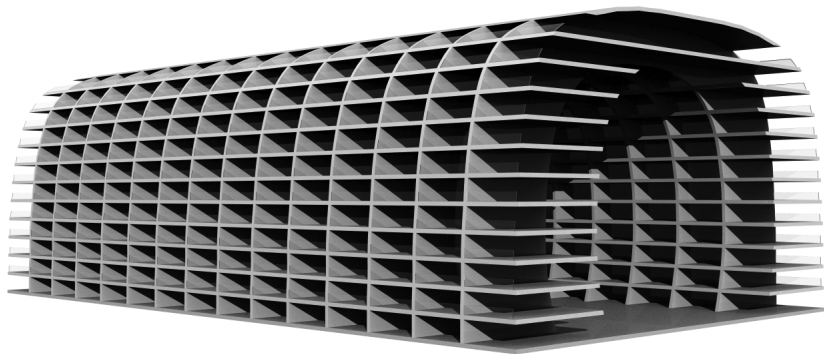
Complex Model Production



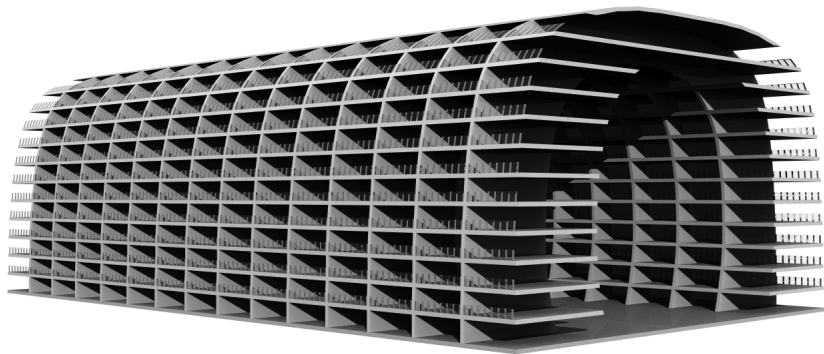
Complex Model Production



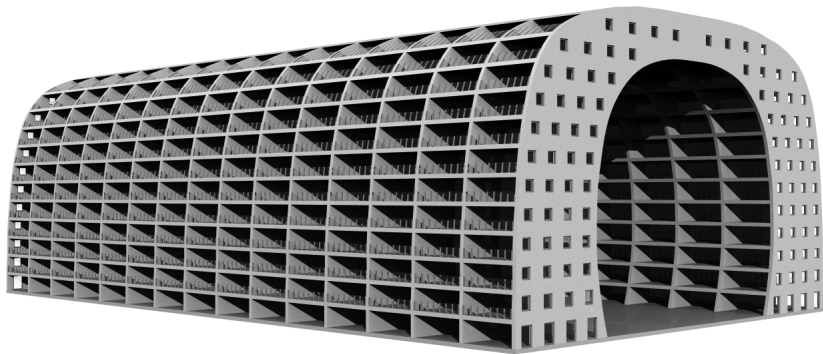
Complex Model Production



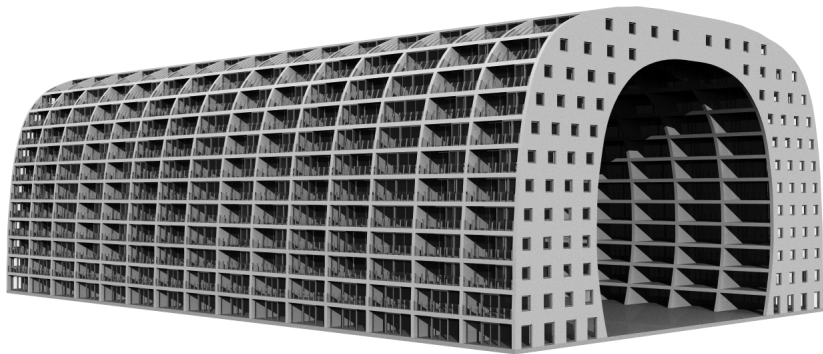
Complex Model Production



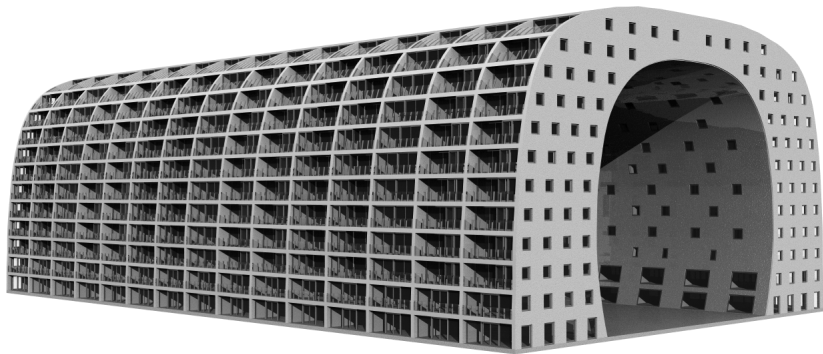
Complex Model Production



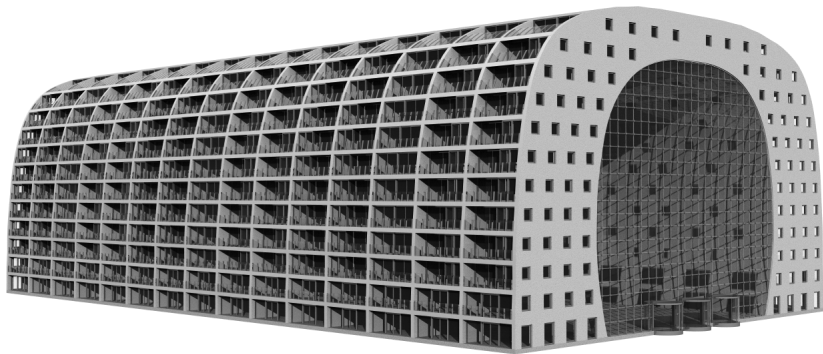
Complex Model Production



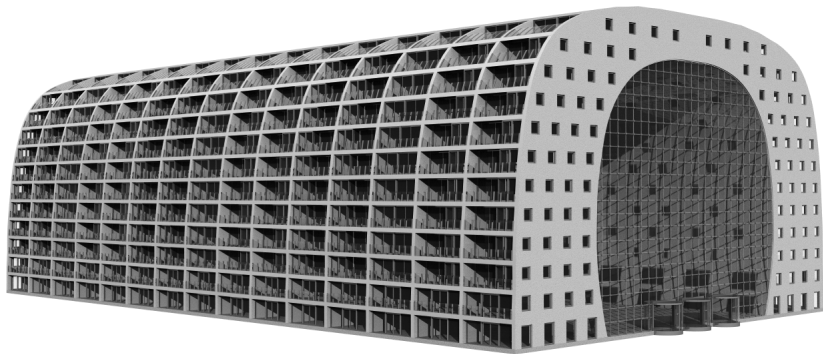
Complex Model Production



Complex Model Production



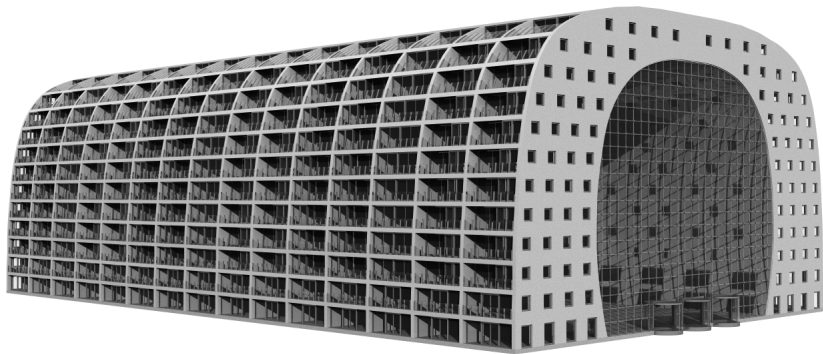
Complex Model Production



Generative

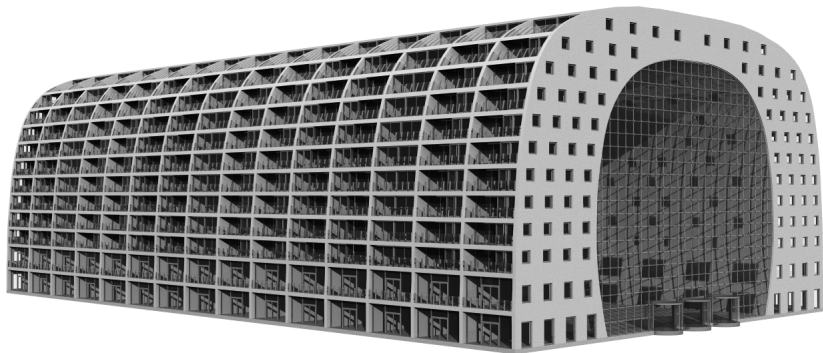
Conventional

Handling Changes



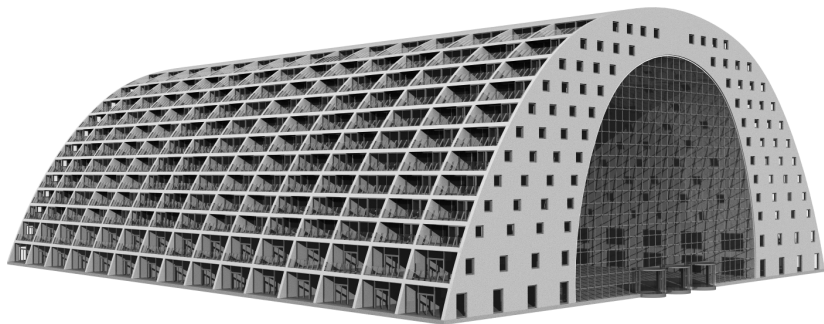
Generative 
Conventional 

Handling Changes--Elements



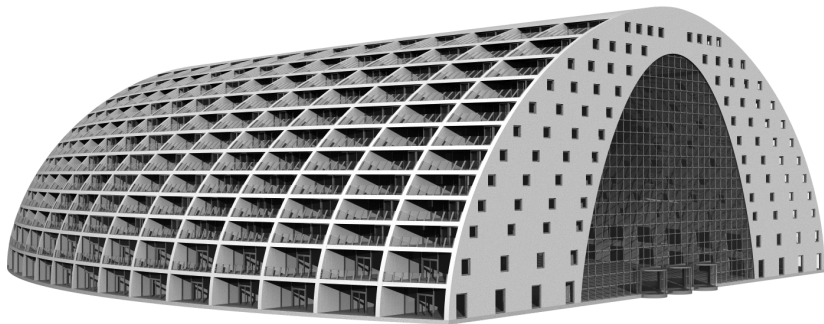
Generative 
Conventional 

Handling Changes--Overall Shape



Generative 
Conventional 

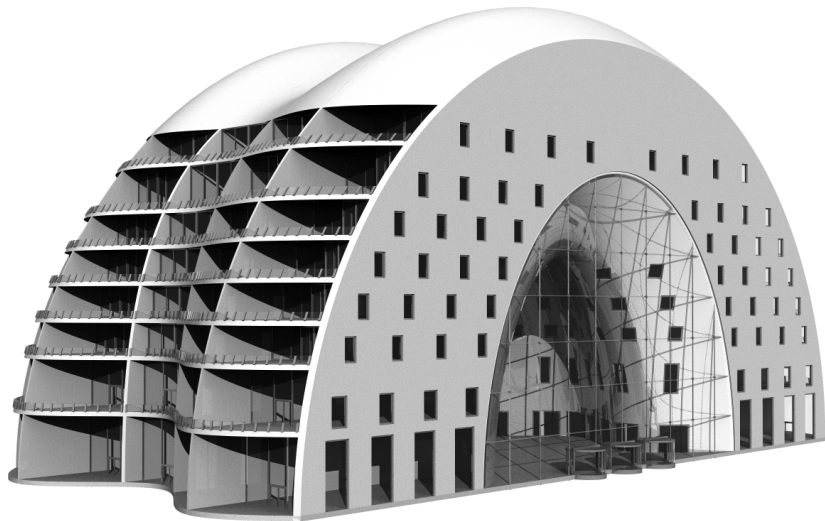
Handling Changes--Overall Shape



Generative

Conventional

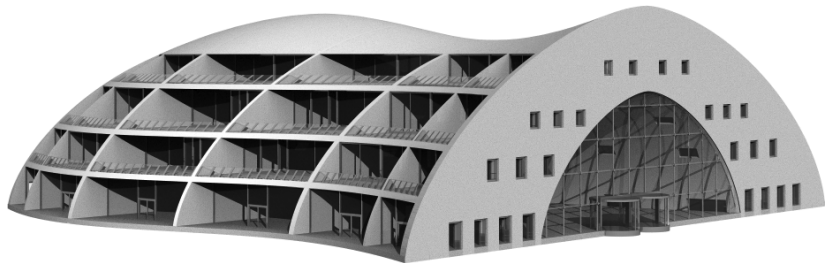
Handling Changes--Shape, Size, Position



Generative
Conventional



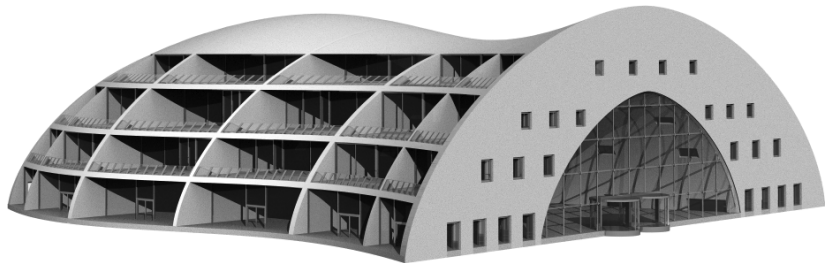
Handling Changes--Shape, Size, Position



Generative
Conventional



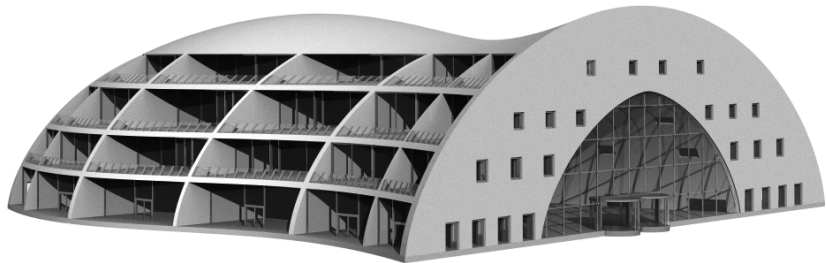
Handling Changes--Shape, Size, Position



Generative 
Conventional 



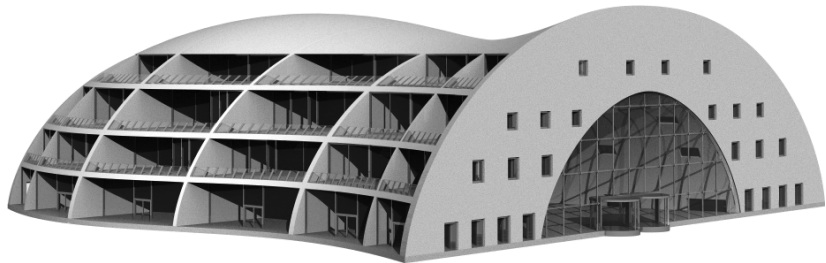
Handling Changes--Shape, Size, Position



Generative 
Conventional 



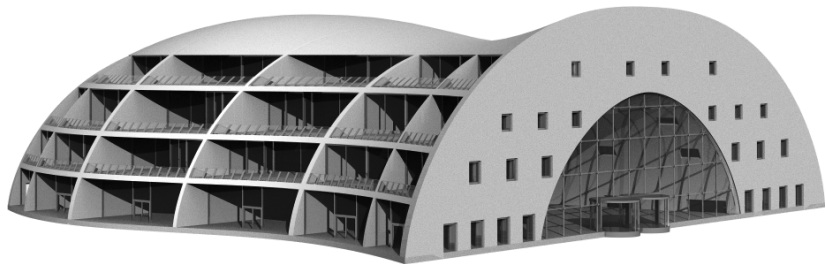
Handling Changes--Shape, Size, Position



Generative 
Conventional 



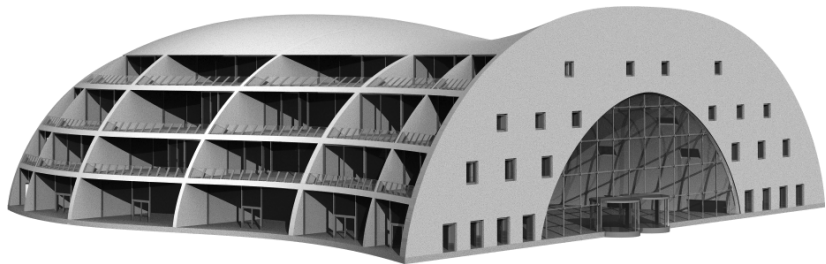
Handling Changes--Shape, Size, Position



Generative 
Conventional 



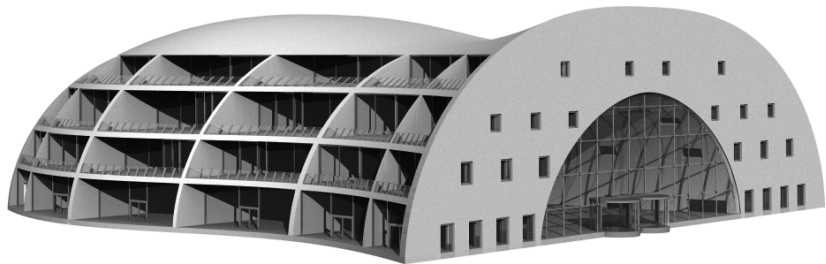
Handling Changes--Shape, Size, Position



Generative 
Conventional 

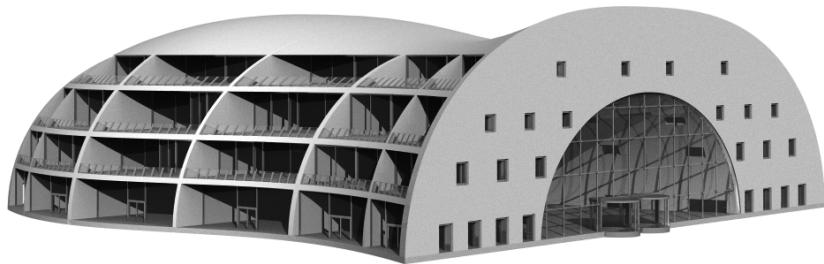


Handling Changes--Shape, Size, Position



Generative 
Conventional 

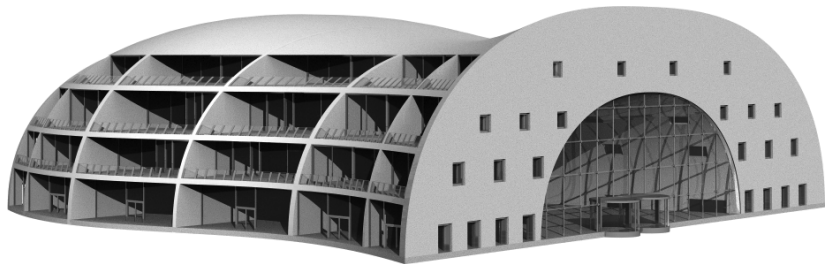
Handling Changes--Shape, Size, Position



Generative 
Conventional 



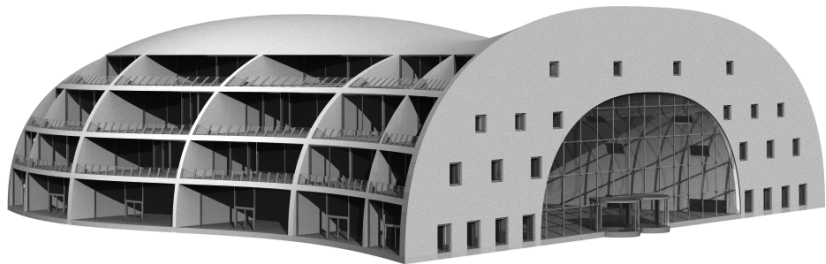
Handling Changes--Shape, Size, Position



Generative 
Conventional 

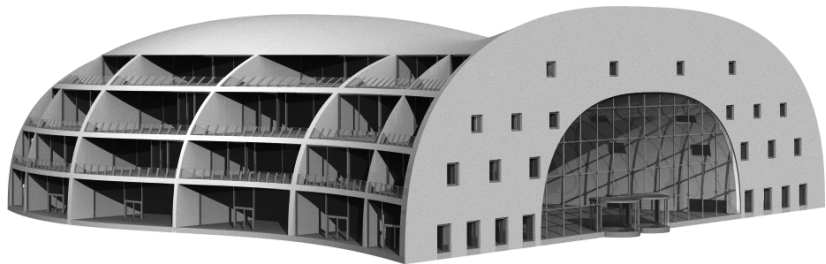


Handling Changes--Shape, Size, Position



Generative 
Conventional 

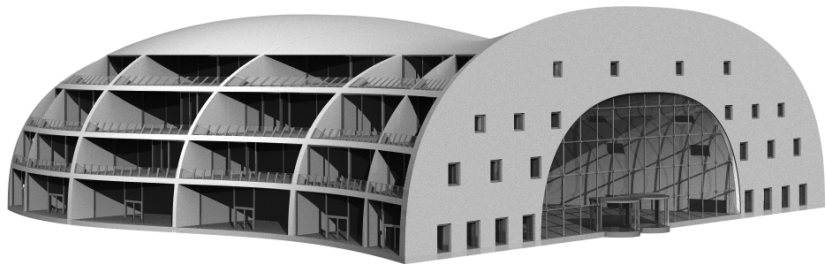
Handling Changes--Shape, Size, Position



Generative
Conventional



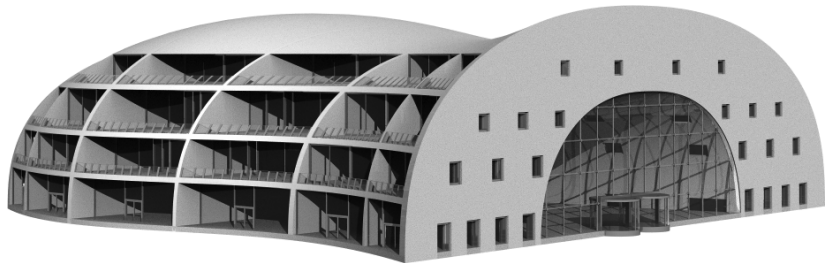
Handling Changes--Shape, Size, Position



Generative 
Conventional 

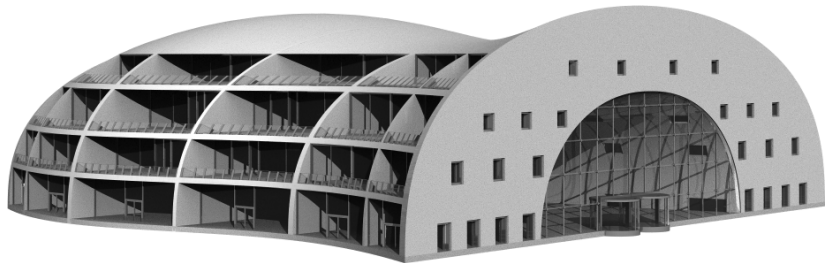


Handling Changes--Shape, Size, Position



Generative 
Conventional 

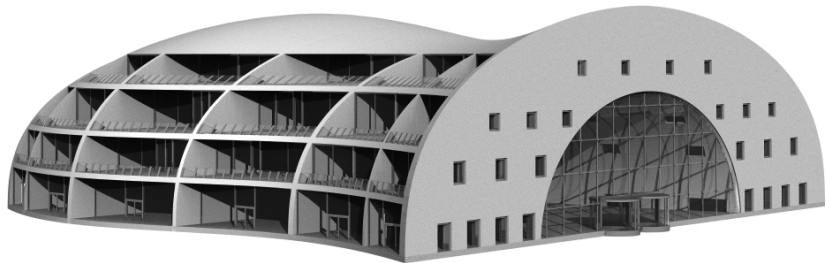
Handling Changes--Shape, Size, Position



Generative 
Conventional 



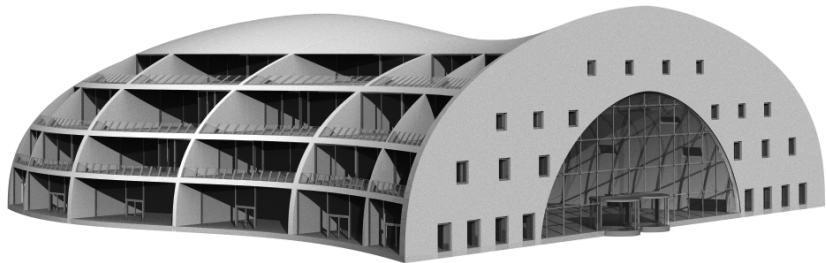
Handling Changes--Shape, Size, Position



Generative 
Conventional 



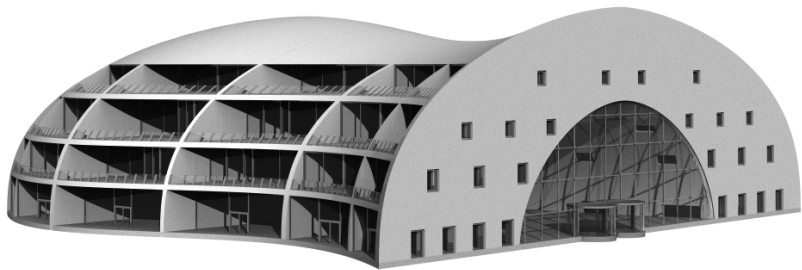
Handling Changes--Shape, Size, Position



Generative 
Conventional 



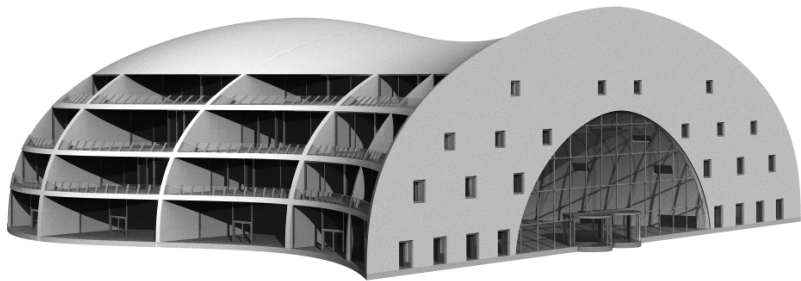
Handling Changes--Shape, Size, Position



Generative 
Conventional 



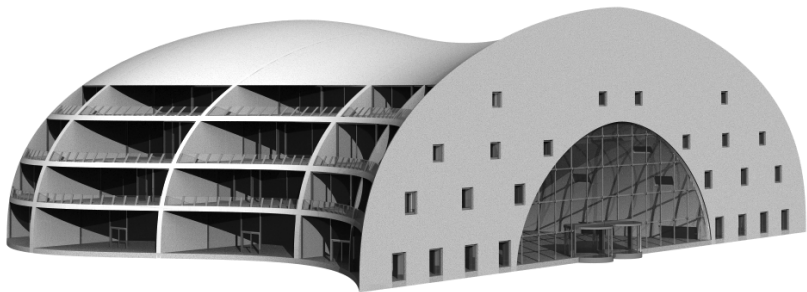
Handling Changes--Shape, Size, Position



Generative
Conventional

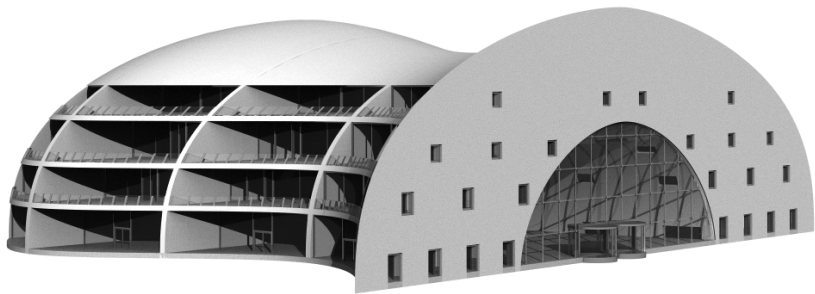


Handling Changes--Shape, Size, Position



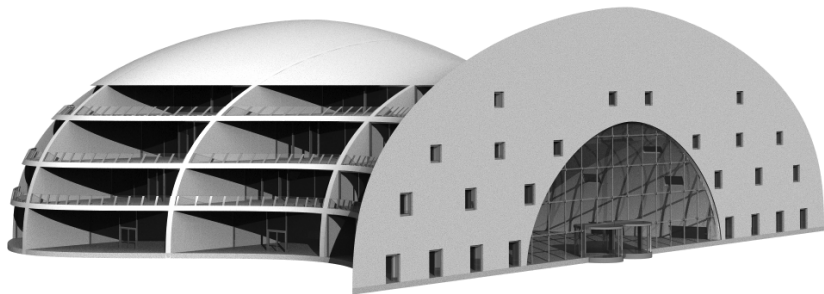
Generative 
Conventional 

Handling Changes--Shape, Size, Position



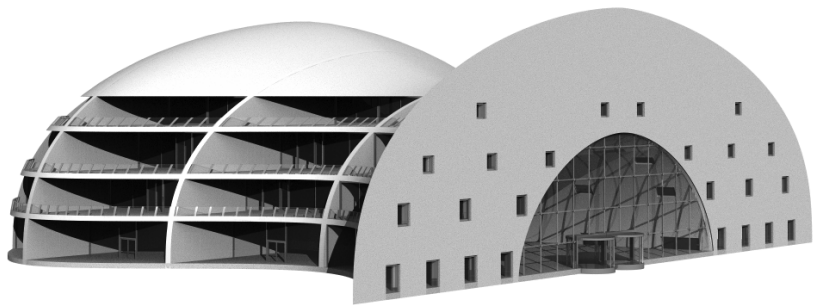
Generative 
Conventional 

Handling Changes--Shape, Size, Position



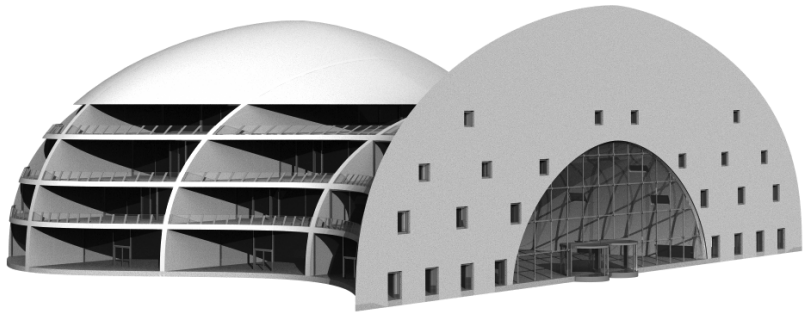
Generative 
Conventional 

Handling Changes--Shape, Size, Position



Generative 
Conventional 

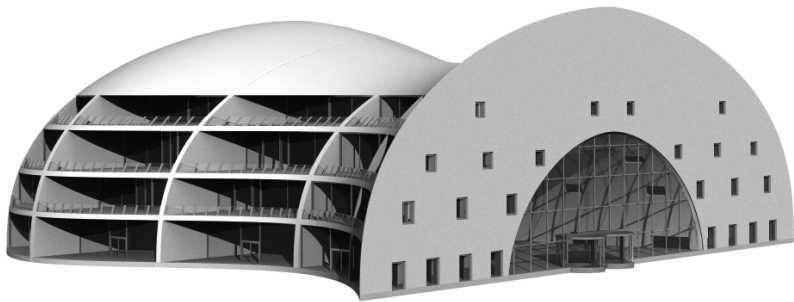
Handling Changes--Shape, Size, Position



Generative
Conventional



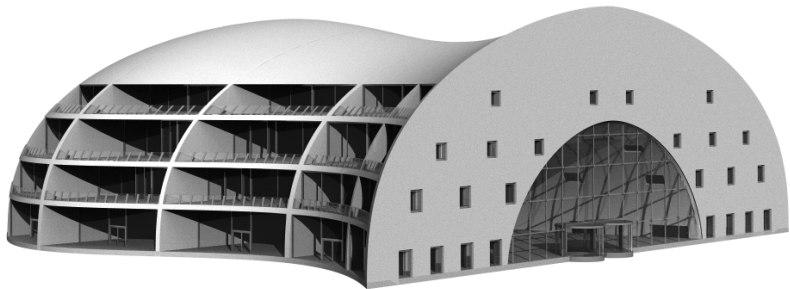
Handling Changes--Shape, Size, Position



Generative 
Conventional 



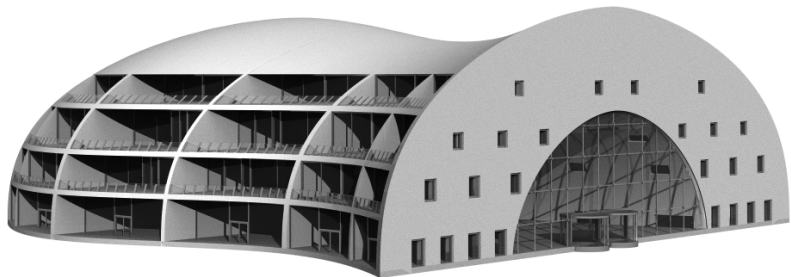
Handling Changes--Shape, Size, Position



Generative
Conventional



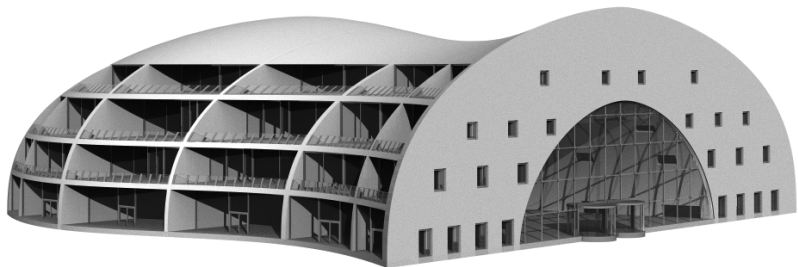
Handling Changes--Shape, Size, Position



Generative 
Conventional 



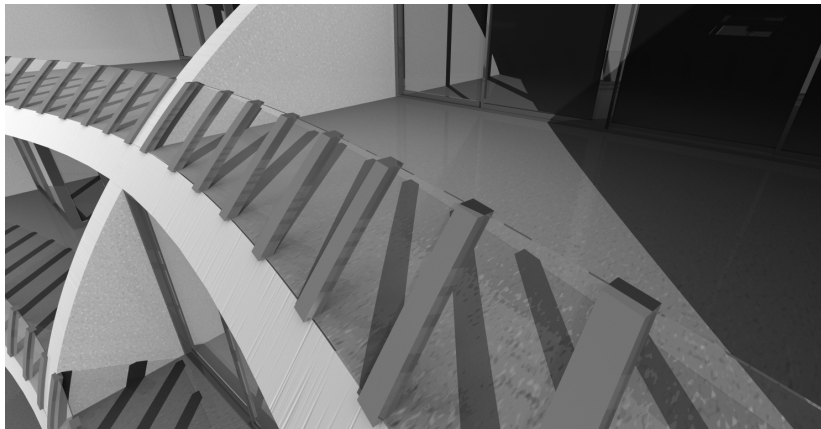
Handling Changes--Shape, Size, Position



Generative 
Conventional 



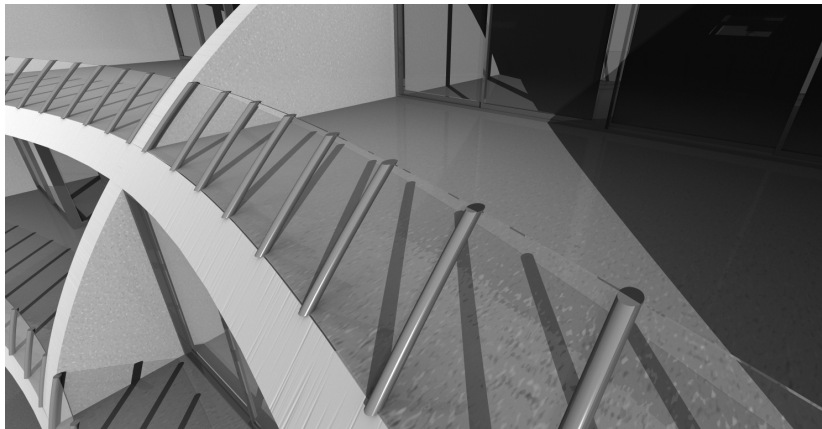
Handling Changes--Elements: Posts



Generative

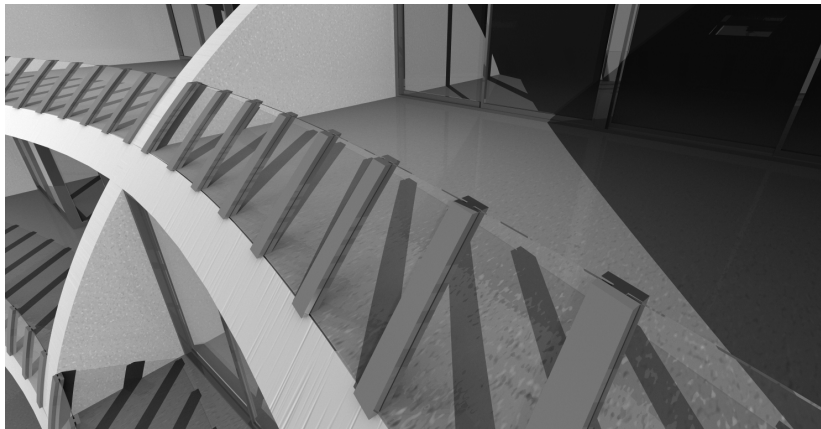
Conventional

Handling Changes--Elements: Posts



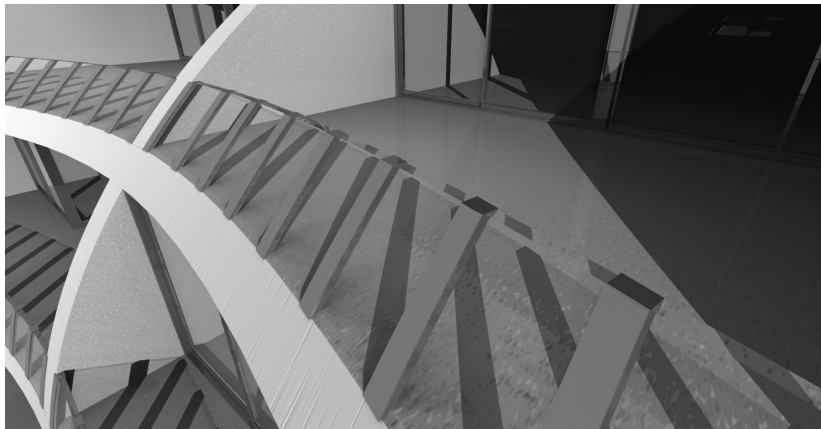
Generative
Conventional

Handling Changes--Elements: Posts



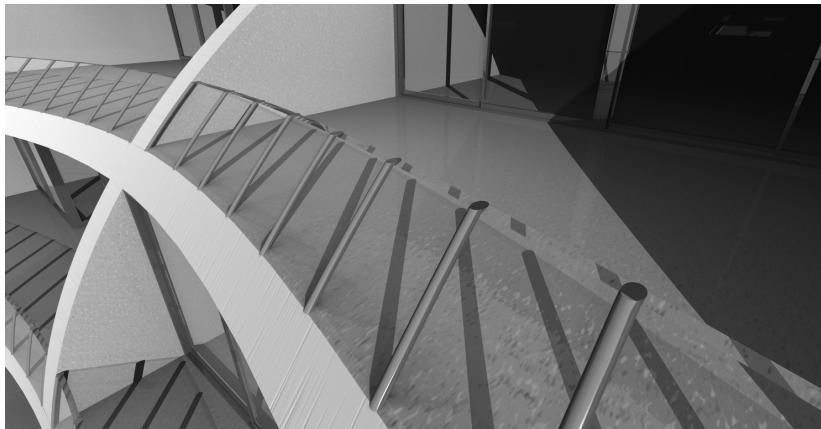
Generative 
Conventional 

Handling Changes--Elements: Posts



Generative 
Conventional 

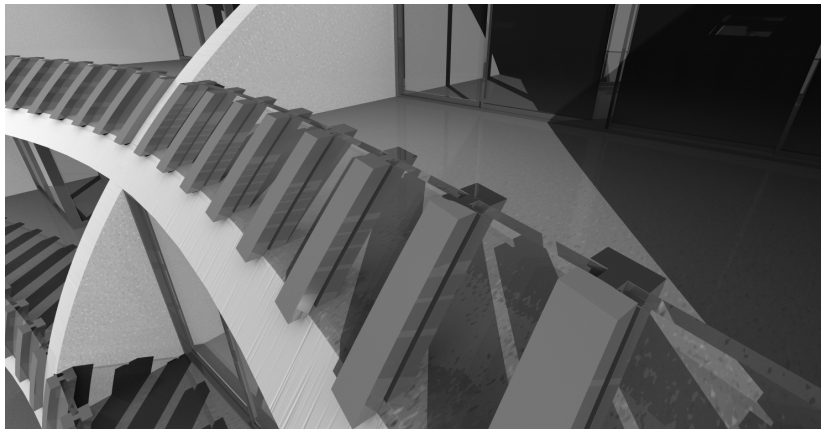
Handling Changes--Elements: Posts



Generative

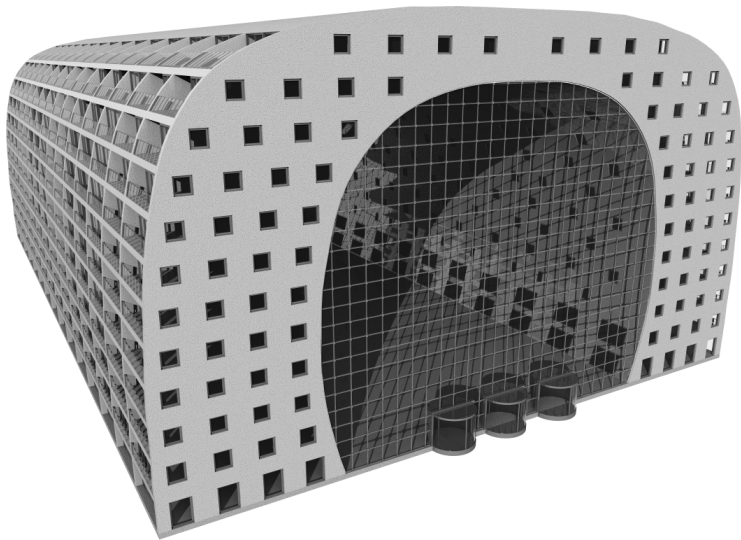
Conventional

Handling Changes--Elements: Posts



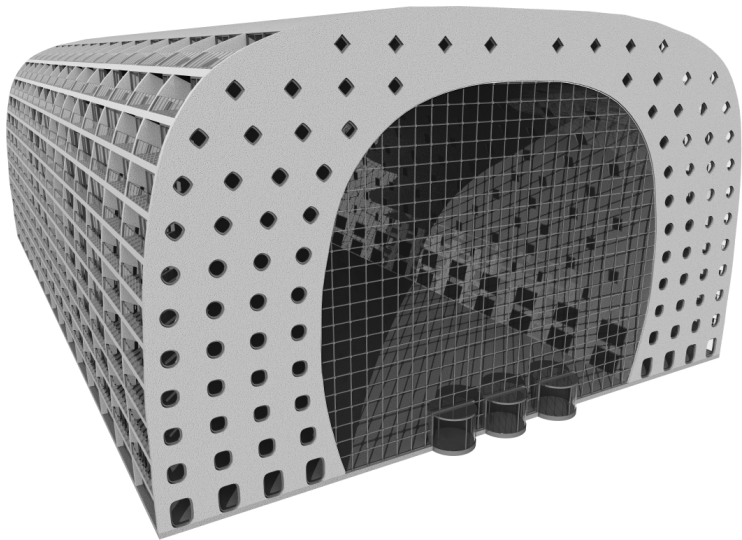
Generative 
Conventional 

Handling Changes--Elements: Windows



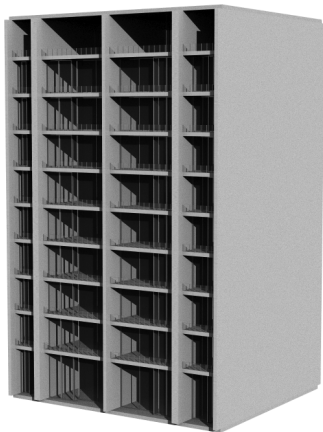
Generative
Conventional

Handling Changes--Elements: Windows



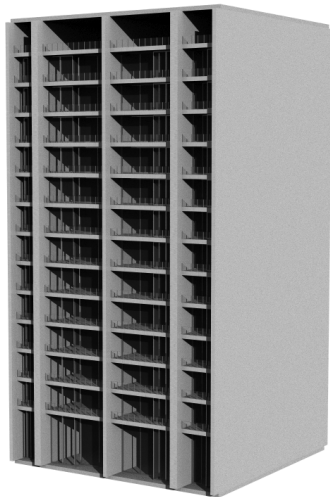
Generative
Conventional

Reuse



Generative 
Conventional 

Reuse



Generative

Conventional



Conclusions

How far can we go using Generative Design?

Conclusions

How far can we go using Generative Design?

- ▶ We don't know yet

Conclusions

How far can we go using Generative Design?

- ▶ We don't know yet
- ▶ But very far, indeed!

Conclusion

It is not the strongest of the species that survives, nor the most intelligent that survives. It is the one that is the most adaptable to change.

Charles Darwin

Conclusion

It is not the strongest of the species that survives, nor the most intelligent that survives. It is the one that is the most adaptable to change.

Charles Darwin

Thank You

Conclusion

It is not the strongest of the species that survives, nor the most intelligent that survives. It is the one that is the most adaptable to change.

Charles Darwin

Thank You

Questions?