

Parametrisch Ontwerp & data uitwisseling

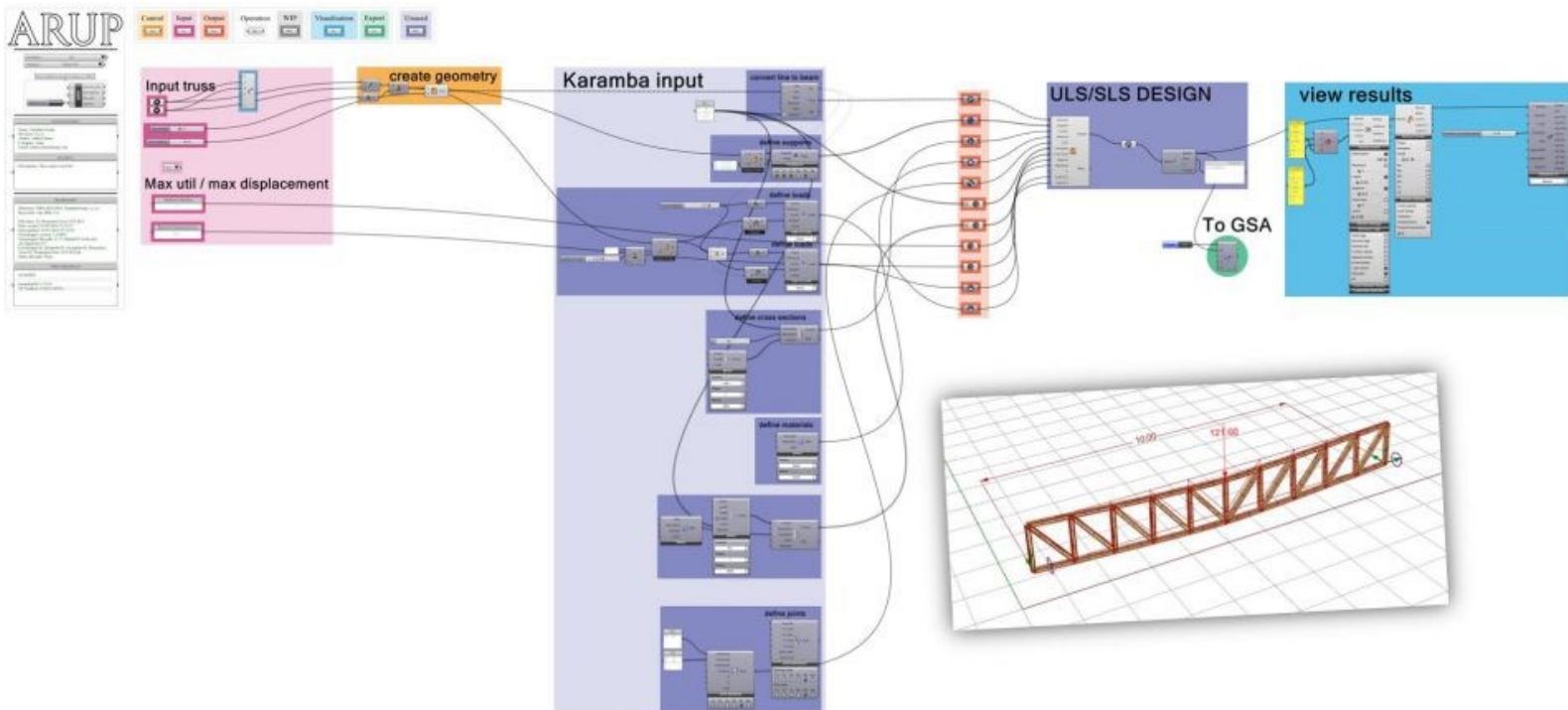
Onderwerpen

- 1) Inleiding parametrisch ontwerp
- 2) Inleiding visueel programmeren
- 3) Inleiding Speckle 3D
- 4) Python & BuildingPy
- 5) XFEM4U
- 6) Voorbeelden parametrische constructies
- 7) Toekomst, AI

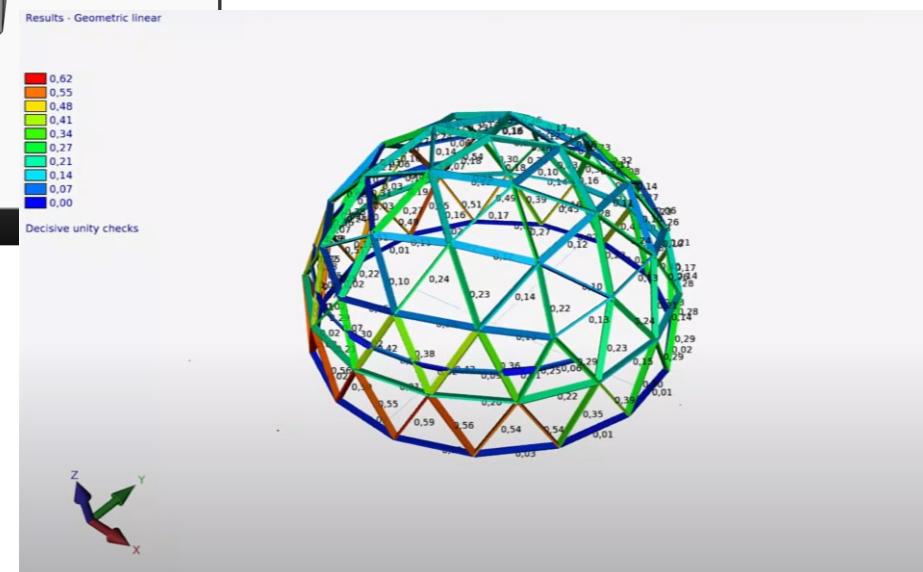
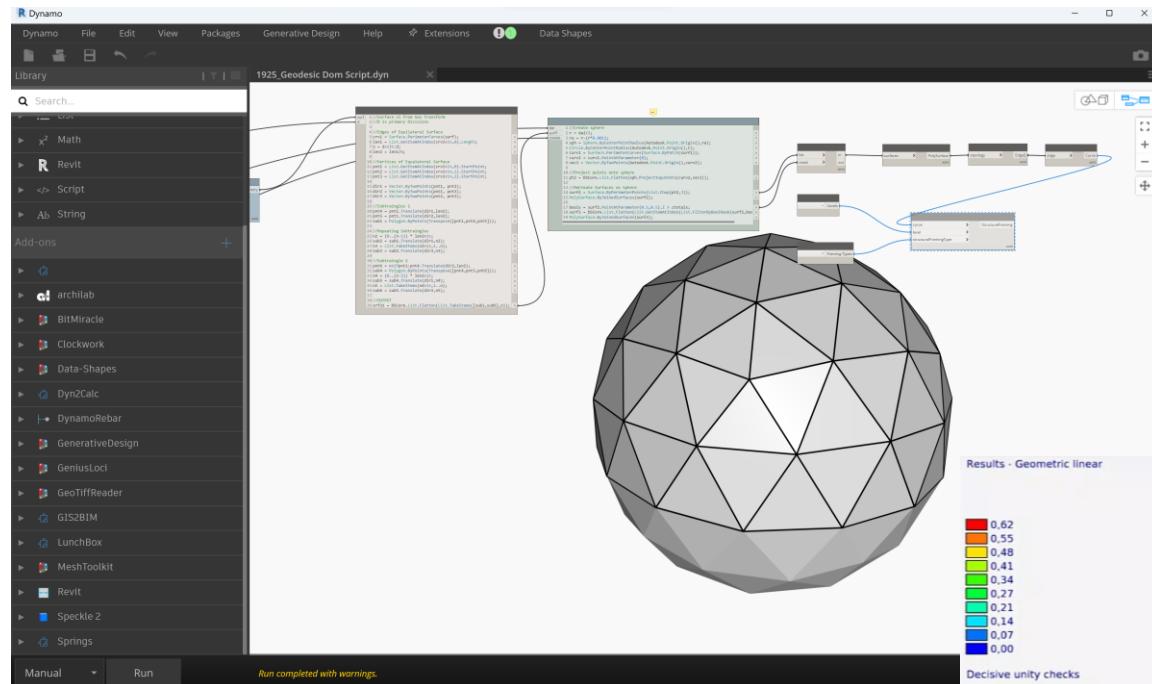
1. Inleiding parametrisch ontwerp

Parametrisch ontwerpen is een **ontwerp methode** waarbij elementen (zoals bouwelementen en technische onderdelen) worden gevormd volgens **algoritmische processen**, in tegenstelling tot een **direct ontwerp**. Bij deze methode bepalen **parameters** en **regels** de relatie tussen **ontwerp intentie** en het **ontwerp resultaat**.

Voorbeeld Grasshopper+Rhino



Voorbeeld Dynamo-XFEM4U



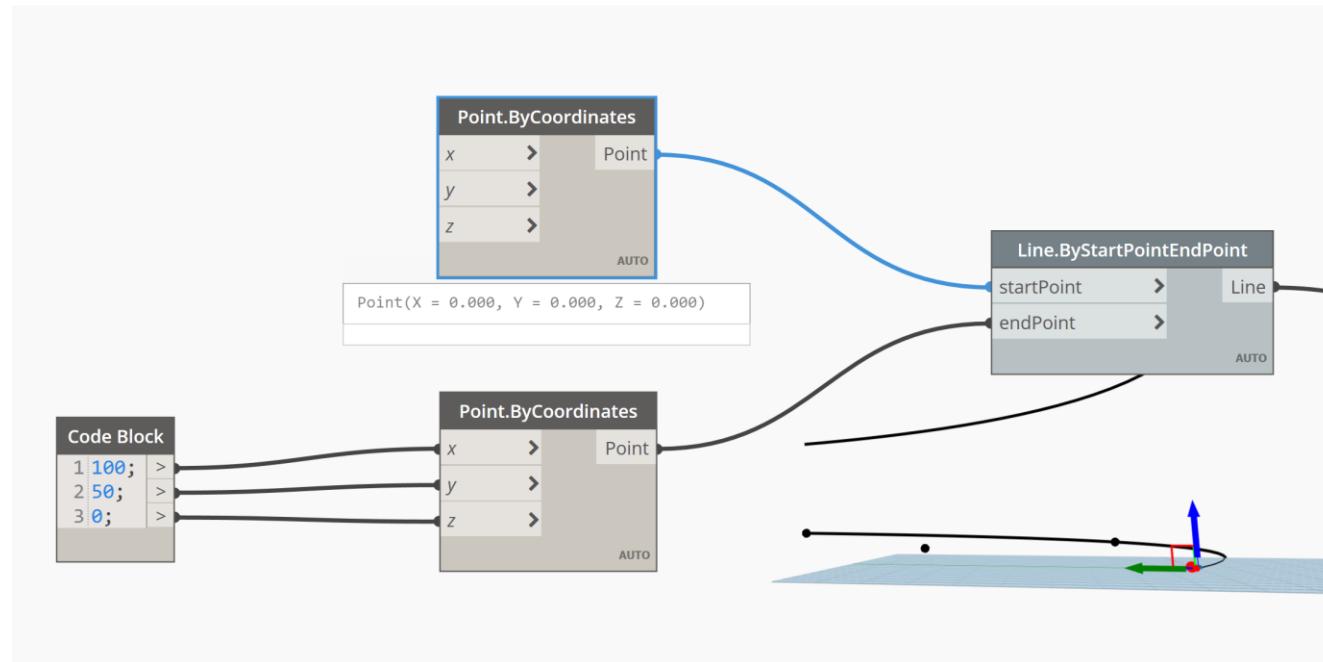
 Struct4U

All together in control

2 Visueel Programmeren

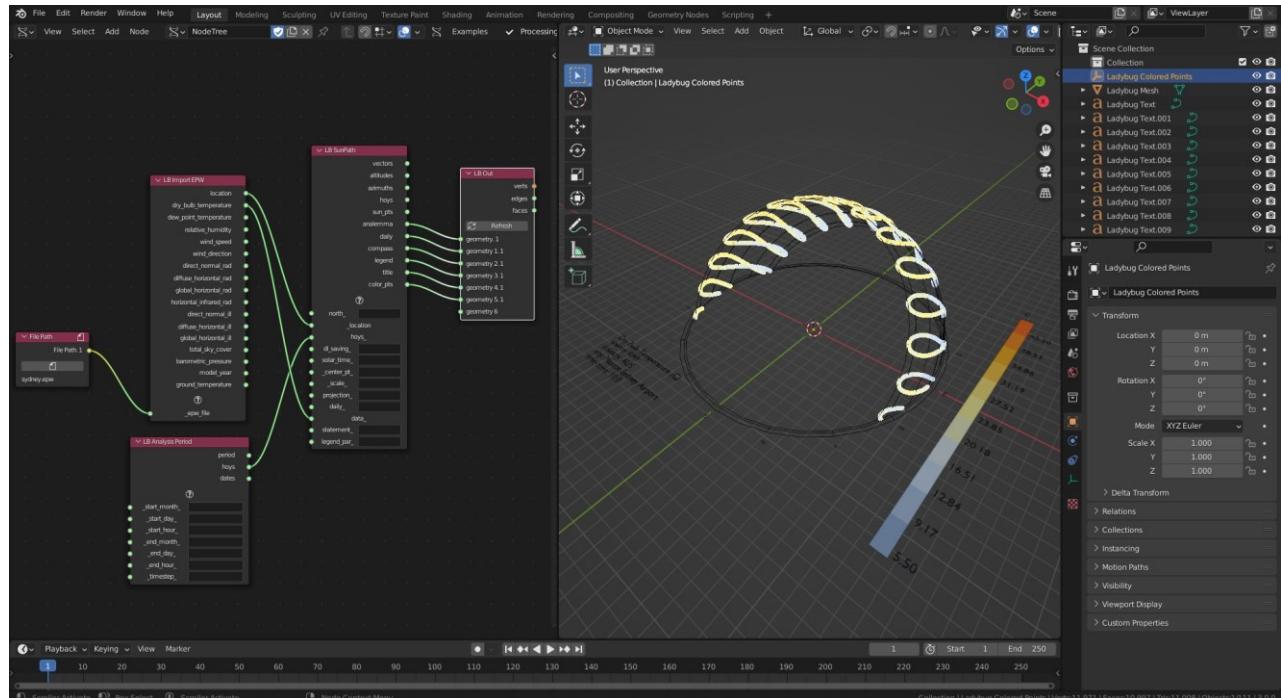
Code Block

```
1 point1 = Point.ByCoordinates(0, 0, 0);  
2 point2 = Point.ByCoordinates(100, 50, 0);  
3 line1 = Line.ByStartPointEndPoint(point1, point2);
```



2 Visueel Programmeren

- 1) Autodesk Revit → Dynamo
- 2) Rhino → Grasshopper
- 3) Blender → Sverchok & Geometry Nodes
- 4) PyFlow(Python)

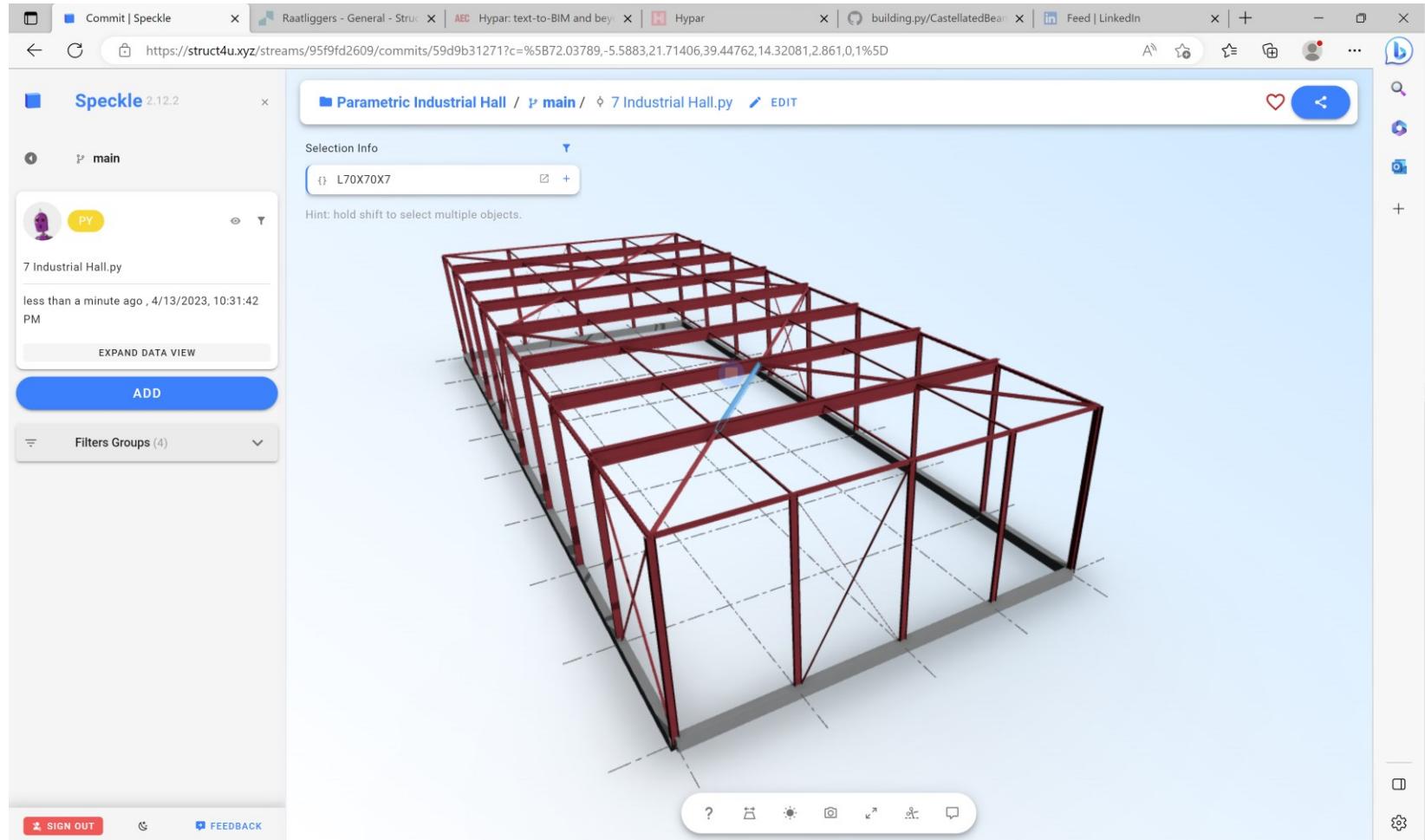


3. Speckle 3D

- Open Source
- Web-based
- Server
- Viewer
- API C#, Python
- Connectors

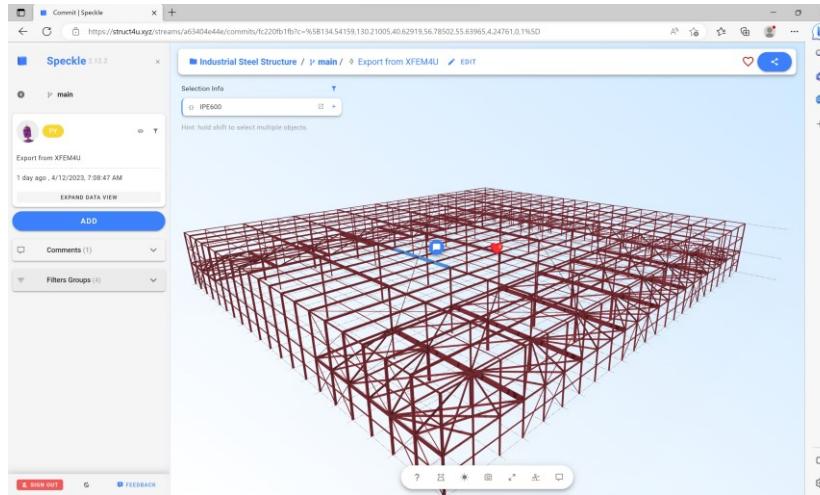


3. Speckle 3D Viewer

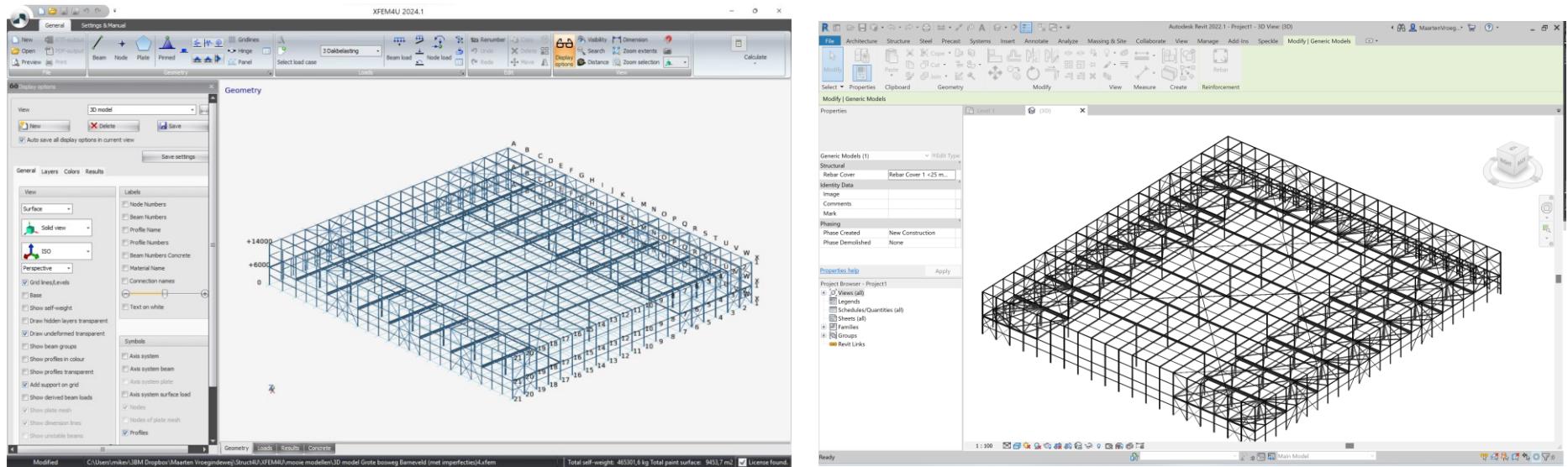


3. Speckle 3D Connectors

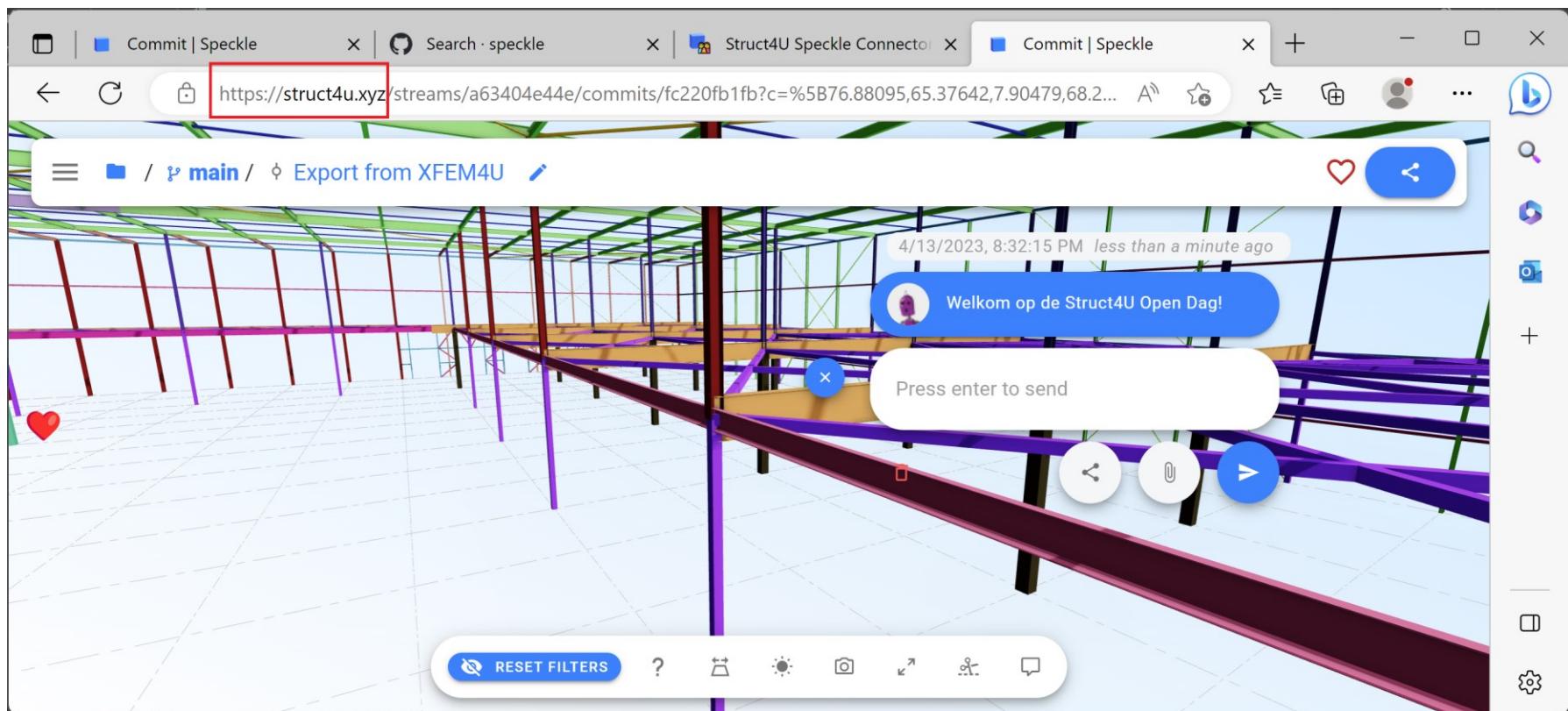
- Revit
- Tekla
- ArchiCAD
- Sketchup



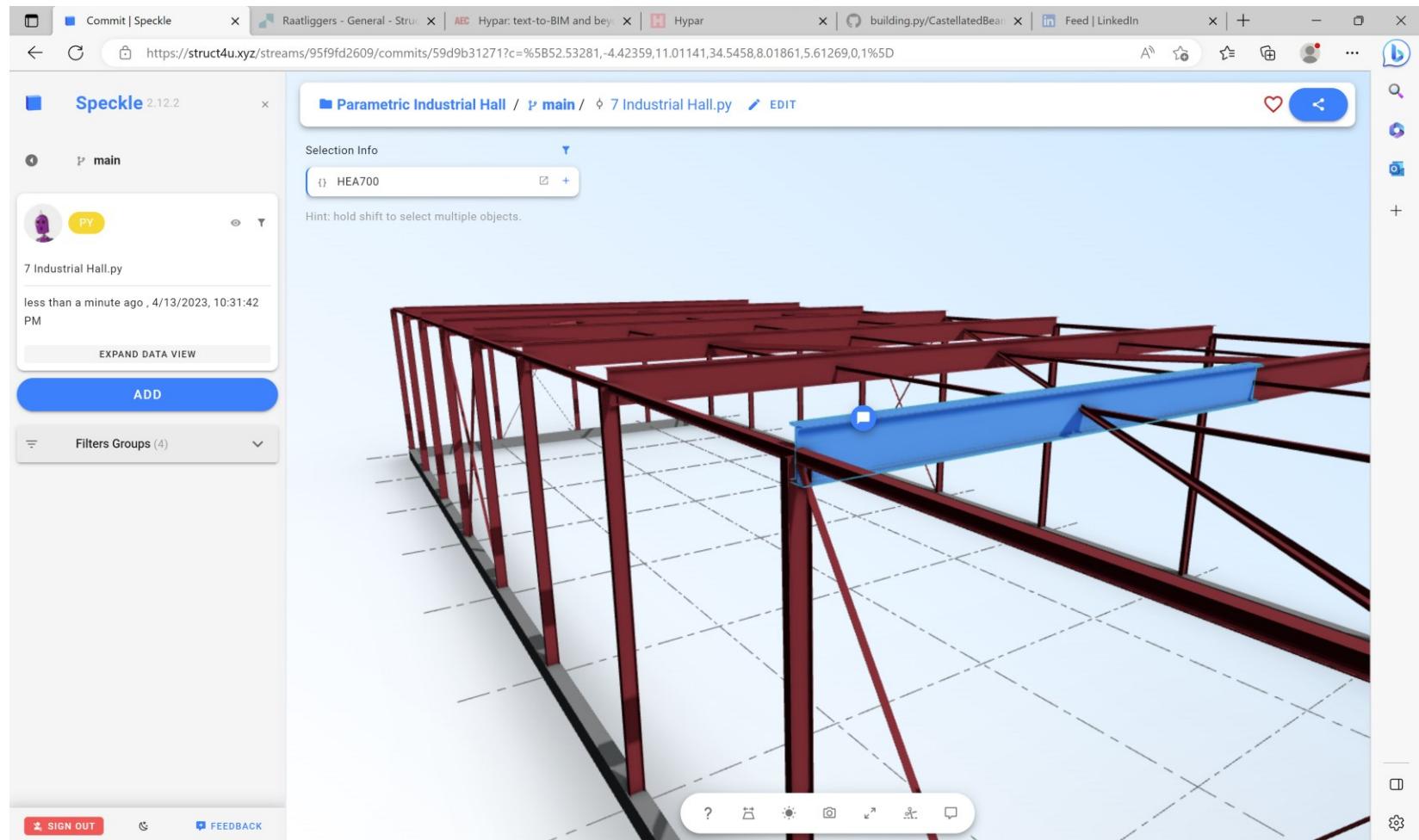
- AutoCAD
- Excel
- XFEM4U
- Navisworks
- Blender



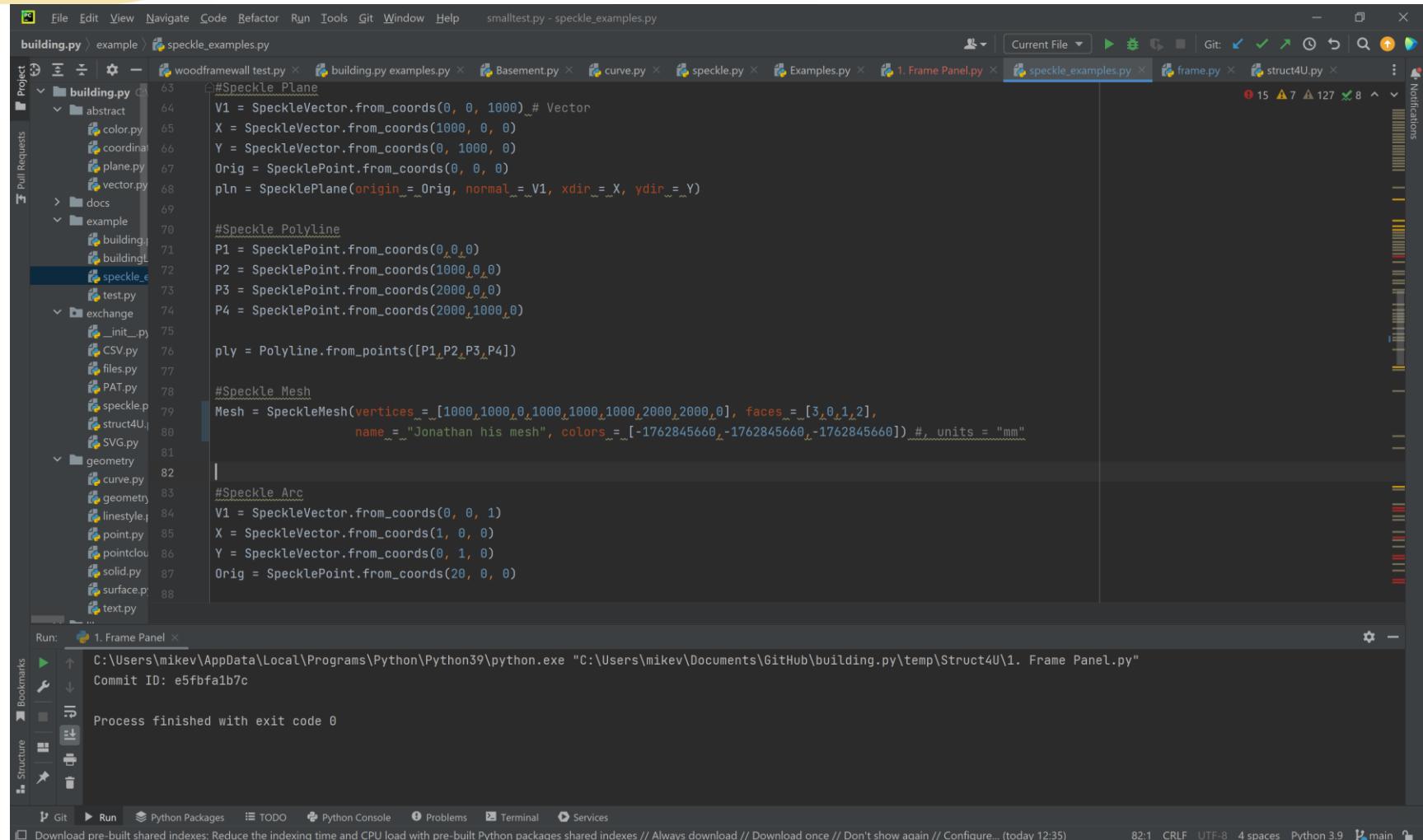
3. Struct4U Speckle Server



3. Speckle 3D Viewer



3. Speckle 3D Python API



The screenshot shows a code editor interface with the following details:

- File Menu:** File, Edit, View, Navigate, Code, Refactor, Run, Tools, Git, Window, Help.
- Current File:** smalltest.py - speckle_examples.py
- Project Explorer:** building.py > example > speckle_examples.py
- Code Content:**

```
#Speckle Plane
V1 = SpeckleVector.from_coords(0, 0, 1000) # Vector
X = SpeckleVector.from_coords(1000, 0, 0)
Y = SpeckleVector.from_coords(0, 1000, 0)
Orig = SpecklePoint.from_coords(0, 0, 0)
pln = SpecklePlane(origin = Orig, normal = V1, xdir = X, ydir = Y)

#Speckle Polyline
P1 = SpecklePoint.from_coords(0,0,0)
P2 = SpecklePoint.from_coords(1000,0,0)
P3 = SpecklePoint.from_coords(2000,0,0)
P4 = SpecklePoint.from_coords(2000,1000,0)

ply = Polyline.from_points([P1,P2,P3,P4])

#Speckle Mesh
Mesh = SpeckleMesh(vertices = [1000,1000,0,1000,1000,1000,2000,2000,0], faces = [3,0,1,2],
                     name = "Jonathan his mesh", colors = [-1762845660,-1762845660,-1762845660]), units = "mm"

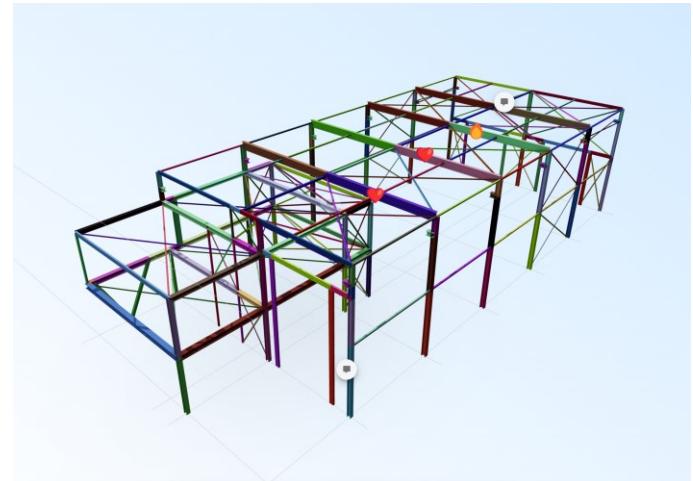
#Speckle Arc
V1 = SpeckleVector.from_coords(0, 0, 1)
X = SpeckleVector.from_coords(1, 0, 0)
Y = SpeckleVector.from_coords(0, 1, 0)
Orig = SpecklePoint.from_coords(20, 0, 0)
```
- Run Tab:** 1. Frame Panel
- Output:** C:\Users\mikev\AppData\Local\Programs\Python\Python39\python.exe "C:\Users\mikev\Documents\GitHub\building.py\temp\Struct4U\1. Frame Panel.py"
Commit ID: e5fbfa1b7c
Process finished with exit code 0
- Bottom Status Bar:** Download pre-built shared indexes: Reduce the indexing time and CPU load with pre-built Python packages shared indexes // Always download // Download once // Don't show again // Configure... (today 12:35) 82:1 CRLF UTF-8 4 spaces Python 3.9 main



All together in control

4 BuildingPy

- Geen afhankelijkheden van bibliotheken anders dan NumPy & SpecklePy
- Classes voor 2D/3D Vector, Point, Line, Arc, PolyCurve, Plane, CoordinateSystem, Surface, Mesh, Solid, Void, Text, Shape, Color, Material.
- Objecten: Frame / Beam, Panel / Plate, Grid, datum
- Analytical: Support, Load, LoadCase, LoadCombination,
- Bibliotheek met staalprofielen incl. synoniemen.
- Kan ook voor visueel programmeren gebruikt worden met PyFlow



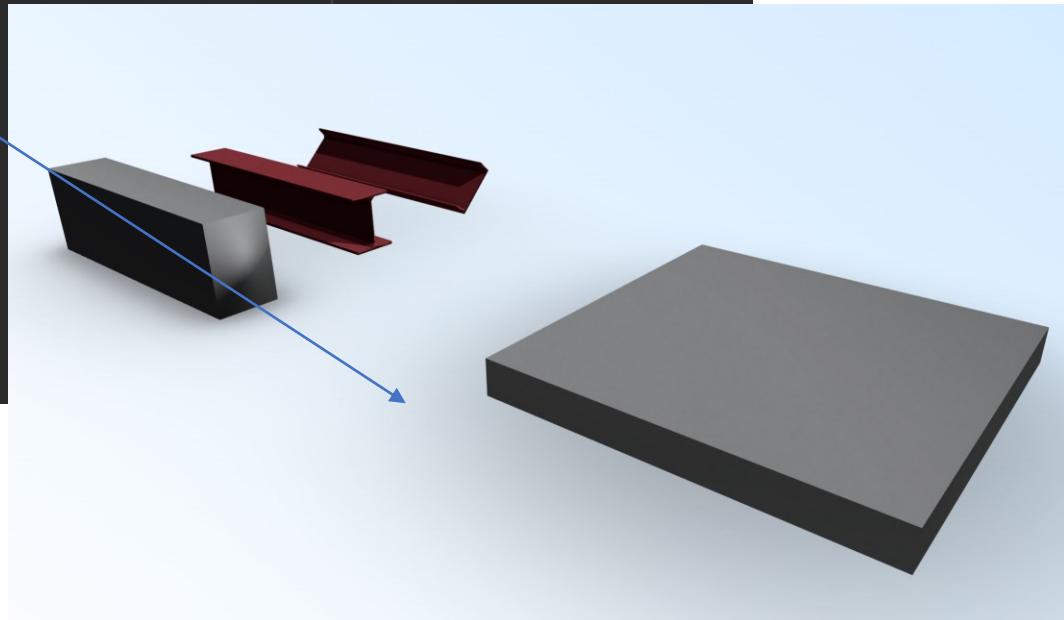
4 BuildingPy

```
#BEAMS
obj.append(Frame.byStartpointEndpoint(Point(0,0,0),Point(2000,0,0),Rectangle("400x600",400,600).curve,"400x600",0,BaseConcrete)) #Concrete Beam
obj.append(Frame.byStartpointEndpointProfileNameShapevector(Point(0,1000,0),Point(2000,1000,0),"HEA400","HEA400",Vector2(0,0),0,BaseSteel)) #Steel Beam

obj.append( #List with elements to Speckle
    Frame.byStartpointEndpointProfileNameShapevector( #Function to create a Frame/Beam
        Point(0,2000,0), #Startpoint
        Point(2000,2000,0), #Endpoint
        "UNP400", #Steel profile name, can be hea400, he400a, HEA400, HEA 400 etc.
        "UNP400-beam",
        Vector2(0,0),
        45, #
        BaseSteel))

#PANELS/ PLATES IN XFEM4U
obj.append(Panel.byPolyCurveThickness(
    PolyCurve.byPoints(
        [Point(4000,0,0),
         Point(6000,0,0),
         Point(6000,2000,0),
         Point(4000,2000,0),
         Point(4000,0,0)]),
    200,
    0,
    "Plate"
    BaseConcrete.colorint))

SpeckleObj = translateObjectsToSpeckleObjects(obj)
Commit = TransportToSpeckle("struct4u.xyz", "558206cfda", SpeckleObj, "Examples.py")
```

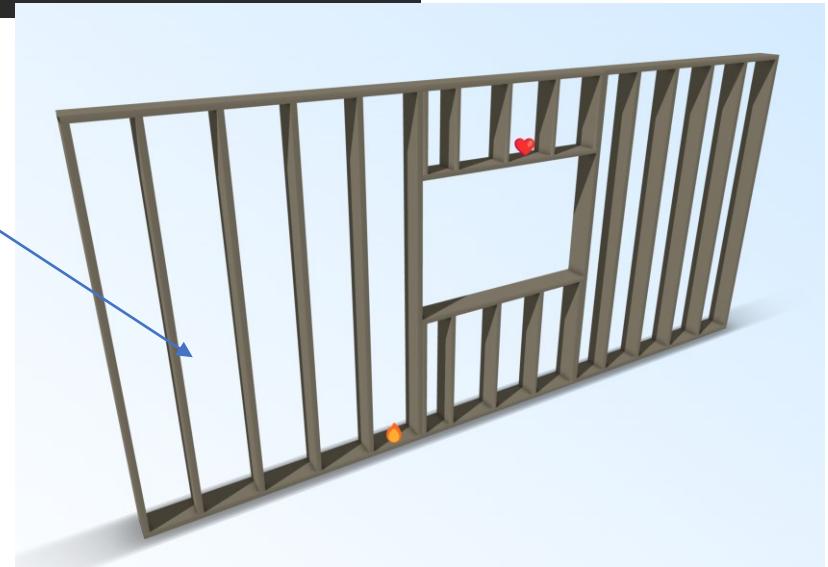


4 BuildingPy

```
a = POCWoodFrameWall(2000, 2600, Point(0, 0, 0), 38, 184, 407)
b = POCWoodFrameWall(1500, 1000, Point(2000, 0, 0), 38, 184, 407)
c = POCWoodFrameWall(1500, 600, Point(2000, 0, 2000), 38, 184, 407)
d = POCWoodFrameWall(2000, 2600, Point(3500, 0, 0), 38, 184, 407)

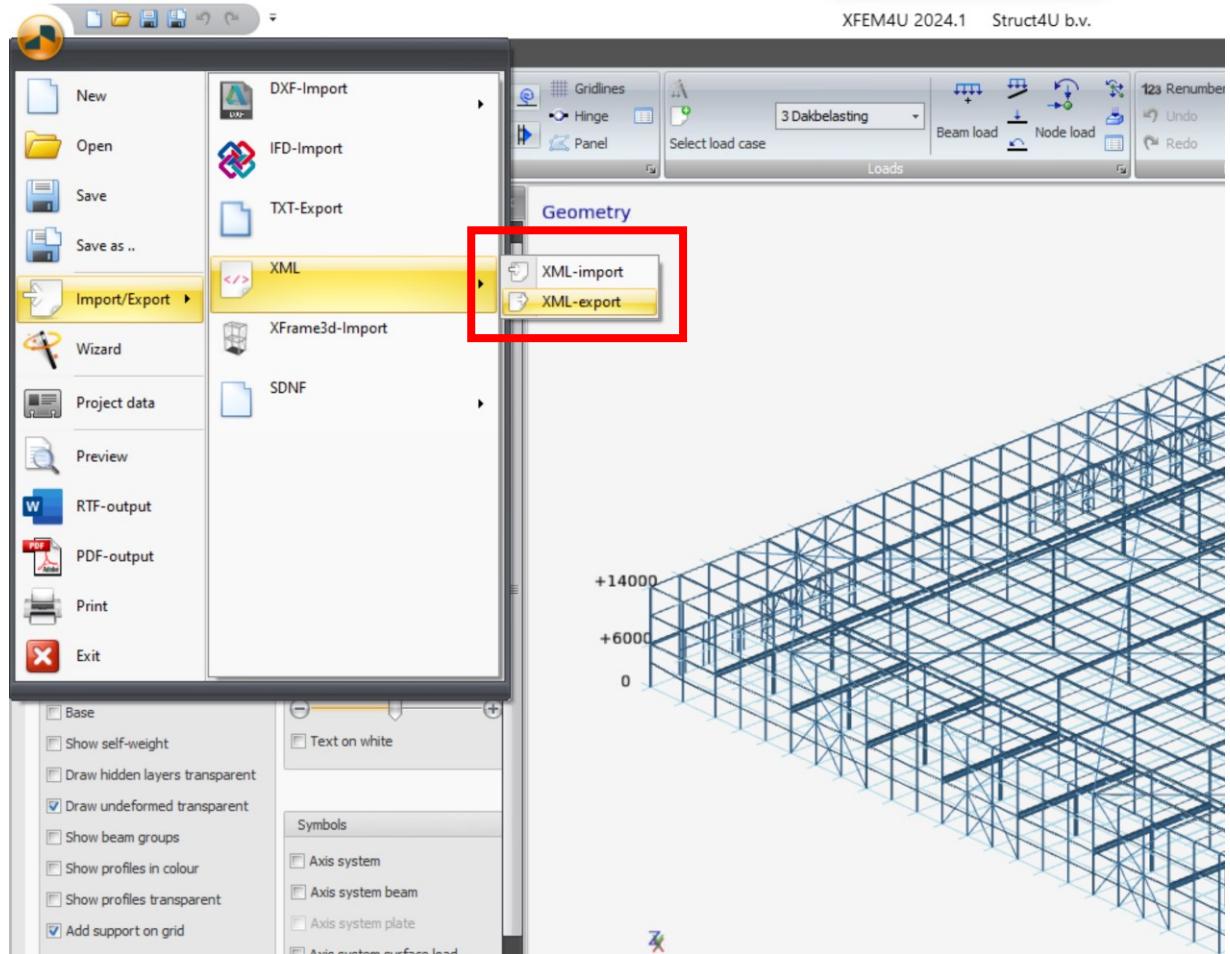
obj = a + b + c + d

SpeckleObj = translateObjectsToSpeckleObjects(obj)
Commit = TransportToSpeckle("struct4u.xyz", "58b53799c6", SpeckleObj, "Test objects")
```

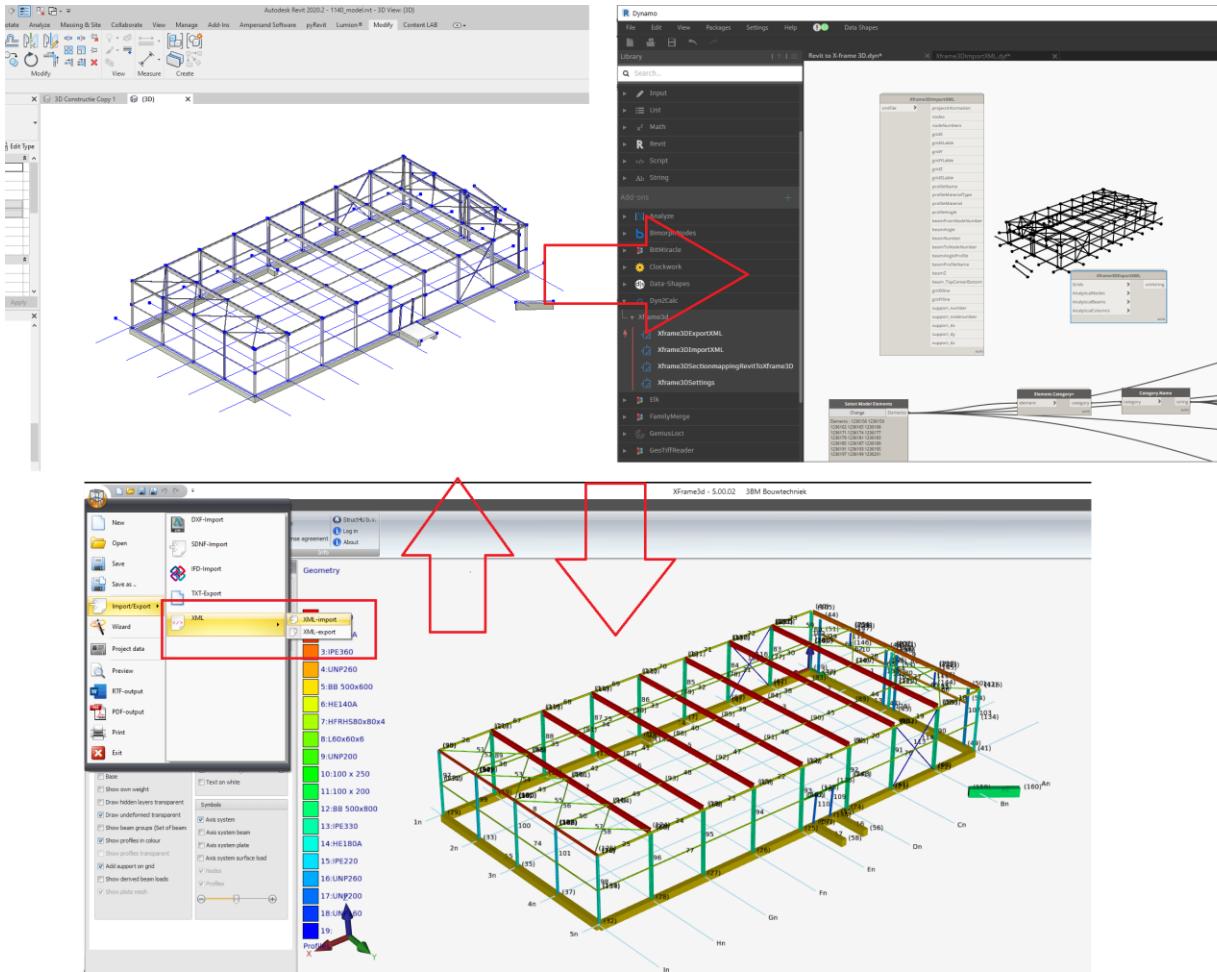


5. XFEM4U

```
<Frame>
  <ProjectName/>
  <ProjectNumber/>
  <ExportDateTime>2023-04-11 23:03:16Z</ExportDateTime>
  <XMLExportVersion>v4.0.30319</XMLExportVersion>
<-Nodes>
  <Number>1</Number>
  <X>0</X>
  <Y>0</Y>
  <Z>0</Z>
  <Number>2</Number>
  <X>0</X>
  <Y>3000</Y>
  <Z>0</Z>
  <Number>3</Number>
  <X>3000</X>
  <Y>0</Y>
  <Z>0</Z>
  <Number>4</Number>
  <X>3000</X>
  <Y>3000</Y>
  <Z>0</Z>
</Nodes>
<+Supports></Supports>
<+Grids></Grids>
<+Profiles></Profiles>
<+Beamgroup></Beamgroup>
<+Beams></Beams>
<+Panels></Panels>
<+LoadCases></LoadCases>
<+BeamLoads/>
<+SurfaceLoads></SurfaceLoads>
<+Combinations></Combinations>
<+RebarLongitudinal/>
<+RebarStirrup/>
<+Layers></Layers>
</Frame>
```



5. XFEM4XML-Revit



Voorbeelden

- 1) Frame, Plate naar Speckle
- 2) Raatligger
- 3) Kelder
- 4) Betonvloer met schaakbordbelasting
- 5) Trap met liggers
- 6) Trap met platen
- 7) Hal
- 8) XFEM4Unaar Speckle
- 9) Geodetische koepel via Dynamo
- 10) Paalposities vanuit XFEM4Unaar Revit
- 11) Samenwerking met derden.
- 12) Download

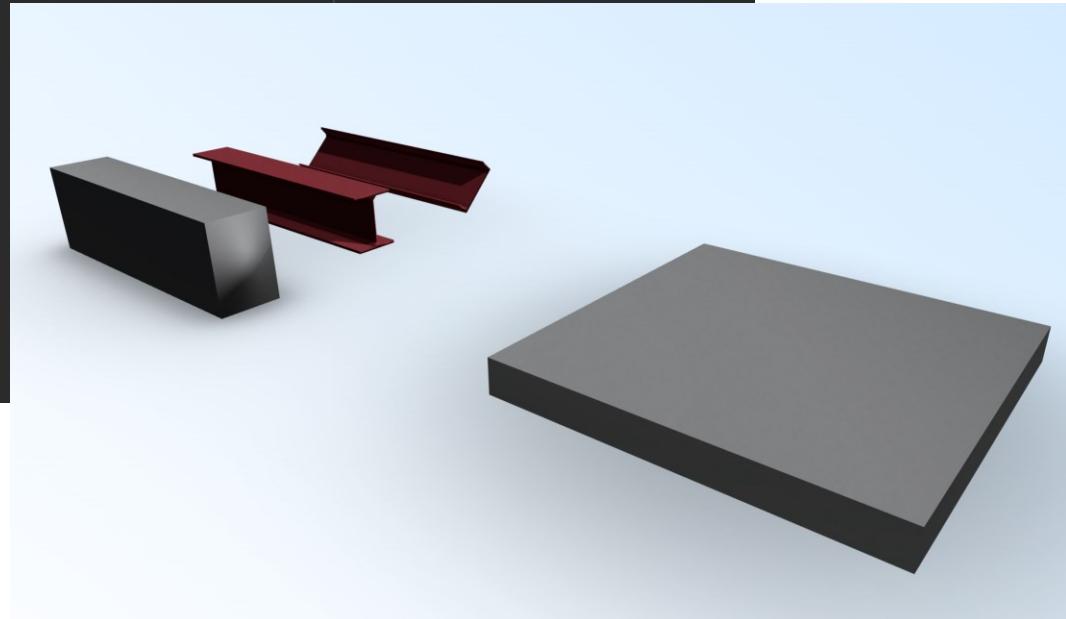
1. Frame, Plate naar Speckle

```
#BEAMS
obj.append(Frame.byStartpointEndpoint(Point(0,0,0),Point(2000,0,0),Rectangle("400x600",400,600).curve,"400x600",0,BaseConcrete)) #Concrete Beam
obj.append(Frame.byStartpointEndpointProfileNameShapevector(Point(0,1000,0),Point(2000,1000,0),"HEA400","HEA400",Vector2(0,0),0,BaseSteel)) #Steel Beam

obj.append( #List with elements to Speckle
    Frame.byStartpointEndpointProfileNameShapevector( #Function to create a Frame/Beam
        Point(0,2000,0), #Startpoint
        Point(2000,2000,0), #Endpoint
        "UNP400", #Steel profile name, can be hea400, he400a, HEA400, HEA 400 etc.
        "UNP400-beam",
        Vector2(0,0),
        45, #
        BaseSteel))

#PANELS/ PLATES IN XFEM4U
obj.append(Panel.byPolyCurveThickness(
    PolyCurve.byPoints(
        [Point(4000,0,0),
         Point(6000,0,0),
         Point(6000,2000,0),
         Point(4000,2000,0),
         Point(4000,0,0)]),
    200,
    0,
    "Plate"
    BaseConcrete.colorint))

SpeckleObj = translateObjectsToSpeckleObjects(obj)
Commit = TransportToSpeckle("struct4u.xyz", "558206cfda", SpeckleObj, "Examples.py")
```



2 Raatlijger(stijfheid)

```
nm = "IPE600 Castellated Beam"
b = 220 #width
tw = 12
tf = 19
h = 750 #height_beam
h1 = 150 #height above opening
b1 = 350 #minimum width of the opening
b2 = 550 #maximum width of the opening
spac = 850 #spacing of the openings
l = 15000 #length

# Maarten Vroegindeweij
def CastellatedBeam(b,h,h1,b1,b2,spac,l):
    ...

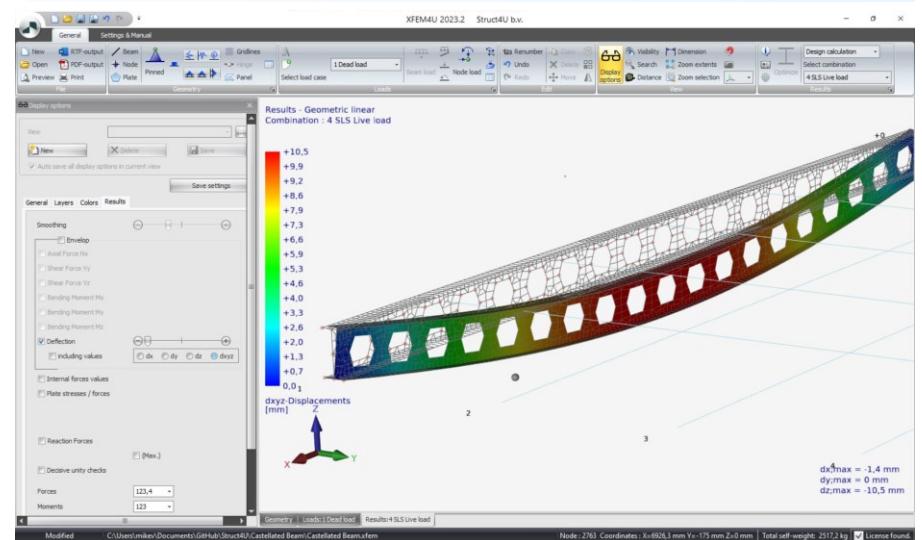
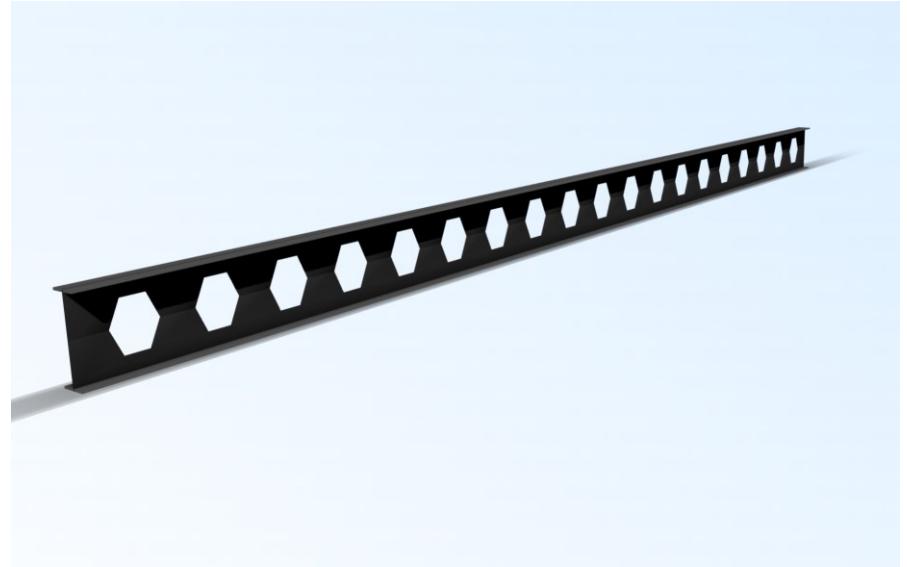
CB = CastellatedBeam(b,h,h1,b1,b2,spac,l)

obj = []

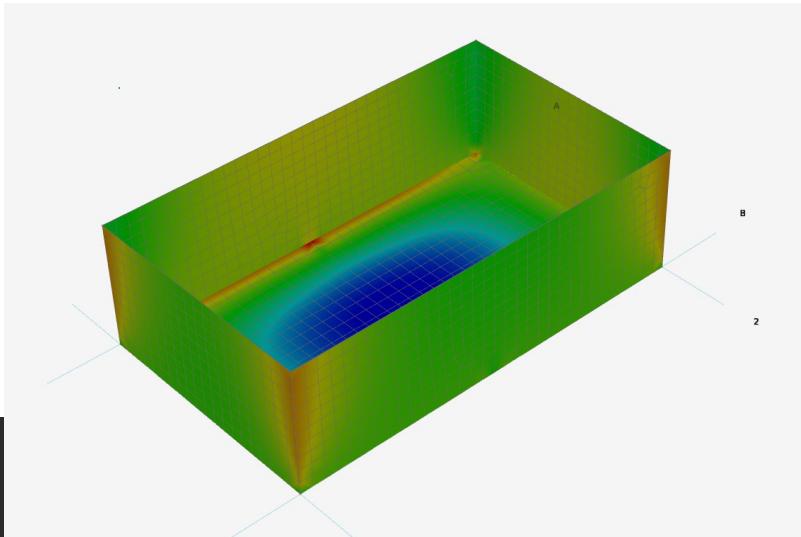
obj.append(Panel.byPolyCurveThickness(CB[0],tf,"top",rgb_to_int([192, 192, 192])))
obj.append(Panel.byPolyCurveThickness(CB[1],tf,0,"bottom",rgb_to_int([192, 192, 192])))
obj.append(Panel.byPolyCurveThickness(CB[2],tw,0,"middle",rgb_to_int([192, 192, 192])))
obj.append(Panel.byPolyCurveThickness(CB[3],tw,0,"middle",rgb_to_int([192, 192, 192])))

SpeckleObj = translateObjectsToSpeckleObjects(obj)

Commit = TransportToSpeckle("struct4u.xyz", "eb801b33ca", SpeckleObj, "Castellated Beam")
```



3. Prefab bouwproducten

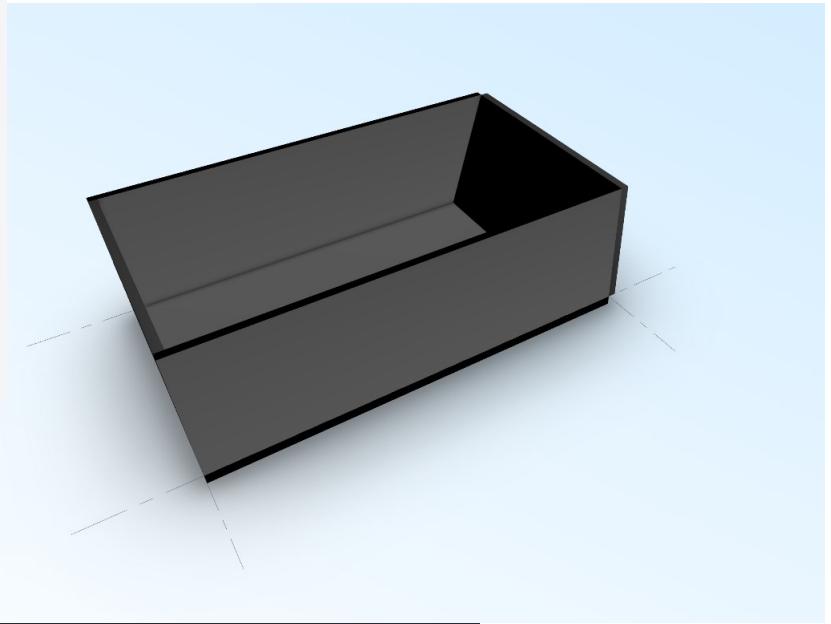


```
#INPUT.in.mm
width = 5000
length = 8500
height = 2600
floorThickness = 200
wallThickness = 160

#Grids
xstr = "0 " + str(width)
ystr = "0 " + str(length)
grids = GridSystem(xstr, seqX, seqY, ext)
obj = grids[0] + grids[1].listWithObjects

#ConcreteFloor
obj.append(Panel_byPolyCurveThickness([
    PolyCurve.byPoints(
        [Point(0,0,0),
         Point(width,0,0),
         Point(width,length,0),
         Point(0,length,0),Point(0,0,0)]),
    floorThickness,
    0,
    "Floor",
    BaseConcrete.colorint)])
obj.append(Panel_byPolyCurveThickness(PolyCurve.byPoints([Point(0,0,0),Point(width,0,0),Point(width,0,height),Point(0,0,height),Point(0,0,0)]),wallThickness,0,"Wall 1",BaseConcrete.colorint)).# Wall 1
obj.append(Panel_byPolyCurveThickness(PolyCurve.byPoints([Point(0,length,0),Point(width,0,0),Point(width,length,0),Point(0,length,0),Point(0,0,0)]),wallThickness,0,"Wall 2",BaseConcrete.colorint)).# Wall 2
obj.append(Panel_byPolyCurveThickness(PolyCurve.byPoints([Point(0,0,0),Point(0,length,0),Point(0,length,height),Point(0,0,height),Point(0,0,0)]),wallThickness,0,"Wall 3",BaseConcrete.colorint)).# Wall 3
obj.append(Panel_byPolyCurveThickness(PolyCurve.byPoints([Point(width,0,0),Point(width,length,0),Point(width,length,height),Point(width,0,height),Point(width,0,0)]),wallThickness,0,"Wall 4",BaseConcrete.colorint)).# Wall 4

#Export to Speckle
SpeckleObj = translateObjectsToSpeckleObjects(obj)
Commit = TransportToSpeckle("struct4u.xyz", "9fd1692151", SpeckleObj, "Parametric Concrete Basement")
```

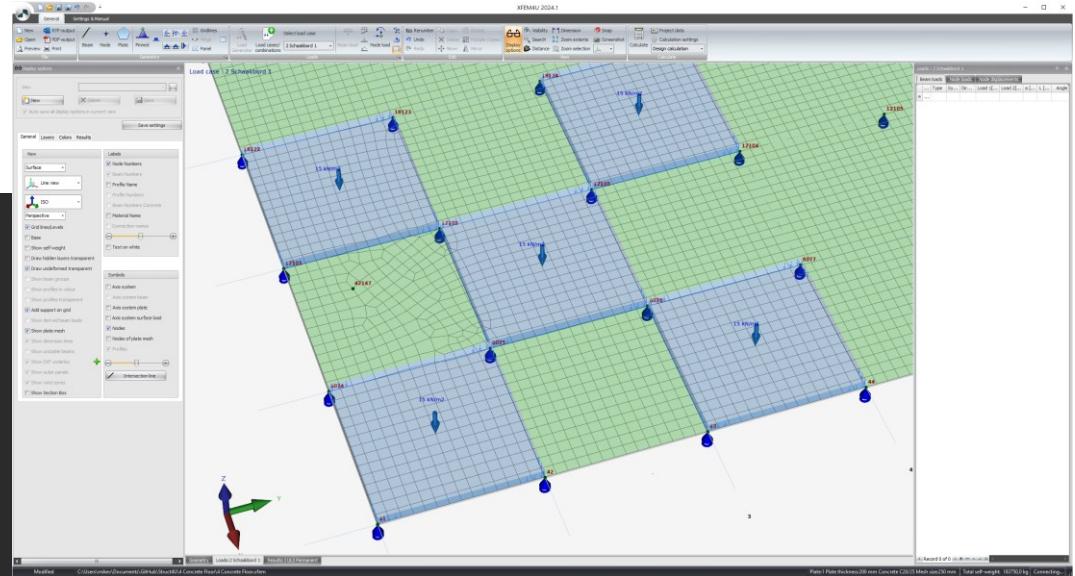


4. Schaakbordbelasting

```
#Concretefloor
obj.append(Panel.byPolyCurveThickness(
    Rect(Vector3(0,0,0),width,height),
    floorthickness,
    0,
    "Concrete Floor",
    BaseConcrete.colorint))

#Supports
x = 0
y = 0
for i in range(nx):
    for j in range(ny):
        sup = Support()
        sup.Point = Point(x,y,0)

x = 0
for i in range(nx+1):
    y = 0
    for j in range(ny+1):
        sup = Support()
        sup.Point = Point(x, y, 0)
        sup.Tz = "S"
        sup.Kz = 35000
        obj.append(sup)
        y = y + spac_y
    x = x + spac_x
```



5. Trap met liggers

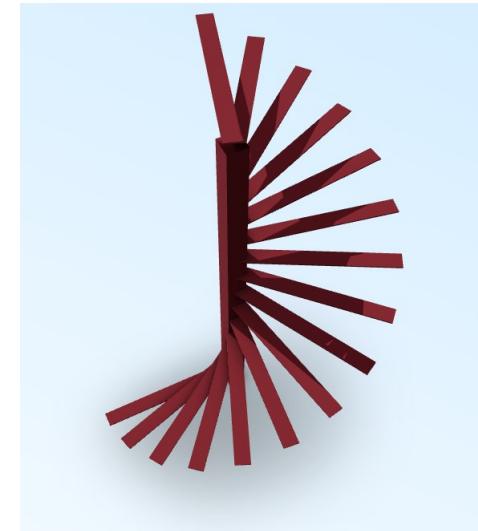
```
Height = 3000
Radius = 1200
Numberoftreads = 16
DegreesSpiral = 270
Column = "K150/150/10"
Tread = "S120/10"
obj = []

obj.append(Frame.byStartpointEndpointProfileName(Point(0,0,0), Point(0,0,Height), Column, "spiral", BaseSteel)) #Spiral

deg = 0
x = 0
y = Radius
dz = Height / Numberoftreads
z = dz
p0 = Point(x, y, z)
ddeg = DegreesSpiral / Numberoftreads

for i in range(Numberoftreads):
    obj.append(Frame.byStartpointEndpointProfileNameJustification(Point(0, 0, z), p2, Tread, "Tread", "Center", "Center", 90, BaseSteel)) # Treads
    p2 = Point.rotateXY(p2, ddeg, dz)
    z = z + dz

SpeckleObj = translateObjectsToSpeckleObjects(obj)
Commit = TransportToSpeckle("struct4u.xyz", "4bb051dcbb", SpeckleObj, "Parametric Spiral Staircase.py")
```



6. Trap met plaatelementen

```
p1 = Point(0, RadiusSpiral, z)
p11 = Point.translate(p1, Vector3(0,0,-HeightTreadSupportStart))
p4 = Point(0, Radius, z)
p41 = Point.translate(p4, Vector3(0,0,-HeightTreadSupportEnd))

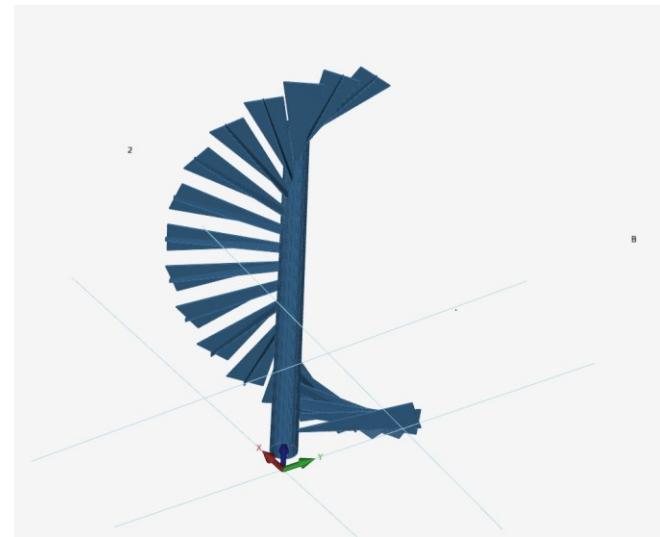
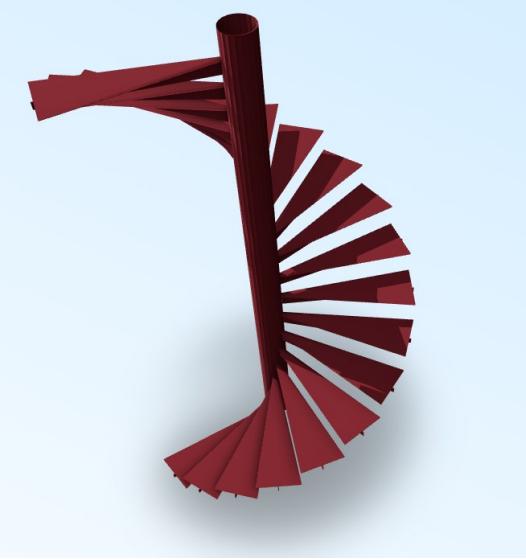
p2 = Point.rotateXY(p1, ddeg * 0.5, 0)
p3 = Point.rotateXY(p1, -ddeg * 0.5, 0)
p5 = Point.rotateXY(p4, ddeg * 0.5, 0)
p6 = Point.rotateXY(p4, -ddeg * 0.5, 0)
TreadCurve = PolyCurve.byPoints([p2,p5,p6,p3,p2])
TreadSupportCurve = PolyCurve.byPoints([p1,p4,p41,p11,p1])

SpiralSegmentCurve = PolyCurve.byPoints([p3, p2, Point.translate(p2, Vector3(0,0,Height)), Point.translate(p3, Vector3(0,0,Height))])

for i in range(NumberOfTreads):
    obj.append(Panel.byPolyCurveThickness(TreadCurve, ThicknessTread, 0, "Tread", BaseSteel.colorint))
    obj.append(Panel.byPolyCurveThickness(TreadSupportCurve, ThicknessSteel, 0, "SupportTread", BaseSteel.colorint))
    obj.append(Panel.byPolyCurveThickness(SpiralSegmentCurve, ThicknessSteel, 0, "SpiralSegment", BaseSteel.colorint))

    TreadCurve = TreadCurve.rotate(ddeg, dz)
    TreadSupportCurve = TreadSupportCurve.rotate(ddeg, dz)
    SpiralSegmentCurve = SpiralSegmentCurve.rotate(ddeg, 0)

SpeckleObj = translateObjectsToSpeckleObjects(obj)
Commit = TransportToSpeckle("struct4u.xyz", "a4d620a049", SpeckleObj, "Parametric Spiral Staircase.py")
```



7. Hal

```
# GridSystem
seqX = "A B C D E F G H I J K L M N O P Q R S T U V W X Y Z AA AB AC"
seqY = "1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24"
ext = 2500 #extension grid

#INPUT in mm
spac = 6500 #grid spacing 1
n = 7 # number of grids 1

spac_y = 5200 #grid spacing 2
nw = 4 # number of grids 2

z = 9000 #height of the structure
afschat = 0

GEVELKOLOM = "IPE330"
HOOFDLIGGER = "HEA700"
RANDLIGGER = "HEA160"
KOPPELLIGGER = "HFRHS80x80x5"
HOEKKOLOM = "HEA300"
KOPGEVELKOLOM = "IPE300"
RANDLIGGER_KOPGEVEL = "HEA160"
WVB_DAK = "L70X70X7"
WVB_GEVEL = "S100x5"
FOUNDATIONBEAM = Rectangle("FB 400x600", 400, 600).curve
```

```
#WINDVERBANDEN GEVEL
wvb = [
    ["K1", 2, 1],
    ["K2", 2, 1],
    ["L1", 4, 1],
    ["L1", 8, 1],
    ["L2", 4, 1]

]

#WINDVERBANDEN DAK
wvbDak = [
    [1, 1, 1], #Stramienvak X, Stramienvak Y, 0=R1, 1=R2, 2=R1,R2
    [2, 2, 1],
    [3, 3, 1],
    [4, 4, 1],
    [5, 4, 0],
    [6, 3, 0],
    [7, 2, 0],
    [8, 1, 0],
    [1, 4, 0],
    [2, 3, 0],
    [3, 2, 0],
    [4, 1, 0],
    [5, 1, 1],
    [6, 2, 1],
    [7, 3, 1],
    [8, 4, 1]
]
```

7. Hal

```
#MODELERING
x = spac #stramienmaat
y = spac_y*nw #width hall
width = y
l = (n+1) * spac
length = l

spacX = str(n+1) + "x" + str(spac) # "13x5400"
spacY = str(nw) + "x" + str(spac_y) # "4x5400"
grids = GridSystem(spacX,seqX,spacY,seqY,ext)
obj1 = grids[0] + grids[1]

#obj1 = []
#SPANTEN
for i in range(n):
    obj1.append(Frame.byStartpointEndpointProfileName(Point(x, 0, z), Point(x, y*0.5, z+afschot), HOOFDLIGGER, "Hoofdligger deel 1", BaseSteel))
    obj1.append(Frame.byStartpointEndpointProfileName(Point(x, y*0.5, z+afschot), Point(x, y, z), HOOFDLIGGER, "Hoofdligger deel 2", BaseSteel))
    obj1.append(Frame.byStartpointEndpointProfileName(Point(x, 0, 0), Point(x, 0, z), GEVELKOLOM, "Kolom 1", BaseSteel))
    obj1.append(Frame.byStartpointEndpointProfileName(Point(x, y, 0), Point(x, y, z), GEVELKOLOM, "Kolom 2", BaseSteel))
    obj1.append(Support.pinned(Point(x, y, 0)))
    obj1.append(Support.pinned(Point(x, 0, 0)))
    x = x + spac
|
#FOUNDATION BEAM
obj1.append(Frame.byStartpointEndpoint(Point(0,0,0), Point(0,y,0), FOUNDATIONBEAM, "FB 1", BaseConcrete))
obj1.append(Frame.byStartpointEndpoint(Point(l,0,0), Point(l,y,0), FOUNDATIONBEAM, "FB 1", BaseConcrete))
obj1.append(Frame.byStartpointEndpoint(Point(0,0,l), Point(l,0,0), FOUNDATIONBEAM, "FB 1", BaseConcrete))
obj1.append(Frame.byStartpointEndpoint(Point(0,y,0), Point(l,y,0), FOUNDATIONBEAM, "FB 1", BaseConcrete))
```

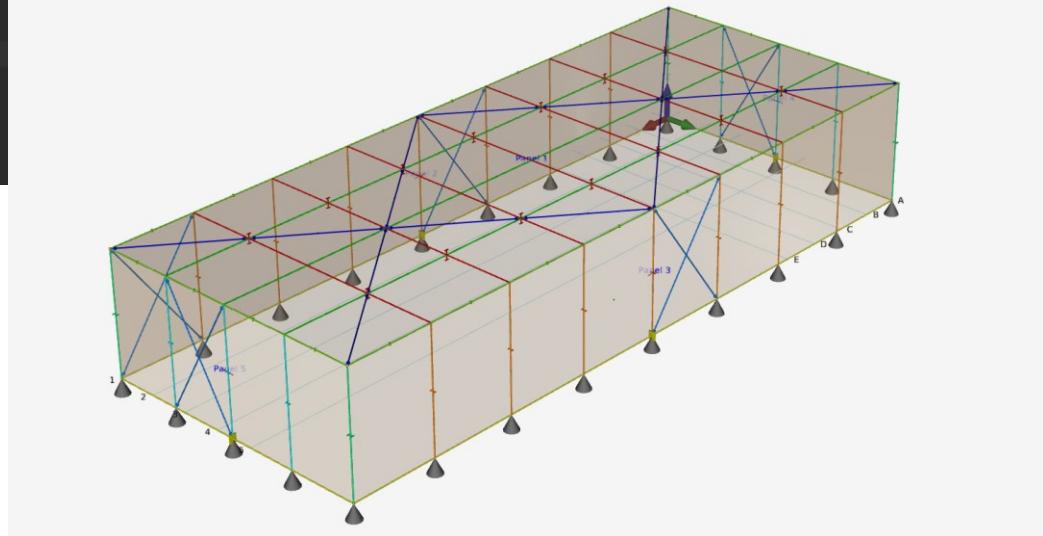
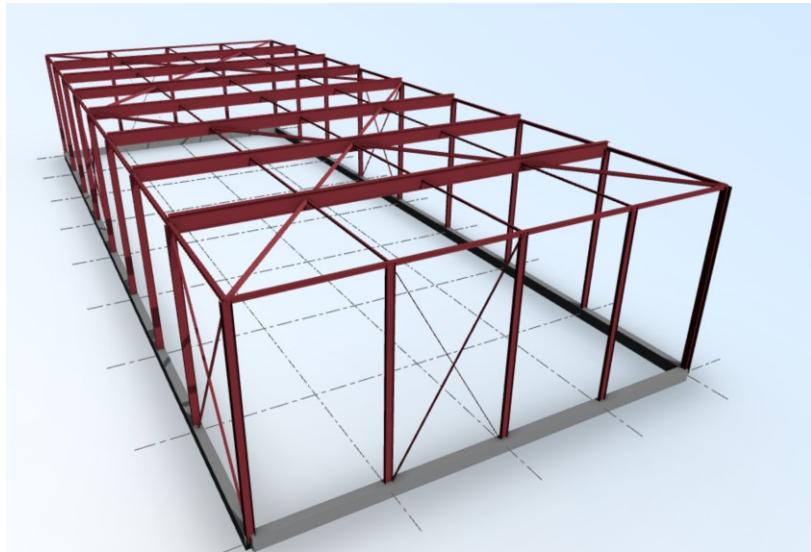
7. Hall

```
SpeckleObj = translateObjectsToSpeckleObjects(obj1)
Commit = TransportToSpeckle("struct4u.xyz", "95f9fd2609", SpeckleObj, "7 Industrial Hall.py")

#Export to XFEM4U XML String

xmlS4U = xmlXFEM4U() # Create XML object with standard values
xmlS4U.addBeamsPlates(obj1) #Add Beams, Profiles, Plates, Beamgroups, Nodes
xmlS4U.addProject("Parametric Industrial Hall")
xmlS4U.addPanels(obj1) #add Load Panels
xmlS4U.addGrids(spacX,seqX,spacY,seqY,z) # Grids
xmlS4U.addSurfaceLoad(obj1)
xmlS4U.addLoadCasesCombinations()
xmlS4U.XML()
XMLString = xmlS4U.xmlstr

filepath = "C:/Users/mikev/Documents/GitHub/Struct4U/7 Industrial Hall/Hall.xml"
file = open(filepath, "w")
a = file.write(XMLString)
file.close()
```



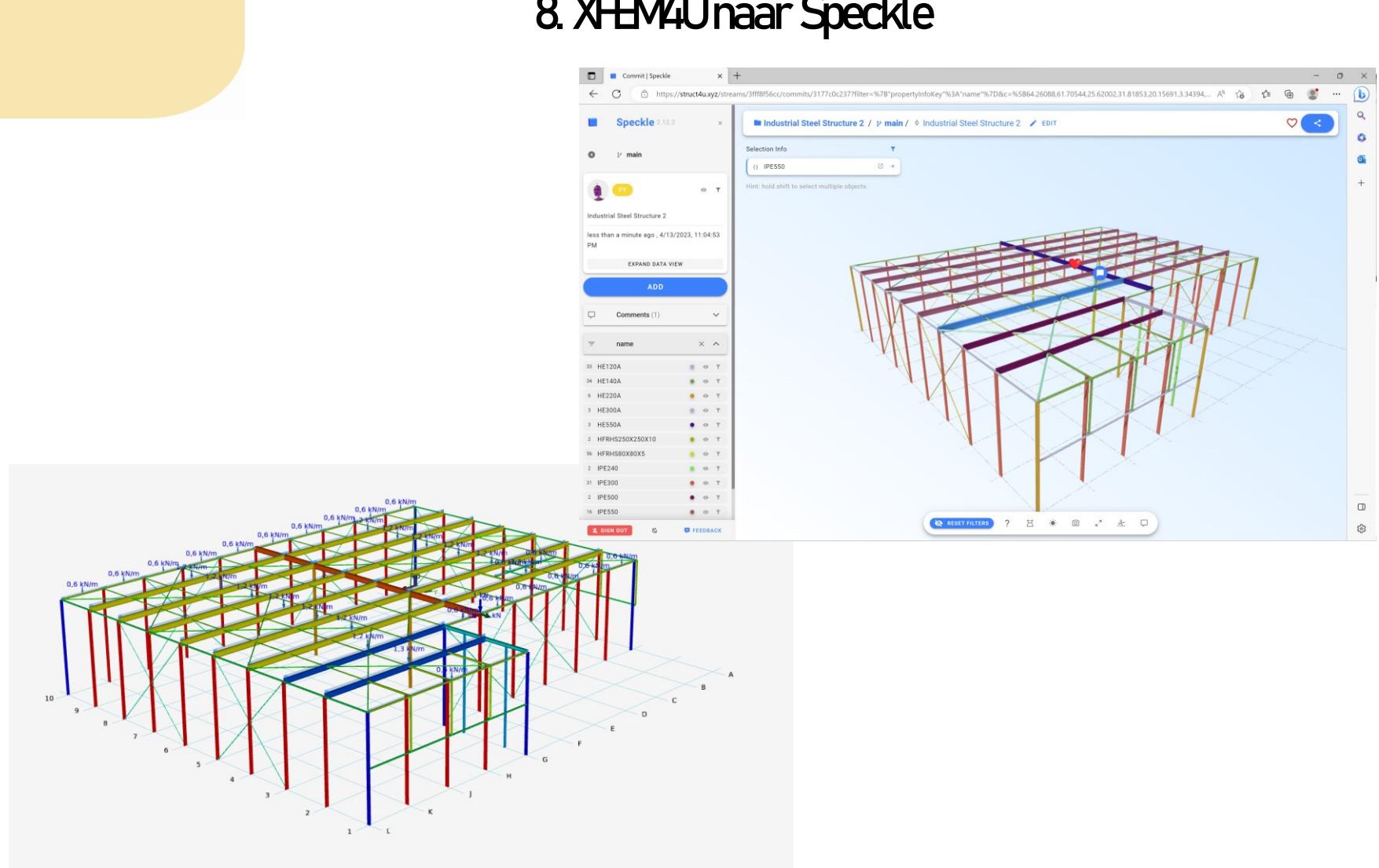
8. XFEM4U naar Speckle

The screenshot shows a Python code editor interface with the following details:

- File Path:** building.py > temp > Struct4U > 8 XFEM4U to Speckle.py
- Code Content:** The script translates XFEM4U objects (BeamsLayer, BeamsRotation, Profiles, Beams) into Speckle objects. It uses the Speckle Python library to perform the translation.
- Project Explorer:** Shows files like analytical.py, datum.py, frame.py, loads.py, panel.py, shape.py, steelshape.py, packages, sandbox, and various .py files under temp/Struct4U.
- Run Tab:** Displays the command run: 8 XFEM4U to Speckle, the path C:\Users\mikev\AppData\Local\Programs\Python\Python39\python.exe, and the commit ID: 3177c0c237.
- Status Bar:** Shows PEP 8: E265 block comment should start with '#!', file size 20:20, CR/LF, UTF-8, 4 spaces, Python 3.9, and main module.

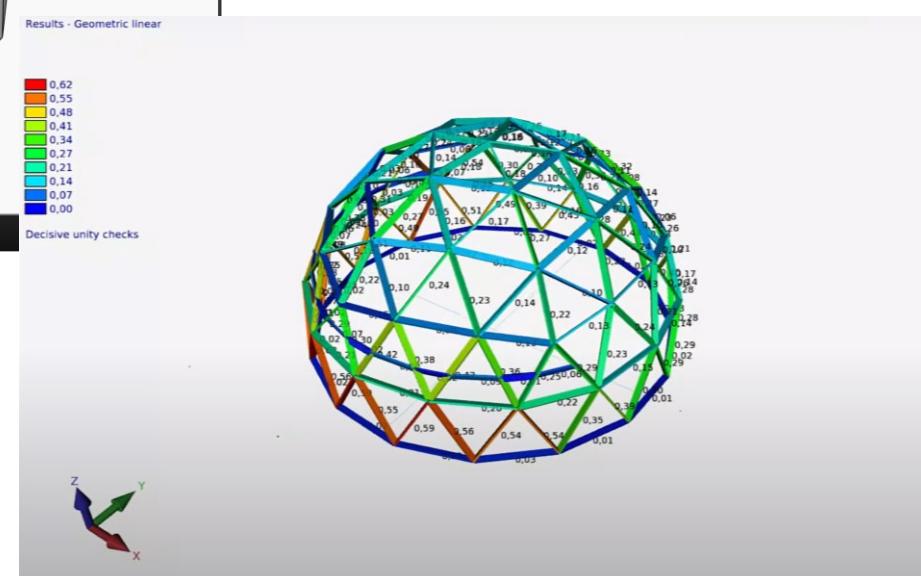
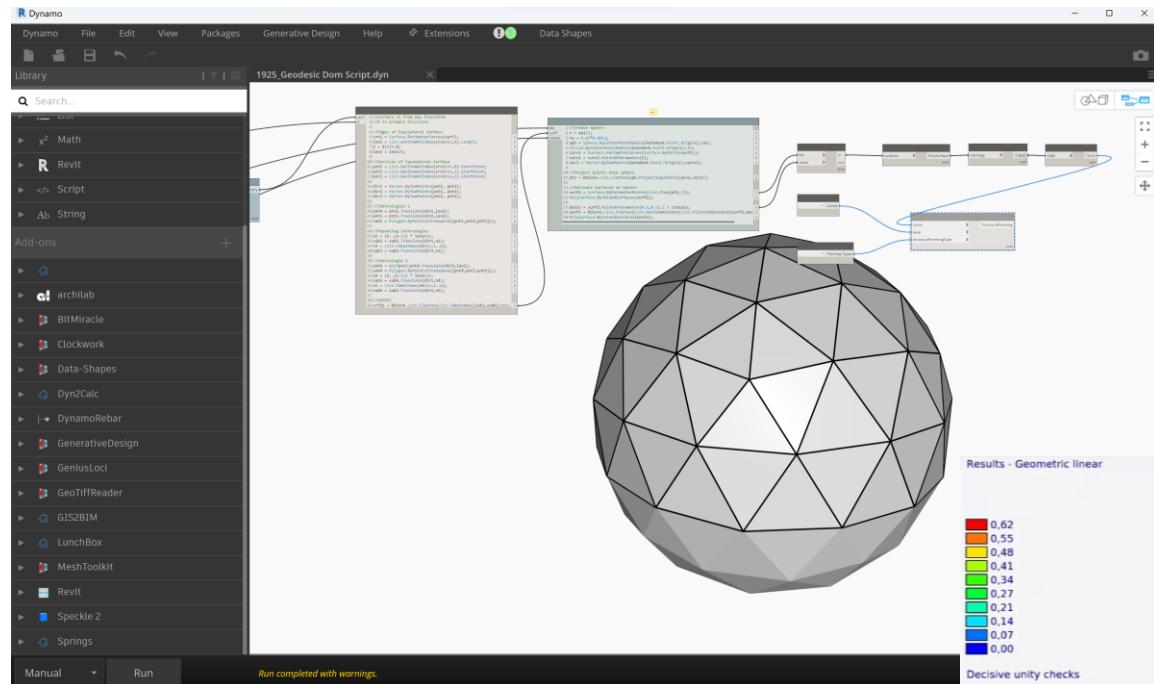
```
27 BeamsLayer = root.findall("./Beams/Layer")
28 BeamsRotation = root.findall("./Beams/Angle")
29
30 #PROFILES
31 ProfileNumber = root.findall("./Profiles/Number")
32 ProfileName = root.findall("./Profiles/Profile_name")
33
34 #BEAMS
35 for i, j, k, l, m in zip(BeamsFrom, BeamsTo, BeamsName, BeamsNumber, BeamsRotation):
36     profile_name = ProfileName[int(k.text)-1].text
37     #profile_name = profile_name.split()[0]
38     if profile_name == None:
39         print("profile_name is None")
40     else:
41         start = XYZ[1][XYZ[0].index(i.text)]
42         end = XYZ[1][XYZ[0].index(j.text)]
43         try:
44             obj.append(Frame.byStartpointEndpointProfileNameShapevector(start, end, profile_name, profile_name + "-" + l.text, Vector2(0,0,0), float(m.text)))
45         except:
46             print("could not translate " + profile_name)
47
48 SpeckleObj = translateObjectsToSpeckleObjects(obj)
49
50 Commit = TransportToSpeckle("struct4u.xyz", "3fff8f56cc", SpeckleObj, "Industrial Steel Structure 2")
```

8. XFEM4Unaar Speckle



[Commit | Speckle \(struct4u.xyz\)](https://struct4u.xyz)

9 Geodetische Koepel



 Struct4U

All together in control

9 Hyparschaal

XFrame3d - 5.04.02 3BM Bouwtechniek

Projectnummer: 1421
Projectomschrijving: Nieuwbouw Woonhuis
Datum - tijd: 27-02-2021 - 19:34

Bestand Weergeven Achtergrond

Verhoogde weergave

Formule berekening:

$$I_{xx} = 0.9 \times 1 = 0.9 \times 996 = 896 \text{ mm}^4$$

$$\sigma_{max} = \frac{0.78 \cdot b^2}{I_{xx}} \cdot E \cdot \alpha = \frac{0.78 \cdot 96^2}{896} \times 7400 = 302.2 \text{ N/mm}^2 \quad (6.32)$$

$$\lambda_{max} = \sqrt{\frac{f_{ck}}{n \cdot \alpha}} = \sqrt{\frac{24}{0.282 \times 0.75}} = 0.282 < 0.75 \rightarrow k_{ax} = 1.00 \quad (6.30)(6.34)$$

$$M_{xx} = 186 \times 10^6 \quad M_{xx} = \frac{186 \times 10^6}{675 \times 10^3} = 0.3 \text{ N/mm}^2 \quad \sigma_{ax} = \frac{N_{ax}}{A_{xx}} = \frac{405}{186} = 0.0 \text{ N/mm}^2$$

$$\lambda_{yy} = \frac{L_{yy} \cdot I_{yy}}{W_y} = \frac{996}{675 \times 10^3} = 0.277 \quad \lambda_{yy} = 0.277 < 0.75 \quad \lambda_{yy} = \sqrt{\frac{f_{ck}}{E_{yy}}} = \sqrt{\frac{24}{7400}} = 0.611 \quad (6.22)$$

$$k_{yy} = 0.541 + \beta \cdot (\lambda_{yy} - 0.3) + \lambda_{yy}^2 = 0.5 \times (1 + 0.2 \times (0.611 - 0.3) + 0.611^2 = 0.72 \quad (6.28)$$

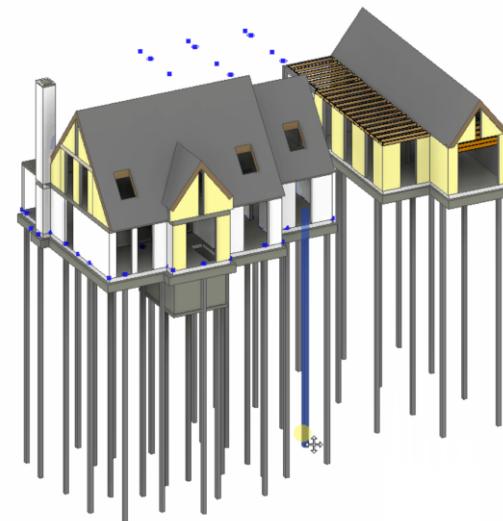
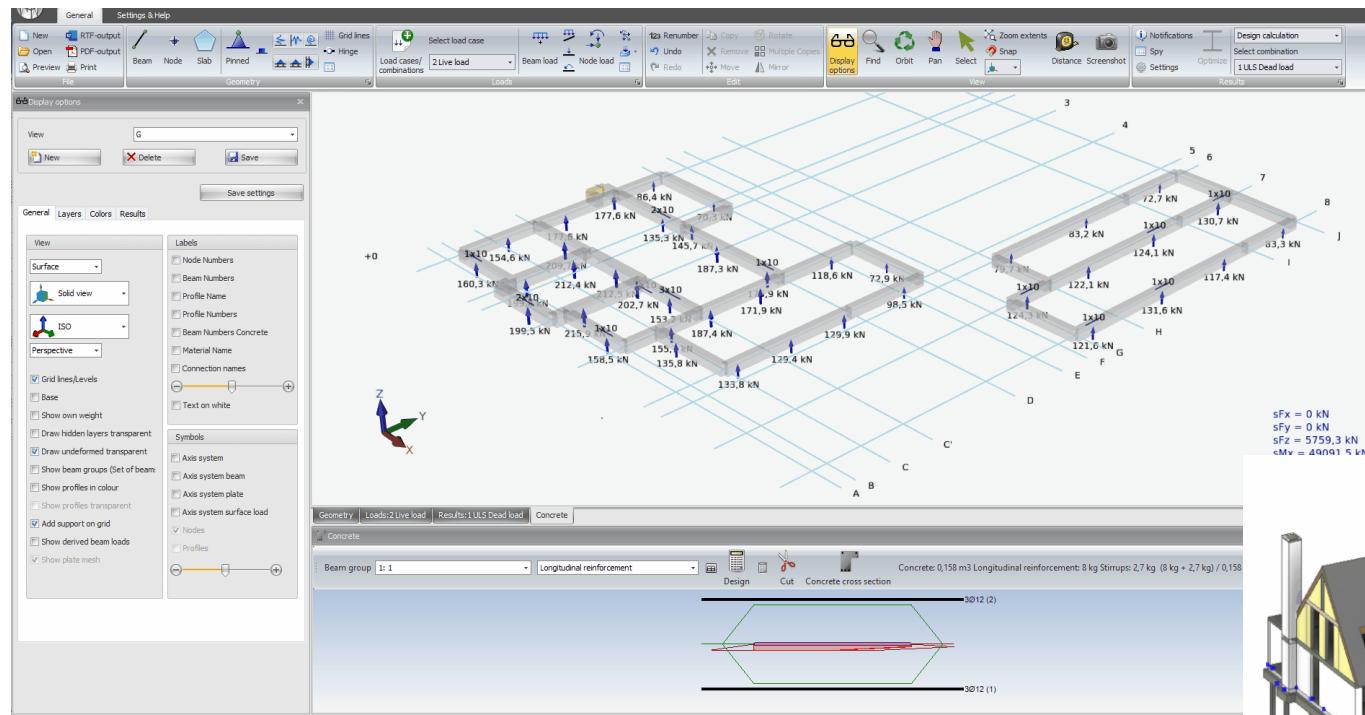
$$k_{xy} = \frac{1}{k_x \cdot \sqrt{k_x^2 + k_y^2}} = \frac{1}{0.72 \cdot \sqrt{0.72^2 + 0.611^2}} = 0.91 \quad (6.26)$$

$$\left(\frac{\sigma_{ax}}{k_{ax} f_{ck}} \right)^2 + \left(\frac{\sigma_{ay}}{k_{ay} f_{ck}} \right)^2 + \left(\frac{\sigma_{xy}}{k_{xy} f_{ck}} \right)^2 = 0.00 < 1.00 \quad (6.35)$$

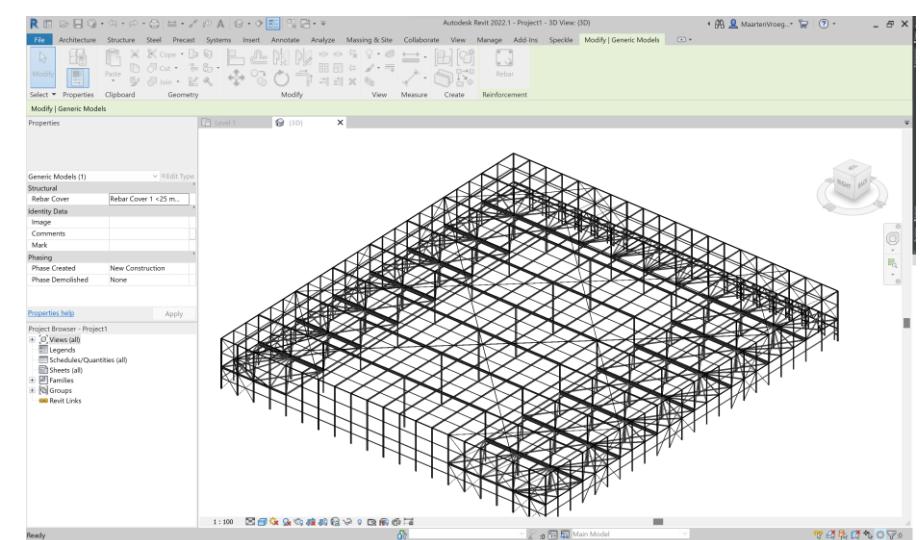
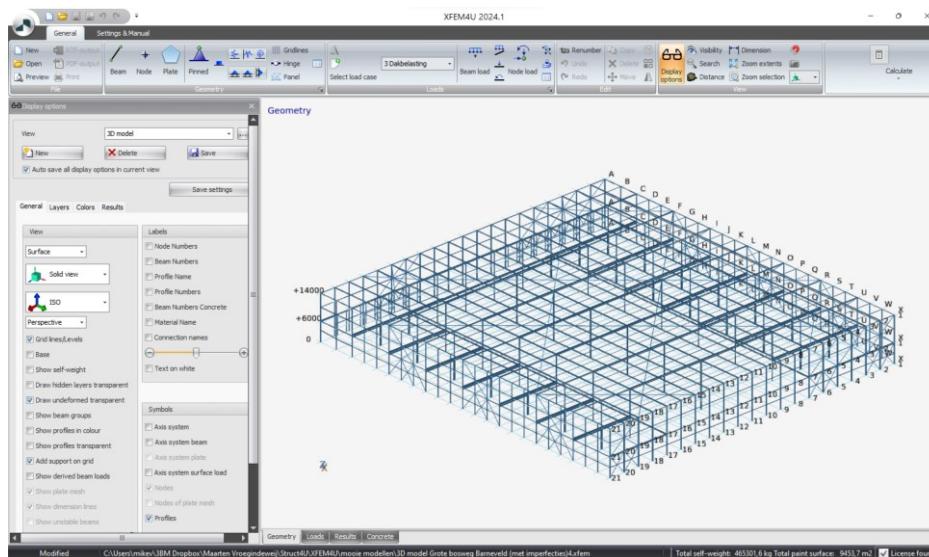
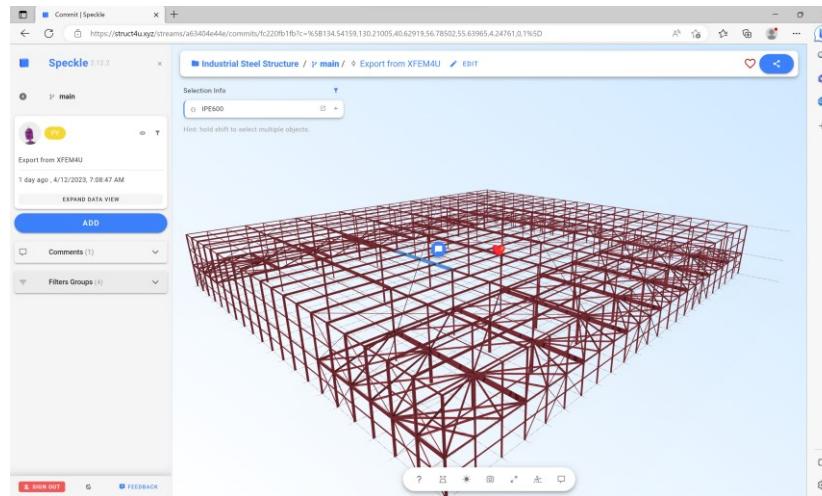
Doorbijgenging:

Combinatie: 4 x 554.4 mm $N = 0.046 \text{ kN}$ $V_y = 0.168 \text{ kN}$ $V_z = 2.696 \text{ kN}$

10 Paalposities vanuit XEM4 Unaar Revit



11 Samenwerking met denden



12 Download

The screenshot shows a web browser window with two tabs open, both titled "Commit | Speckle". The active tab displays the GitHub repository page for "DutchSailor/Struct4U". The repository is public and has 1 branch and 0 tags. The main content area lists 12 commits from "DutchSailor" in reverse chronological order, all made "now" (726586a) and each consisting of 6 commits. The commits are categorized under "Code" and include:

- 2 Castellated Beam
- 3 Concrete Basement
- 4 Concrete Floor
- 5 Stair Beams
- 6 Stair Plates
- 7 Industrial Hall
- 8 XFEM4U to Speckle
- 9 Geodesic Dome
- Struct4U
- .gitignore

The right sidebar contains sections for "About", "Releases", and "Packages".

About

Struct4U Python Scripts

- Readme
- GPL-3.0 license
- 0 stars
- 1 watching
- 0 forks

Releases

No releases published

[Create a new release](#)

Packages

No packages published

[Publish your first package](#)

<https://github.com/DutchSailor/Struct4U>

0.1

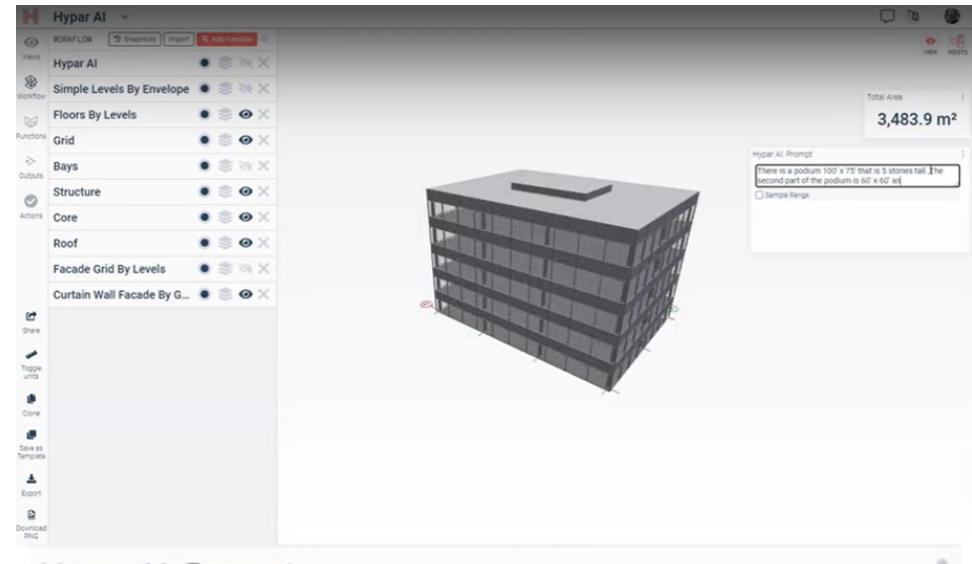


All together in control

2023

- Online generator
- API in XFEM4U inclusief resultaten
- Directe Speckle export vanuit XFEM4U
- Excel voorbeelden van parametrische constructies
- API verder uitbreiden met:
 - Verbindingen
 - Ruimtelijke vakwerkliggers

GPT → Generative Pre-trained Transformer(GPT)



Hyper AI: Prompt

There is a podium 100' x 75' that is 5 stories tall. The second part of the podium is 60' x 60' an|