

# Automatic creation of navigation network from IFC models

Progress: 2 Feb , 2016

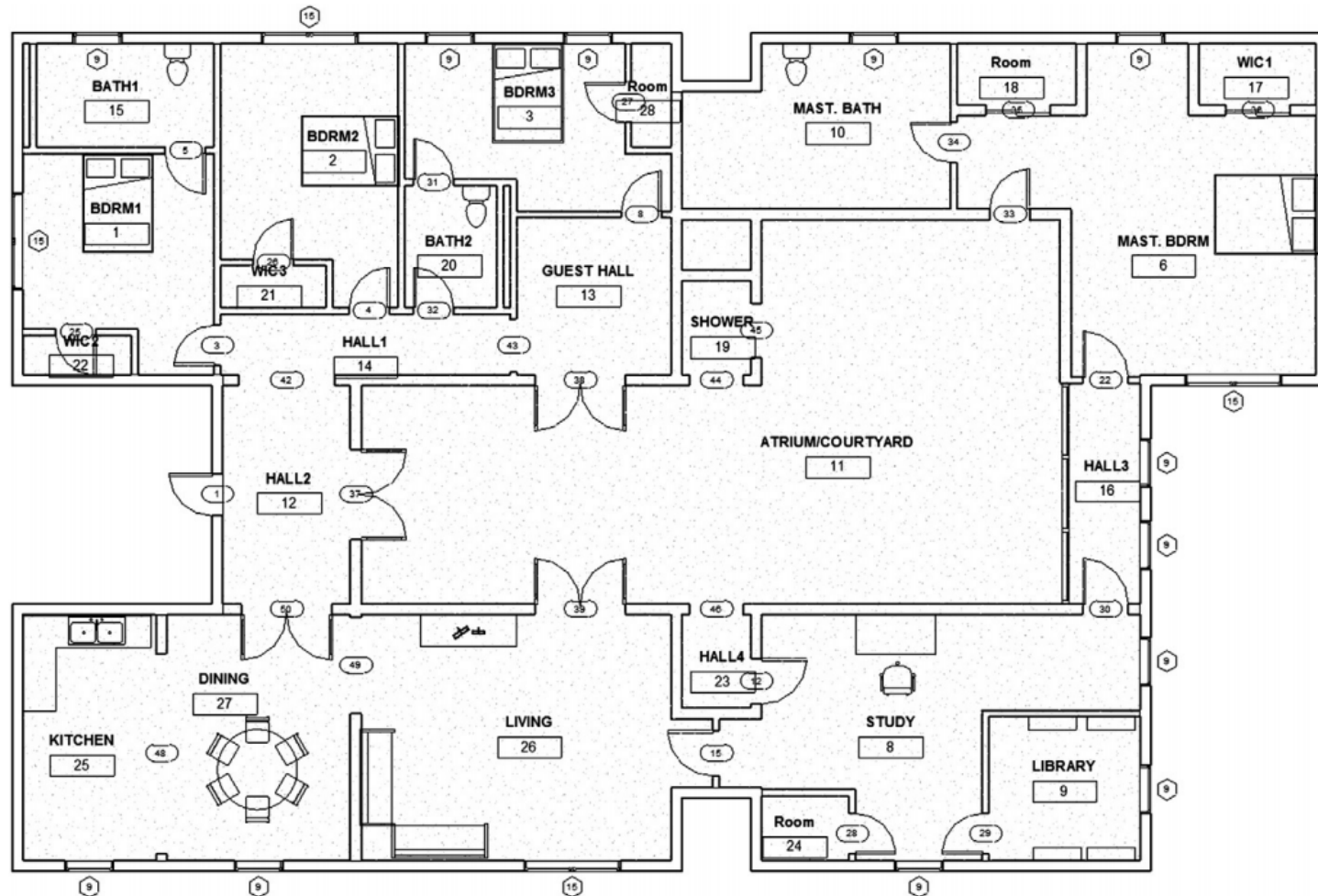
**Shuangfeng Wei** Dr. Eng.  
3D Group,BK

# Content

- **1. Related work**
- **2. Goal of the research**
- **3. Steps to be completed**
- **4. Progress**

# 1.Related work(1)

use BIM's floor plan with **Revit API**(Wei Yan,2010)



# 1.Related work(2)

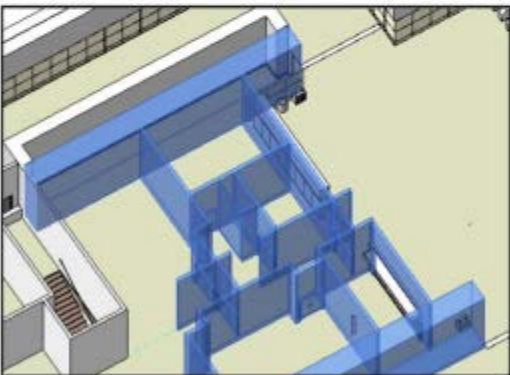
C# + Revit API + Matlab for 2D path-finding with A-MAT-VG  
(Albert Y. Chen and Ting Huang, Oct, 2015)

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using Autodesk.Revit.UI;
using Autodesk.Revit.DB;
namespace revit_api_test1
{
    [Autodesk.Revit.Attributes.Transaction(Autodesk.Revit.Attributes.TransactionMode.Manual)]
    [Autodesk.Revit.Attributes.NoRegeneration(Autodesk.Revit.Attributes.NoRegenerationMode.NoRegeneration)]
    public class Class1 : IExternalCommand
    {
        public Result Execute(ExternalCommandData commandData, ref string
        {
            UIDocument uidoc = commandData.Application.ActiveUIDocument;
            Autodesk.Revit.UI.Selection.SelectionSet selElements =
            uidoc.Selection.Elements;
            //bool checker = true;
            int i = 0;
            StringBuilder st = new StringBuilder();
            StringBuilder st1 = new StringBuilder();
        }
    }
}
```

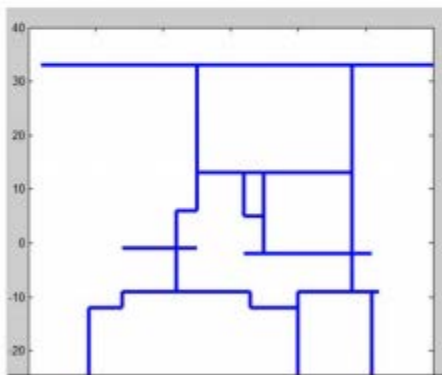
1. C# Revit API

```
-20.3722704002289, 33.5831797279153
35.7300918044956, 33.5831797279152
35.7300918044956, 33.5831797279152
35.7300918044949, -29.4089462563368
35.7300918044949, -29.4089462563368
-20.372270400229, -29.4089462563362
-20.372270400229, -29.4089462563361
-20.3722704002289, 33.5831797279154
26.5437400984588, 33.5831797279152
26.5437400984588, 8.97688051531678
26.5437400984588, 8.97688051531678
35.7300918044954, 8.97688051531675
-20.3722704002289, 11.2734684418261
8.17103668638531, 11.273468441826
8.1710366863853, 11.273468441826
8.1710366863853, 11.273468441826
```

3. Attributes Output



2. BIM Object Selection



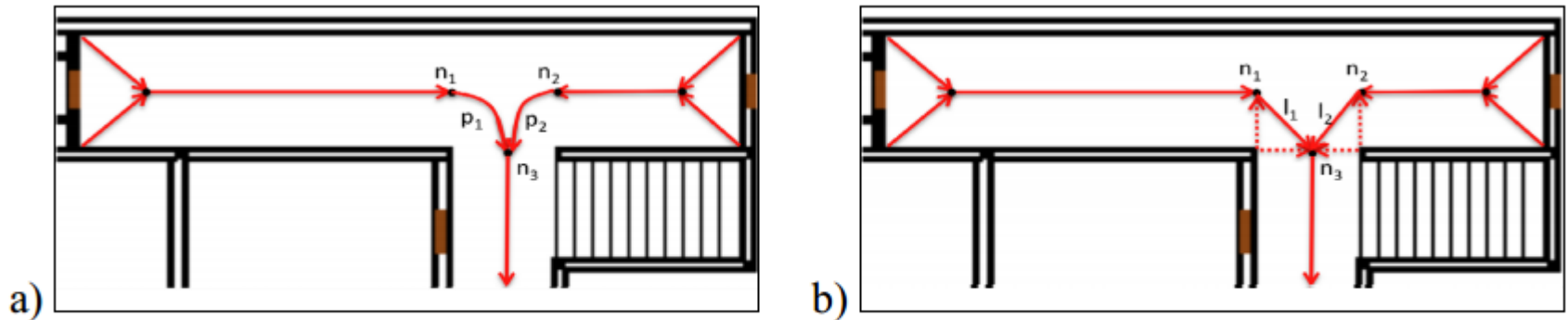
4. Layout in Matlab



# 1.Related work(3)

Transform IFC to GTN(Geometric Topology Network) in 2D  
path-finding with S-MAT

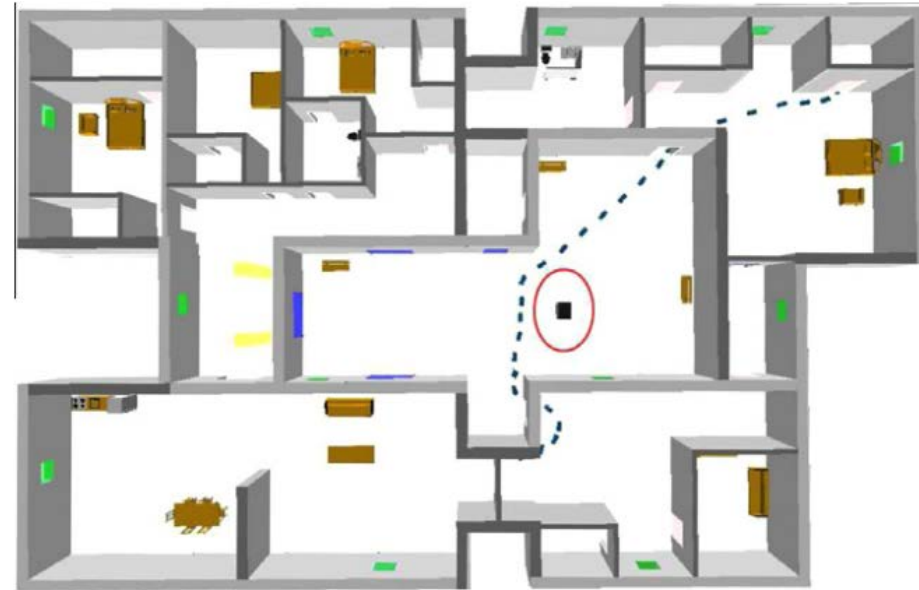
(S. Taneja,B. Akinci, J.H. Garrett,2011)



**Figure 6. a) Voronoi diagram of the polygon, b) Modified medial axis of the polygon.**

# 1.Related work(4)

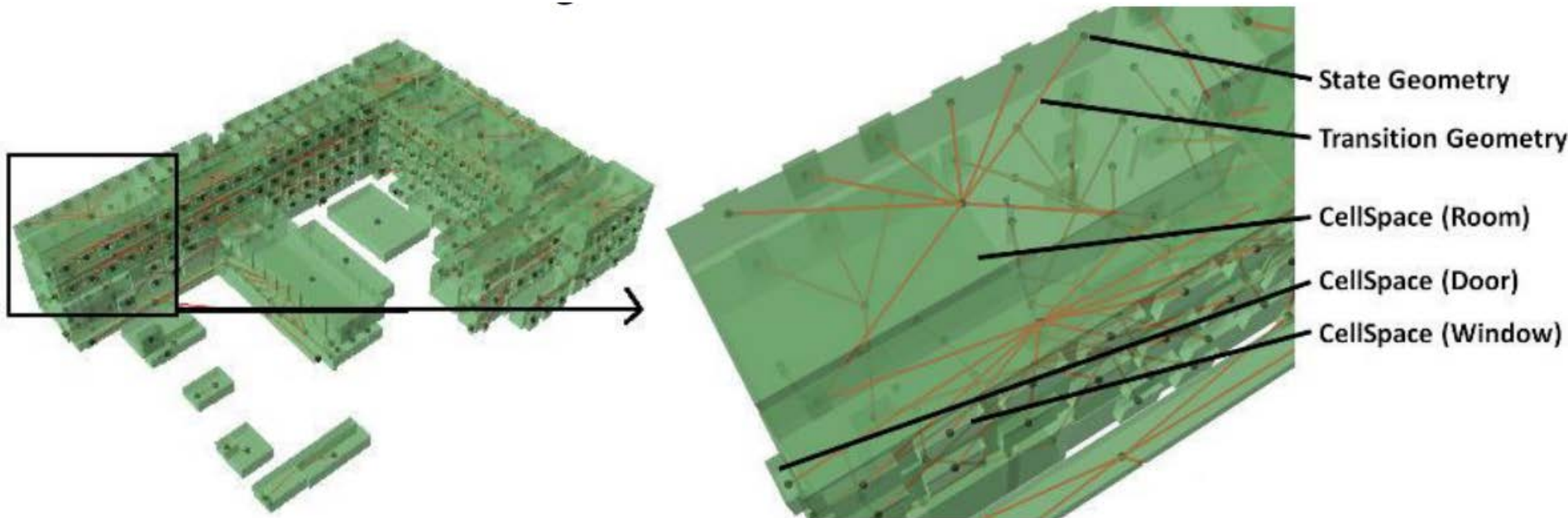
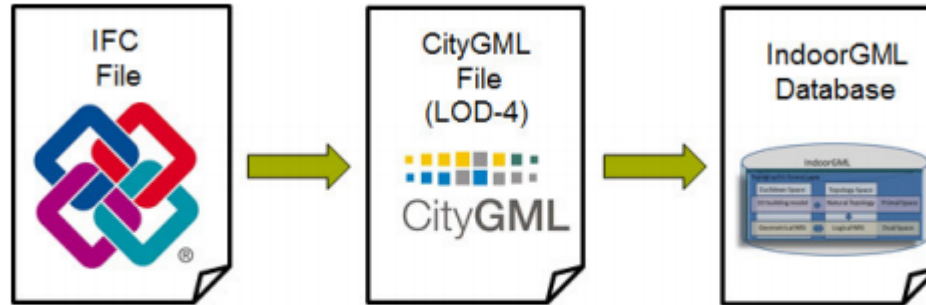
- a) extracting both geometric and semantic information
- b) discretizing and mapping the extracted information into a **planar grid**,
- c) path-finding  
(Ya-Hong Lin etc.,2012)





# 1.Related work(5)

(Aftab A. K. etc.Andreas D., Thomas H. K., Jun.2014)



# 1.Related work(6)

(Aftab A. K. etc. Zhihang Y., Thomas H. K.,Nov.2014)

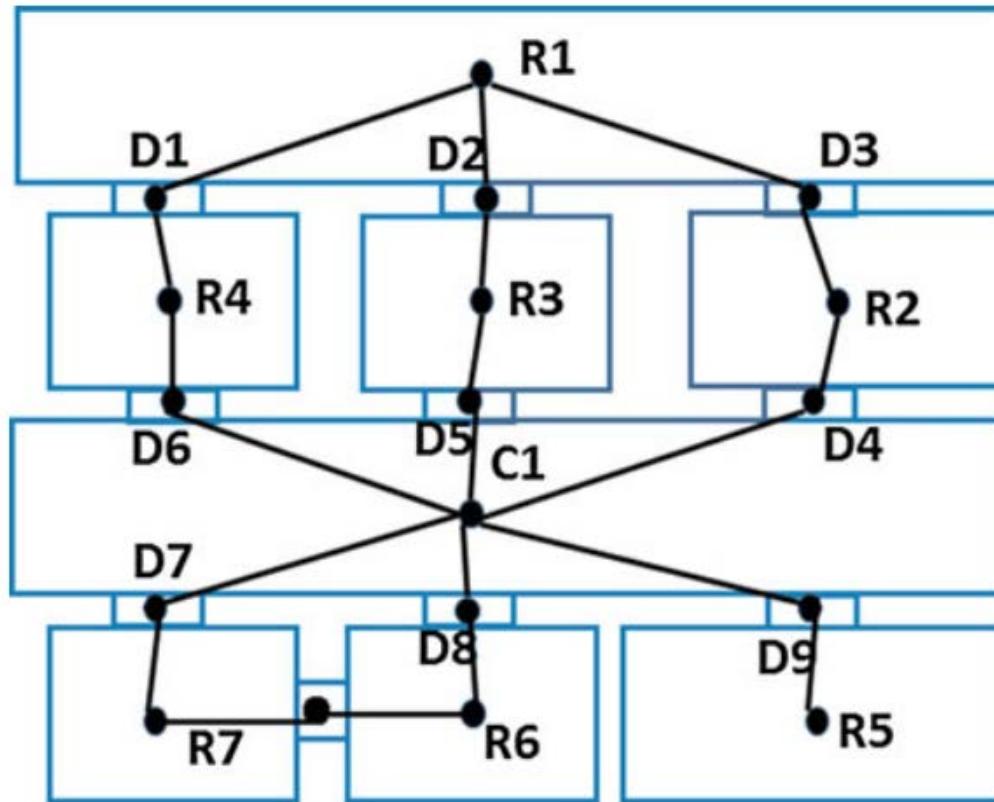


**Main 3D building model of TUM**



# 1.Related work(6)

(Aftab A. K. etc. Zhihang Y., Thomas H. K.,Nov.2014)



Room-door-room-door Network

# 1.Related work(6)

(Aftab A. K. etc. Zhihang Y., Thomas H. K.,Nov.2014)

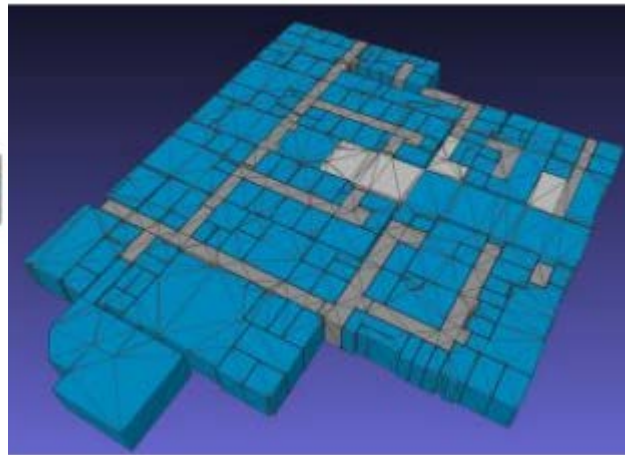
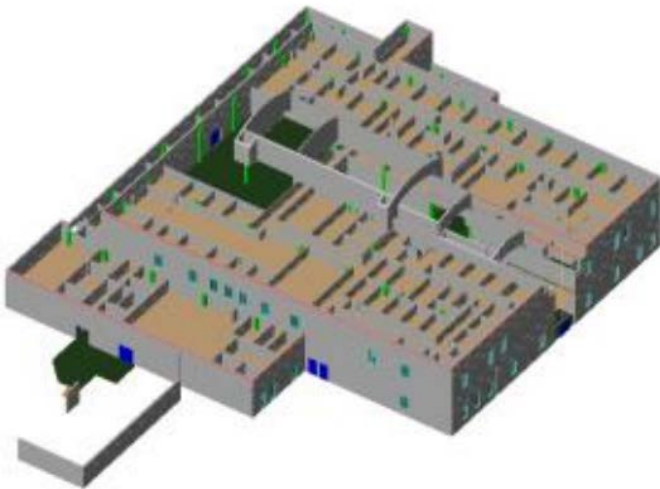


Network model of TUM main building

# 1.Related work(7)

multi-step indoor navigation network deviation process from IFC:

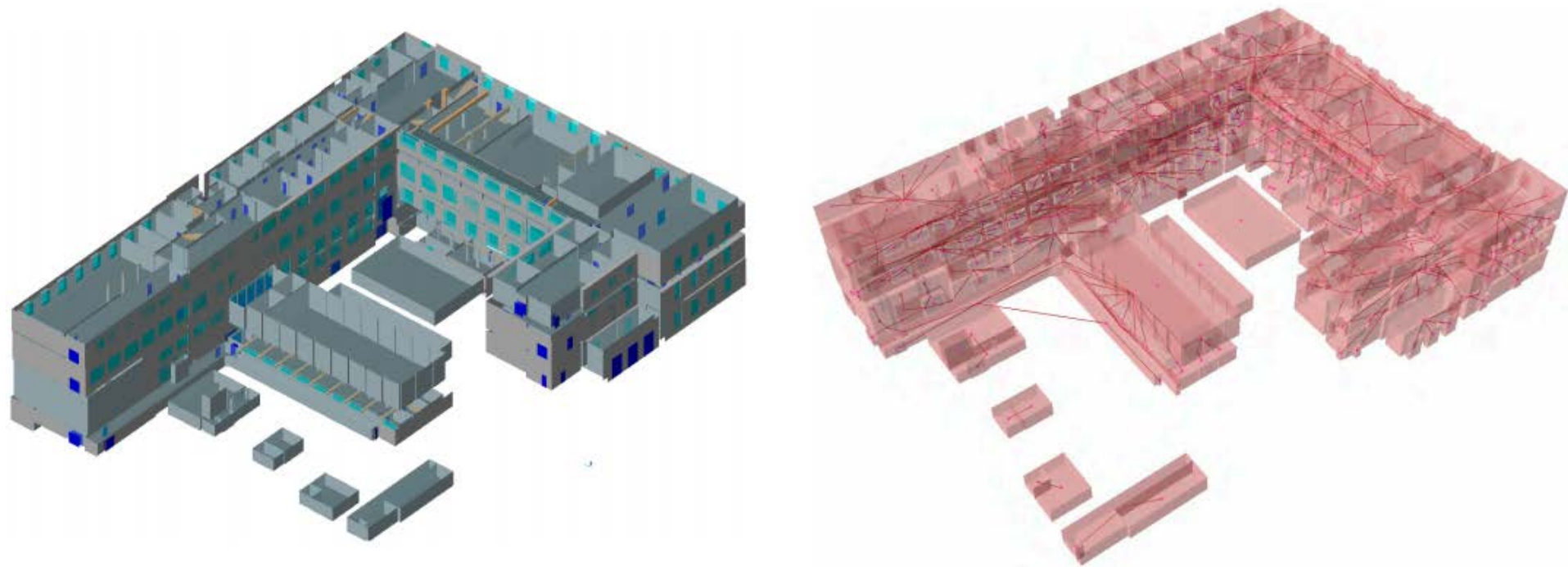
- a) semantic cleaning;
  - b) walkable features extraction;
  - c) Multi-Storey **2D Mapping**;
  - d) automatically generate network with S-MAT.
- (S. J. Tang, Q. Zhu, W.W.WANG etc.2015)



# Question

Above mentioned method get indoor navigation network **indirectly** from IFC by transforming to CityGML or 2D mapping.

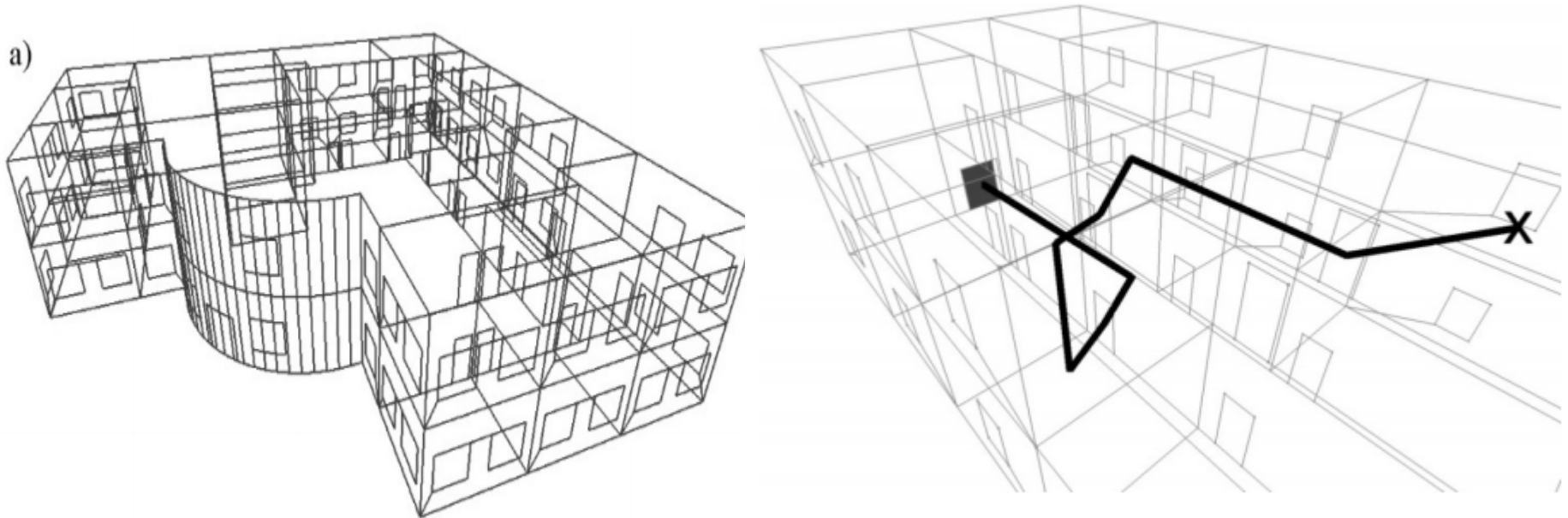
**Why not create automatically it from IFC directly?**





# A little simple example:

**not considering important useful IfcObjects such as stairs, elevator, obstacles (IfcFurnishingElement, IFurniture) and other IfcObjects.**



(P. Boguslawski, L. Mahdjoubi, V. Zverovich., Sept.2015)



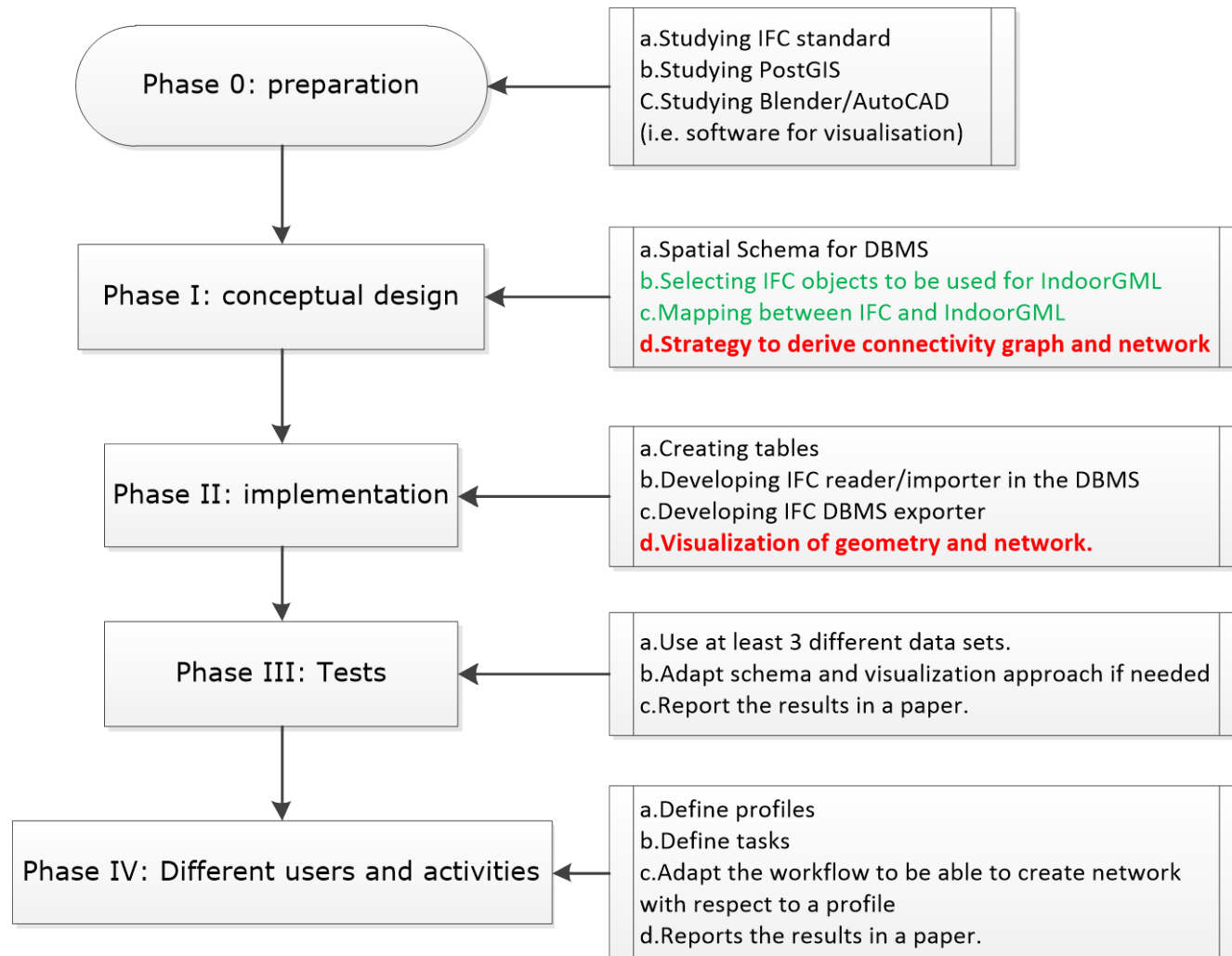
## 2.Goal of the research

This research will concentrate on an approach for 3D indoor navigation considering obstacles. Two major problems will be addressed:

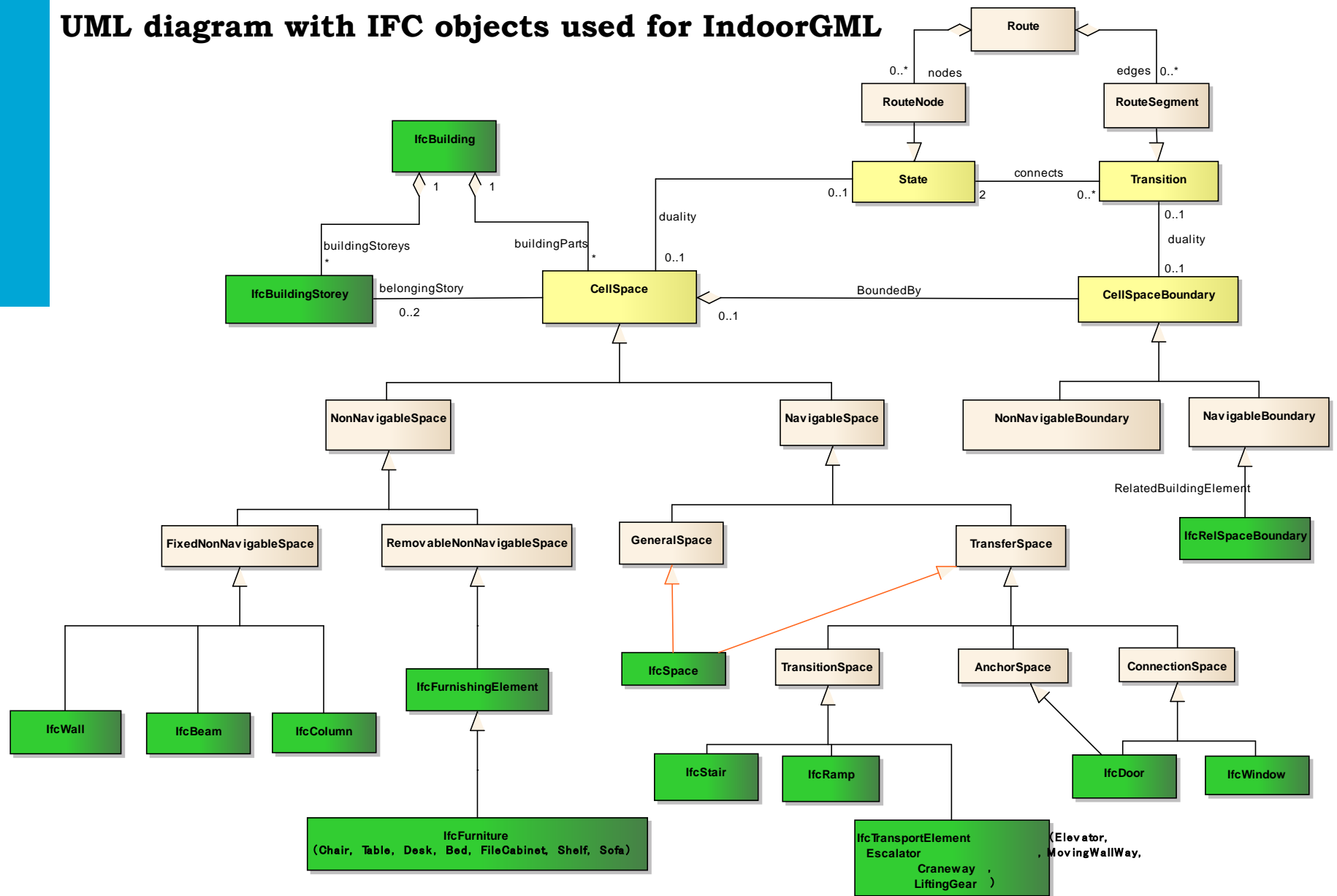
- 1) creating a spatial schema for IndoorGML (introducing adaptations for obstacles if needed)
- 2) automatic generation of this model from 3D semantic-rich models Building Information models (i.e. IFC).

The goal of the research is to **investigate automatic procedures for network derivation** for different users and their activities.

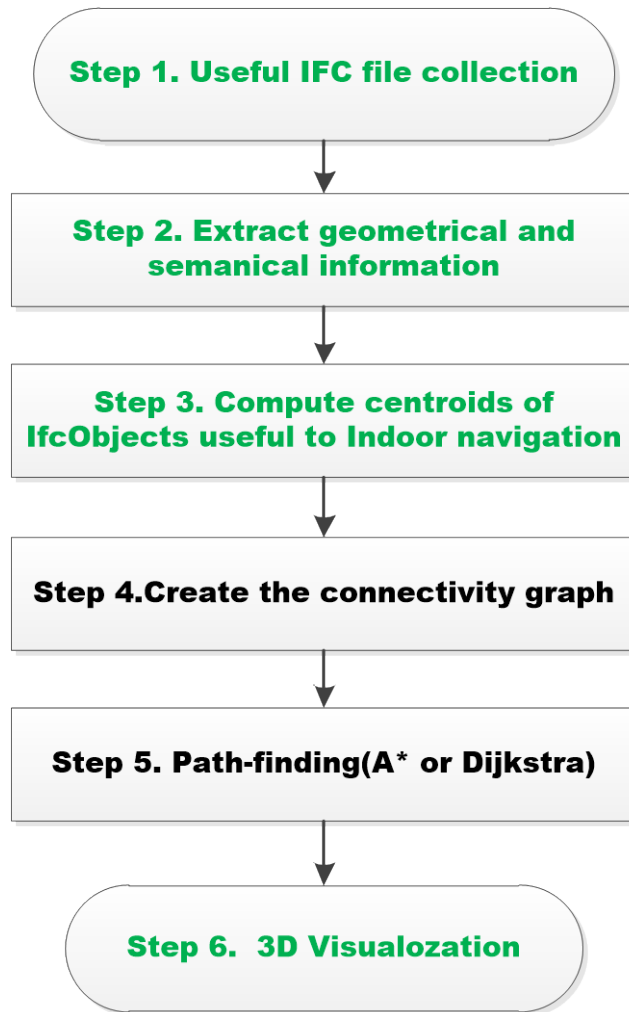
# Work Plan(Jun,2015-Jun,2016)

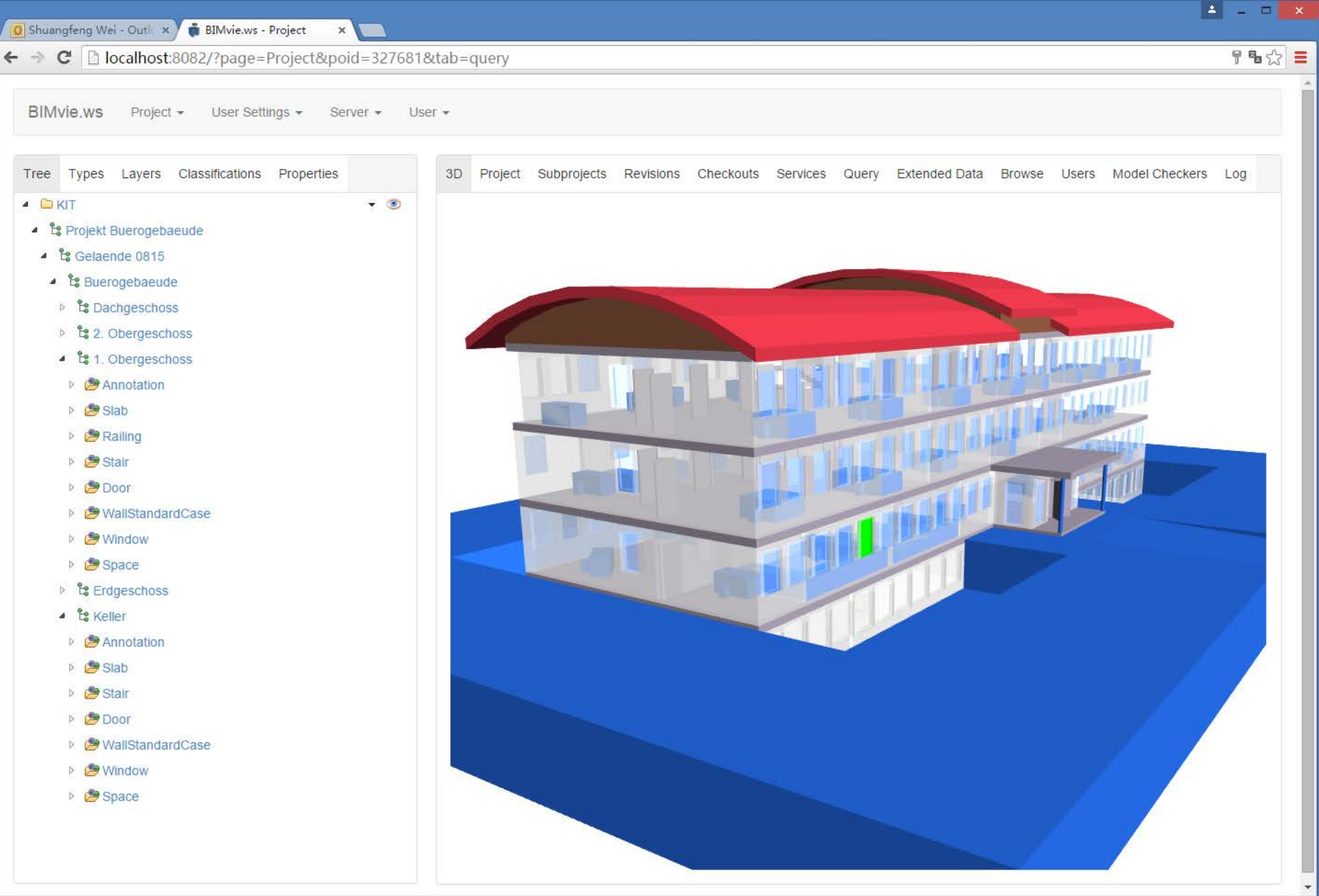


# UML diagram with IFC objects used for IndoorGML



# Strategy to derive connectivity graph







Shuangfeng Wei - Outlook x BIMvie.ws - Project x

localhost:8082/?page=Project&poid=327681&tab=query

BIMvie.ws Project User Settings Server User

Tree Types Layers Classifications Properties

- KIT
  - Projekt Buerogebaeude
    - Gelaende 0815
      - Buerogebaeude
        - Dachgeschoss
        - 2. Obergeschoss
        - 1. Obergeschoss
          - Annotation
          - Slab
          - Railing
          - Stair
          - Door
          - WallStandardCase
          - Window
          - Space
        - Erdgeschoss
        - Keller
          - Annotation
          - Slab
          - Stair
          - Door
          - WallStandardCase
          - Window
          - Space

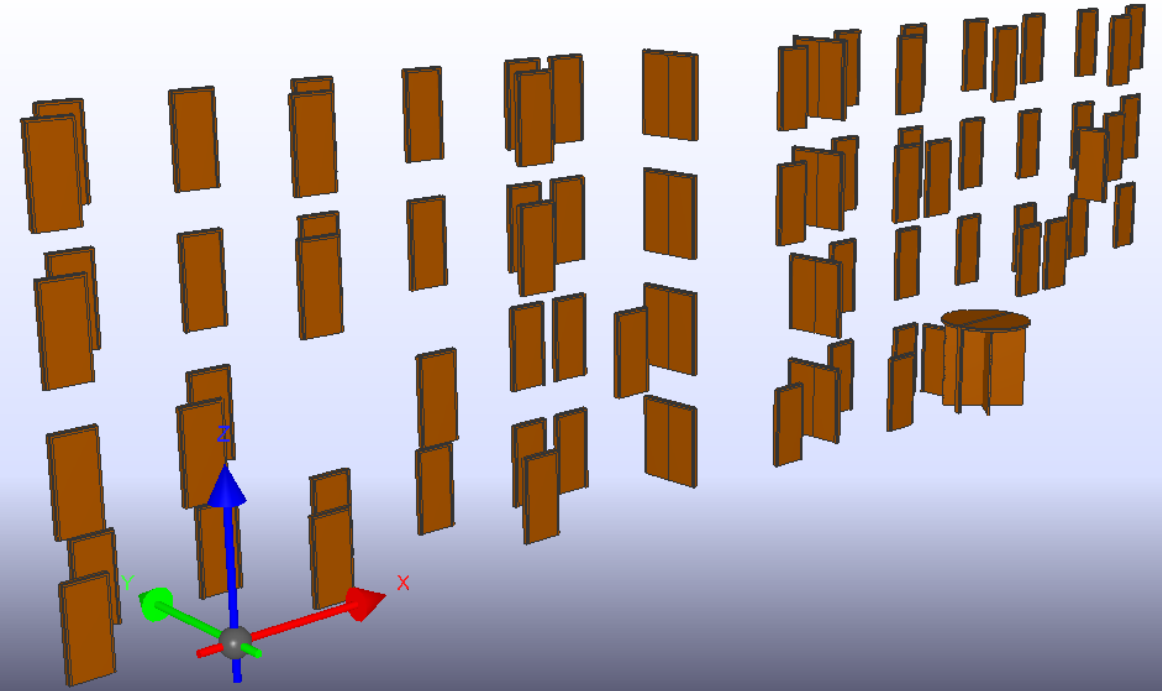
3D Project Subprojects Revisions Checkouts Services Query Extended Data Browse Users Model Checkers Log

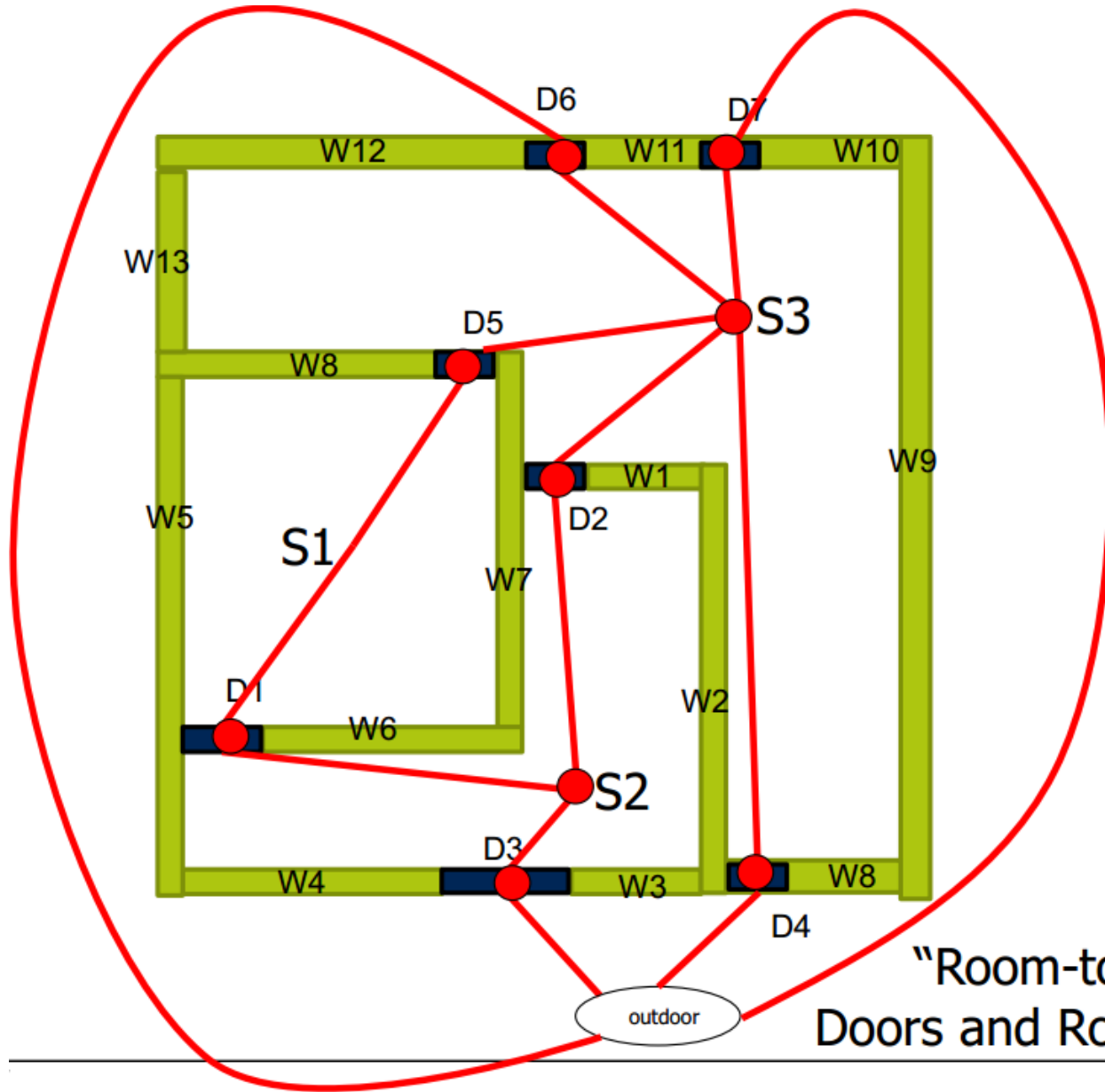
Querying Revision 1 (Last)

Query engine BimQL Engine Load example

Select \$Var1  
Where \$Var1.EntityType = IfcDoor

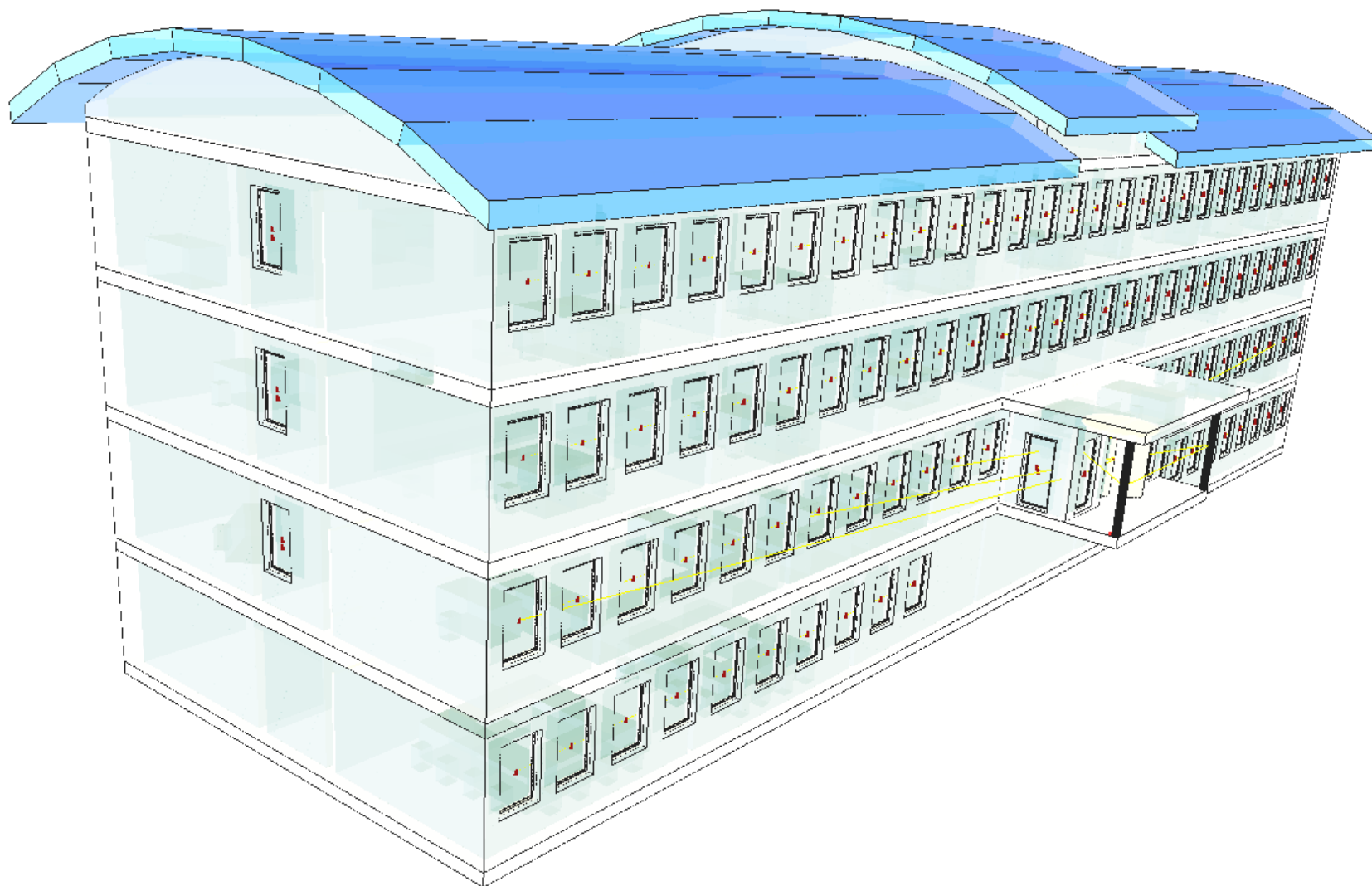
Return all rooted entities  
Return all rooted entities of type IfcDoor  
Return all rooted entities of which the OverallHeight attribute equals 2.325

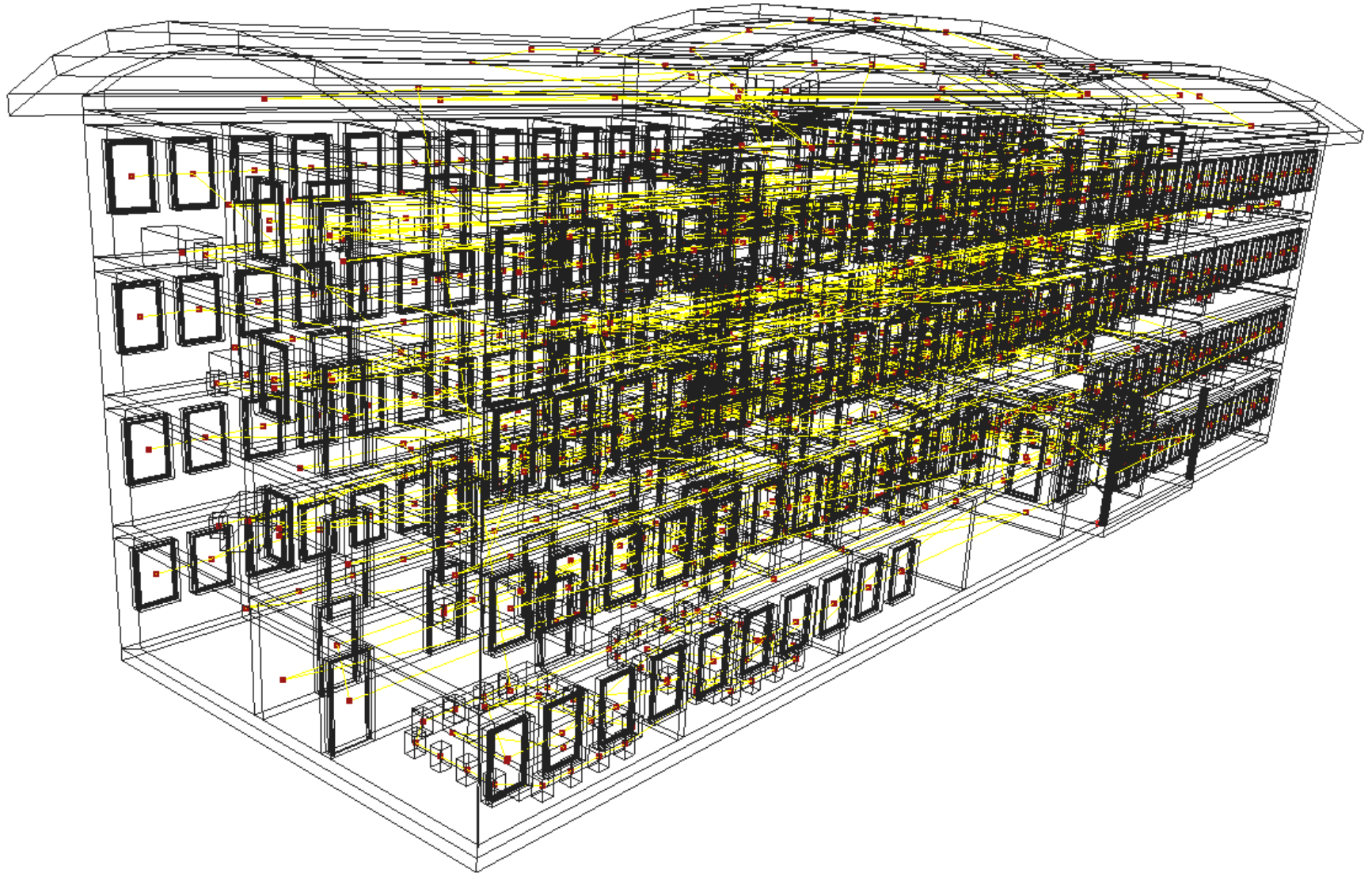




Use only rooms  
Use only doors  
Use both

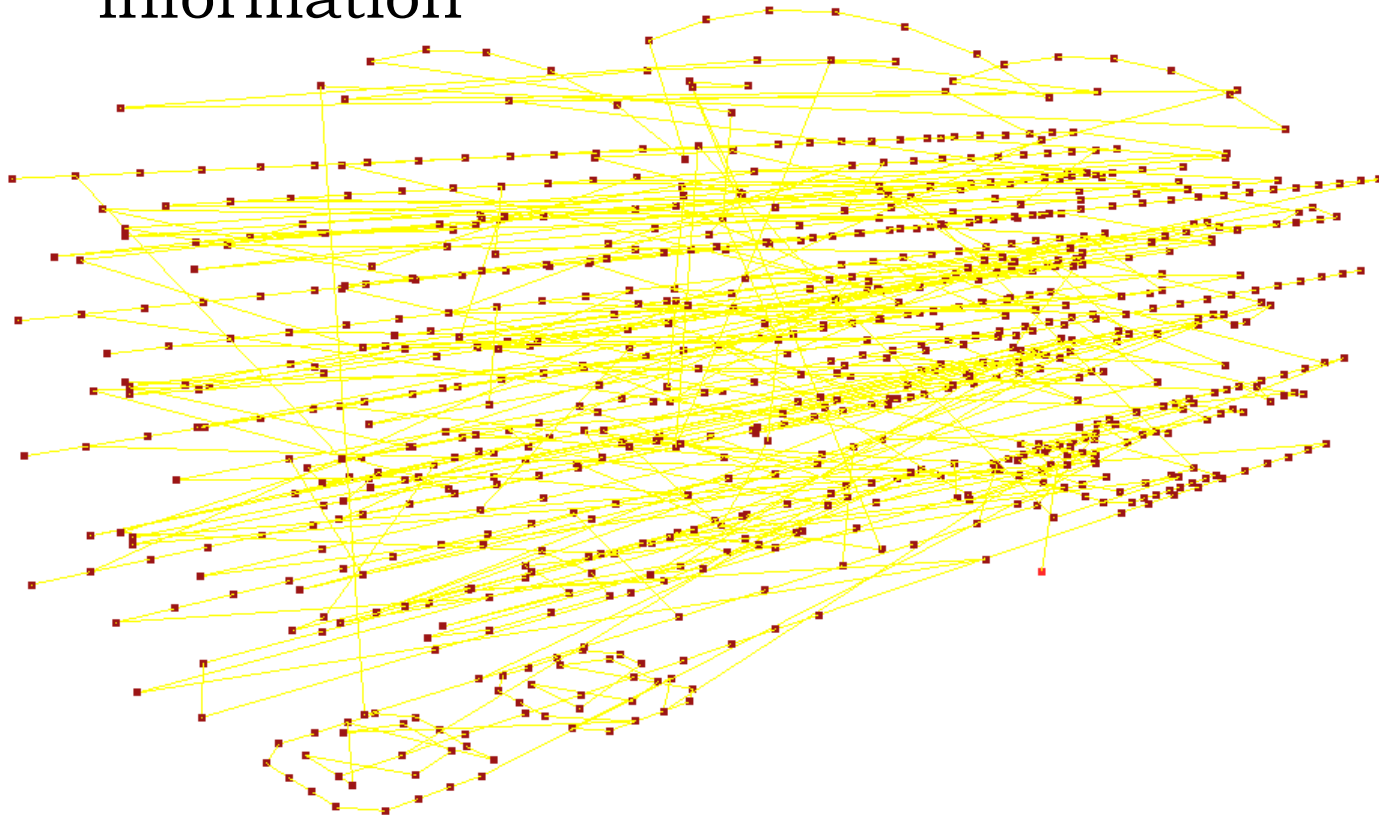
“Room-to-door-to-room”  
Doors and Rooms are nodes





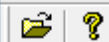
# Connectivity graph and network

But it's incorrect or ideal! Since the lines is only connected naturally with retrieved Centroids not using semantical information





File Edit View ?

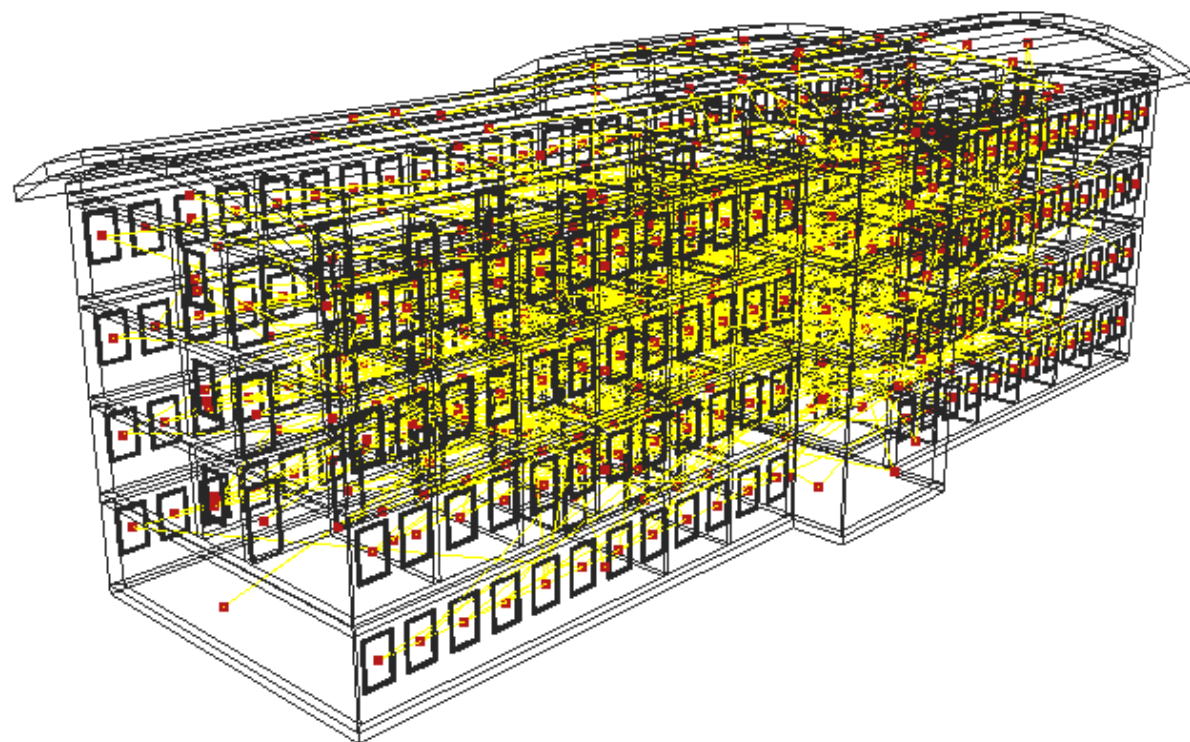


```
#268704895= IFCCARTESIANPOINT((1.45,2.38));
#268704900= IFCCARTESIANPOINT((1.45,2.395));
#268704904= IFCPOLYLINE((#268704883,#268704887,#268704891,#268704895,#268704900));
#268704908= IFCCOMPOSITECURVESEGMENT(.CONTINUOUS,.,F.,#268704904);
#268704912= IFC2DCOMPOSITECURVE((#268704908),.U.);
#268704916= IFCCURVEBOUNDEDPLANE(#268704880,#268704912,());
#268704920= IFCCONNECTIONSURFACEGEOMETRY(#268704916,$);
#268704921= IFCRELSPACEBOUNDARY('13577A6gD2HAV_tMdUcpzv',#268437024,'Second Level',,$,#268555768,#268616405,#268704920,.,PHYSICAL,.,INTERNAL.);
#268704922= IFCDIRECTION((0,.-1,0.));
#268704926= IFCDIRECTION((-1,0,0.));
#268704930= IFCAXIS2PLACEMENT3D(#268437051,#268704926,#268704922);
#268704933= IFCPLANE(#268704930);
#268704936= IFCCARTESIANPOINT((1.15,2.135));
#268704940= IFCCARTESIANPOINT((0.265,2.135));
#268704944= IFCCARTESIANPOINT((0.265,0.));
#268704948= IFCCARTESIANPOINT((1.15,0.));
#268704952= IFCCARTESIANPOINT((1.15,2.135));
#268704956= IFCPOLYLINE((#268704936,#268704940,#268704944,#268704948,#268704952));
#268704960= IFCCOMPOSITECURVESEGMENT(.CONTINUOUS,.,F.,#268704956);
#268704964= IFC2DCOMPOSITECURVE((#268704960),.U.);
#268704968= IFCCURVEBOUNDEDPLANE(#268704933,#268704964,());
#268704972= IFCCONNECTIONSURFACEGEOMETRY(#268704968,$);
#268704973= IFCRELSPACEBOUNDARY('3sq_6W4xtEwBmqwX8Slu3M',#268437024,'Second Level',,$,#268555768,#268503479,#268704972,.,PHYSICAL,.,INTERNAL.);
#268437024= IFCOWNERHISTORY(#268437023,#268437016,$,.,NOCHANGE,,$,$,1138628715);
#268555768= IFCSPACE('3QWivMZ9nC0PvQnq6tDx7e',#268437024,'K-1',,$,$,#268555765,#268555754,'Windfang',.ELEMENT,.,INTERNAL,,$);
#268503479= IFCDOOR('0vFlS6Wth9Xf6tyOfY88Ko',#268437024,'Door-045',,$,$,#268496047,#268503475,$,2.135,0.885);
#268704972= IFCCONNECTIONSURFACEGEOMETRY(#268704968,$);
#268704974= IFCDIRECTION((-1,0,0.));
#268704978= IFCCARTESIANPOINT((1.45,0,0.));
#268704982= IFCAXIS2PLACEMENT3D(#268704978,#268437043,#268704974);
#268704985= IFCPLANE(#268704982);
#268704988= IFCCARTESIANPOINT((0.315,2.055));
#268704992= IFCCARTESIANPOINT((1.2,2.055));
#268704996= IFCCARTESIANPOINT((1.2,1.255));
#268705000= IFCCARTESIANPOINT((0.315,1.255));
```

## Header Info

☒ 'Projekt Buerogebaeude', 'No real Project'☒ Not Referenced0000000

☒ 'Tuer-016' = '<description>' (IfcDoor)  
☒ 'Tuer-004' = '<description>' (IfcDoor)  
☒ 'Tuer-002' = '<description>' (IfcDoor)  
☒ 'Tuer-004' = '<description>' (IfcDoor)  
☒ 'Tuer-005' = '<description>' (IfcDoor)  
☒ 'Tuer-005' = '<description>' (IfcDoor)  
☒ 'Tuer-002' = '<description>' (IfcDoor)  
☒ 'Tuer-009' = '<description>' (IfcDoor)  
☒ 'Tuer-013' = '<description>' (IfcDoor)  
☒ 'Tuer-016' = '<description>' (IfcDoor)  
☒ 'Tuer-018' = '<description>' (IfcDoor)  
☒ 'Tuer-010' = '<description>' (IfcDoor)  
☒ 'Tuer-005' = '<description>' (IfcDoor)  
☒ 'Tuer-015' = '<description>' (IfcDoor)  
☒ 'Tuer-017' = '<description>' (IfcDoor)  
☒ 'Tuer-009' = '<description>' (IfcDoor)  
☒ 'Tuer-010' = '<description>' (IfcDoor)  
☒ 'Tuer-008' = '<description>' (IfcDoor)  
☒ 'Tuer-003' = '<description>' (IfcDoor)  
☒ 'Tuer-005' = '<description>' (IfcDoor)  
☒ 'Tuer-014' = '<description>' (IfcDoor)  
☒ 'Tuer-017' = '<description>' (IfcDoor)  
☒ 'Tuer-008' = '<description>' (IfcDoor)  
☒ 'Tuer-018' = '<description>' (IfcDoor)  
☒ 'Tuer-004' = '<description>' (IfcDoor)  
☒ 'Tuer-016' = '<description>' (IfcDoor)  
☒ 'Tuer-019' = '<description>' (IfcDoor)  
☒ 'Tuer-016' = '<description>' (IfcDoor)  
☒ 'Tuer-012' = '<description>' (IfcDoor)  
☒ 'Tuer-005' = '<description>' (IfcDoor)  
☒ 'Tuer-006' = '<description>' (IfcDoor)  
☒ 'Tuer-005' = '<description>' (IfcDoor)  
☒ 'Tuer-016' = '<description>' (IfcDoor)  
☒ 'Tuer-005' = '<description>' (IfcDoor)  
☒ 'Tuer-015' = '<description>' (IfcDoor)  
☒ 'Tuer-012' = '<description>' (IfcDoor)  
☒ 'Tuer-010' = '<description>' (IfcDoor)  
☒ 'Tuer-012' = '<description>' (IfcDoor)



But other important IfcObjects  
having relationships with  
IfcSpace can not be retrived:

**IfcStair**

**IfcFurnishingElement**

**Elevator**

.....

**It is the work for next step!**

Thank you!