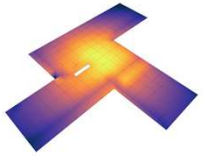
Integrating topologicpy with the semantic web: Opportunities for blockchain encoded digital twins

Wassim Jabi and Theo Dounas



26 May 2024

W3C Linked Building Data Community Group Meeting

topologicpy

CARDIFF UNIVERSITY
PRIFYSGOL CAERDYDD

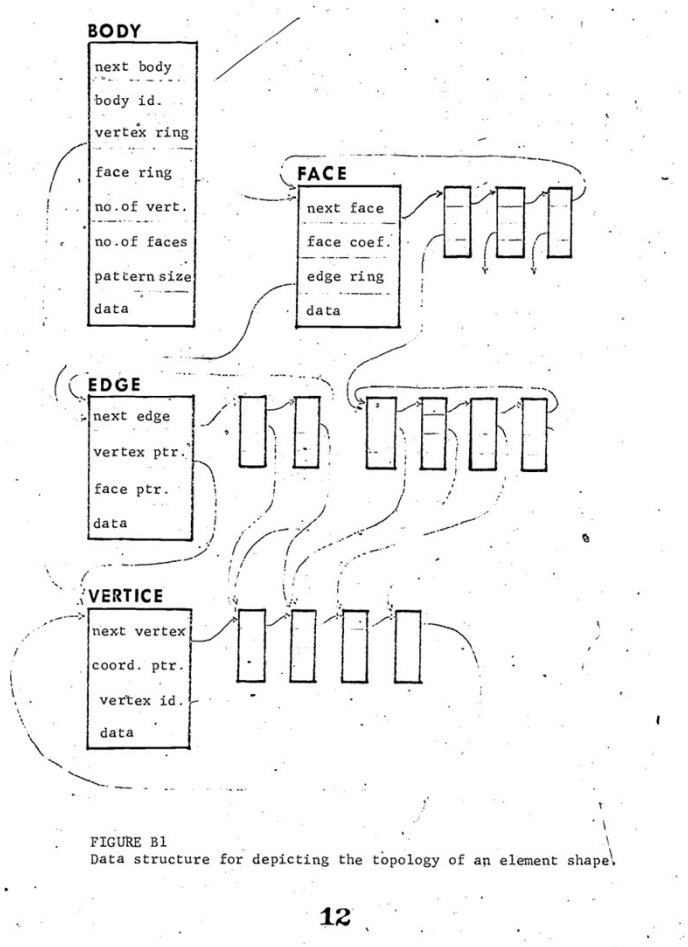


FIGURE B1
Data structure for depicting the topology of an element shape.

12

"An Outline of the Building Description System" by Charles Eastman and others 1974

Source: https://files.eric.ed.gov/fulltext/ED113833.pdf

topologicpy

CARDIFF UNIVERSITY
PRIFYSGOL CAERDYDD

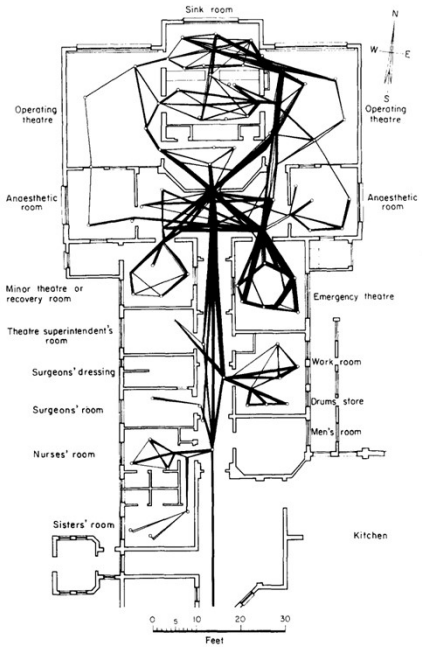
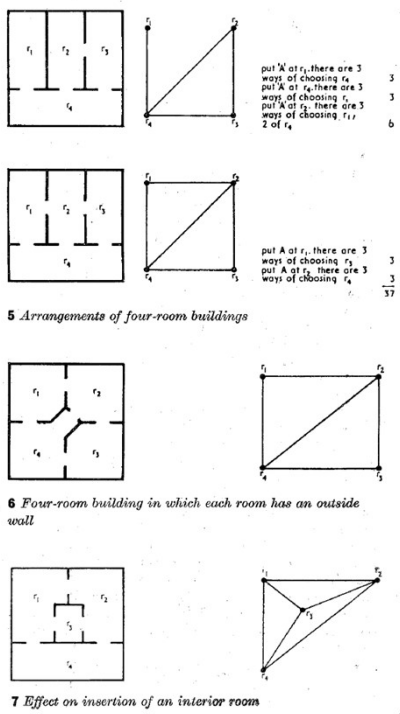


Fig. 1. String diagram of movements of state enrolled nurse during one day's duty in existing operating theatre suite.

"The Planning of Single-Storey Layouts" by Whitehead and Elders 1964



5 *Arrangements of four-room buildings*

6 *Four-room building in which each room has an outside wall*

7 *Effect on insertion of an interior room*

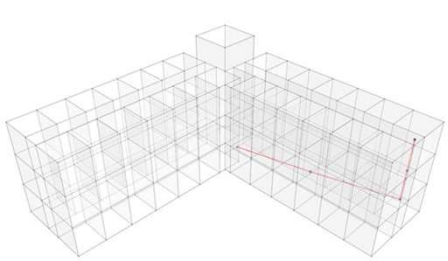
"Use of Graphs to Decide the Optimum Layout of Buildings" by Levin 1964

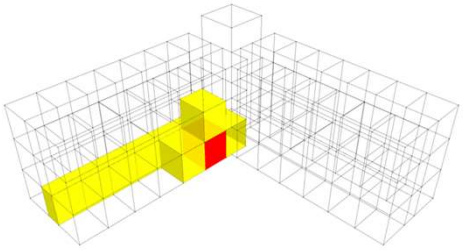
topologicpy

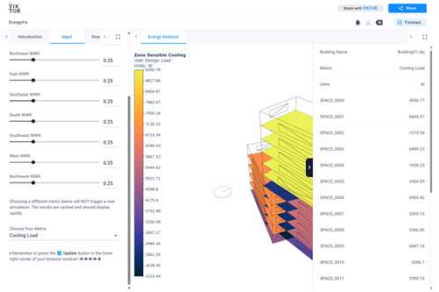
https://pypi.org/project/topologicpy/

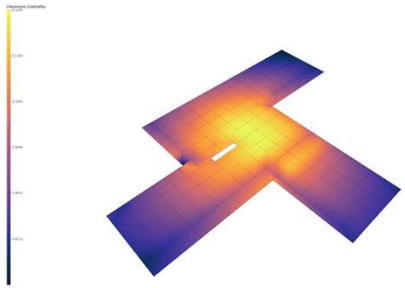
Topologicpy

Topologicpy is an AI-powered open-source python3 spatial modelling and analysis software library that revolutionizes the way you design architectural spaces, buildings, and artefacts. Topologicpy's advanced features enable you to create hierarchical and topological information-rich 3D representations that offer flexibility and control in your design process. In short, Topologicpy enriches Building Information Models with Building Intelligence Models.









topologicpy

CARDIFF UNIVERSITY
PRIFYSGOL CAERDYDD

Roots

From 2016 to 2019, a team from Cardiff University and UCL were funded by the Leverhulme Trust (£300,000) to enhance the representation of space in building information modelling. This resulted in the creation of the Topologic software library. Since 2019, Topologic and topologicpy have been in continuous development by Professor Jabi at Cardiff University.

Prof. Wassim Jabi

Project Director

Welsh School of Architecture
Cardiff University

Prof. Robert Aish

Co-Investigator

Bartlett School of Architecture
University College London

Dr. Simon Lannon

Co-Investigator

Welsh School of Architecture
Cardiff University

Dr. Aikaterini Chatzivasileiadi

Research Associate

Welsh School of Architecture
Cardiff University

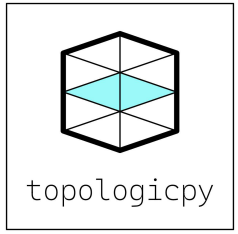
Dr. Nicholas Mario Wardhana

Research Associate

Welsh School of Architecture
Cardiff University


topologicpy


CARDIFF
UNIVERSITY
PRIFYSGOL
CAERDYDD


topologicpy


AUTODESK
REVIT


Rhinoceros


blender


IFC


VIK
TOR


Blockchain


TestFit Inc.


MACHINE
LEARNING


Linux


Dynamo


Grasshopper


SVERCHOK


Homemaker


BLENDER
BIM


neo4j


python


PyTorch


MacOS


BIM2BEM


FreeCAD


speckle.
works


OpenStudio


EnergyPlus


Ladybug/Honeybee Tools

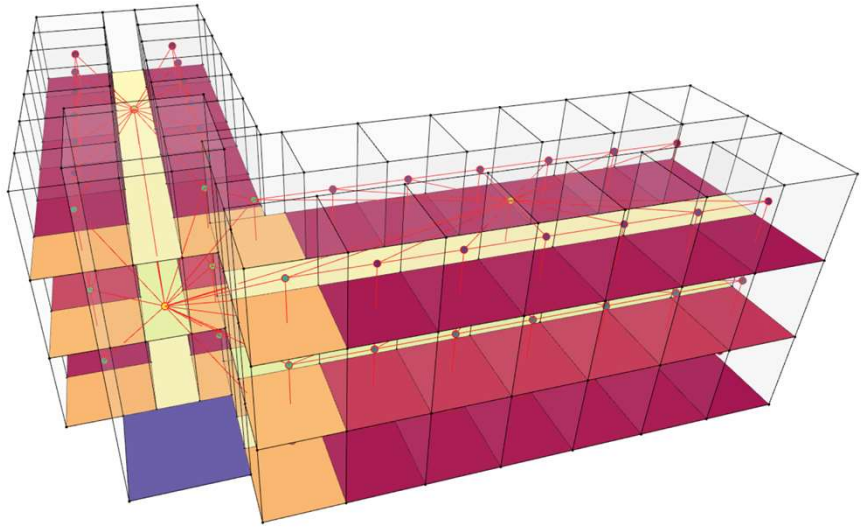

Pollination

topologicpy

https://pypi.org/project/topologicpy/

Non-manifold Topology and Graphs

Topologicpy uses non-manifold geometry to represent conceptual models with full adjacency and topological rigour. It balances a simplified geometric model with deep topological information and semantics. Specifically, it stores and processes topological, scalar, textual, logical, and compound information at several levels of the topology. Behind-the scenes, topologicpy builds primary and dual graphs of topological entities that enable spatial and performance analysis.

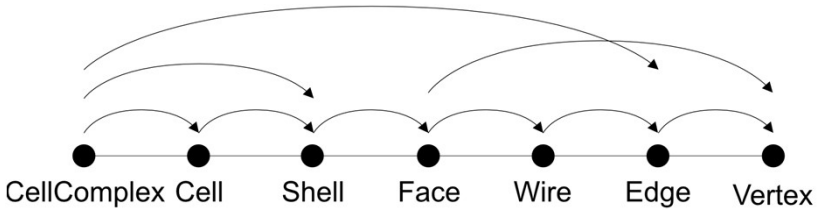


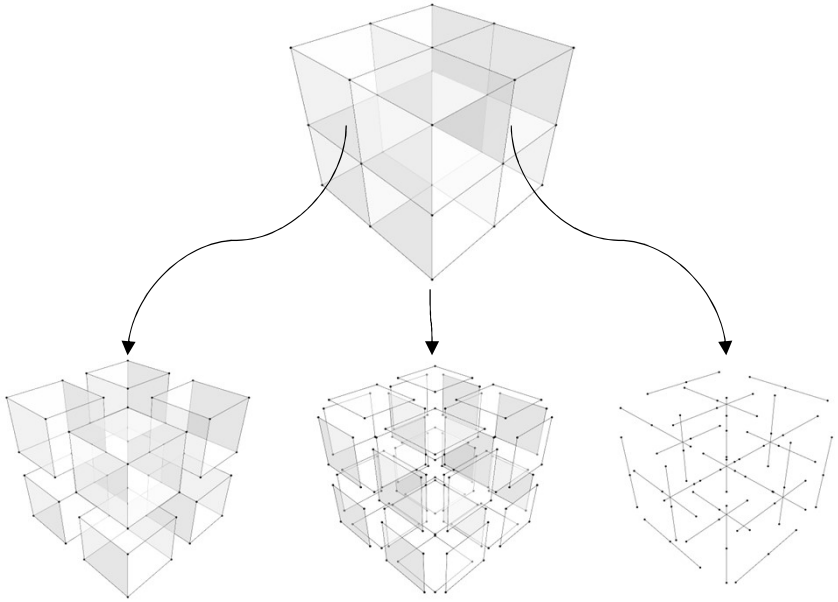
topologicpy

CARDIFF UNIVERSITY
PRIFYSGOL CAERDYDD

Decompositional Topology

In topologicpy, Entities can be decomposed into their constituent parts which, in turn, can be decomposed further. This can be both consecutive and non-consecutive



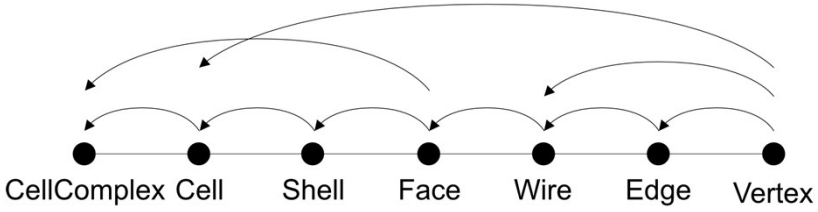


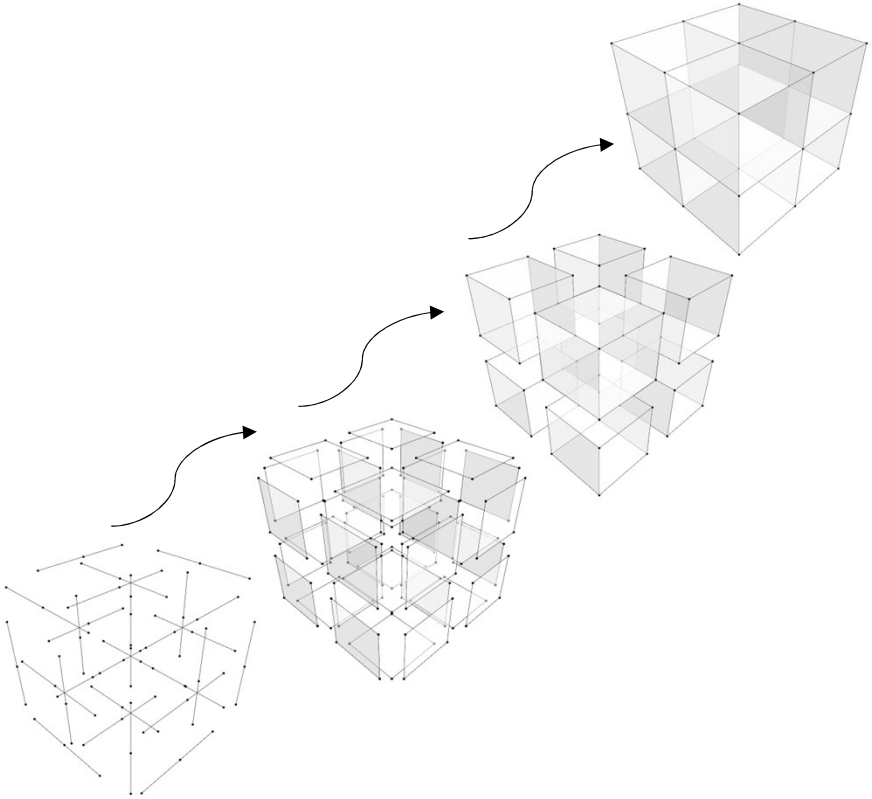
topologicpy

CARDIFF UNIVERSITY
PRIFYSGOL CAERDYDD

Compositional Topology

In topologicpy, super-topologies can be retrieved from entities which, in turn, can be queried further for their super-topologies. This can be both consecutive and non-consecutive





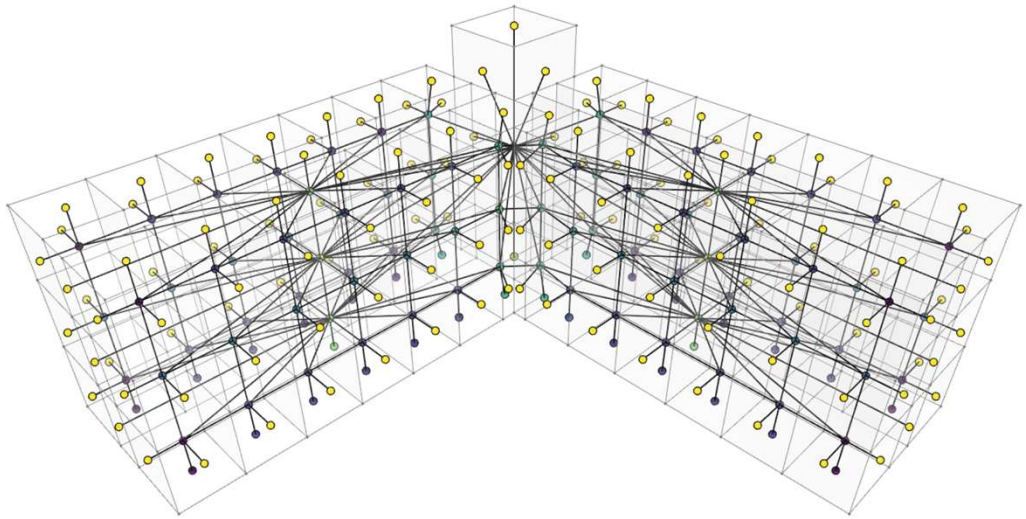
topologicpy

CARDIFF UNIVERSITY
PRIFYSGOL CAERDYDD

Graphs

Graphs can be customised by connecting additional entities:

1. Apertures (Exterior and Interior)
2. Exterior Topologies
3. Interior Topologies
4. Contents
5. Remote/Unconnected Entities

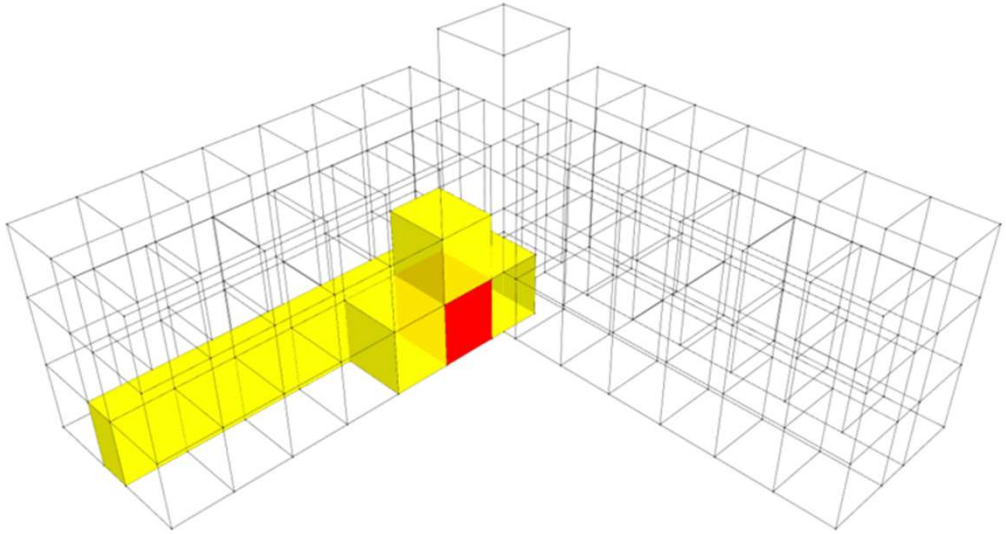


topologicpy

CARDIFF UNIVERSITY
PRIFYSGOL CAERDYDD

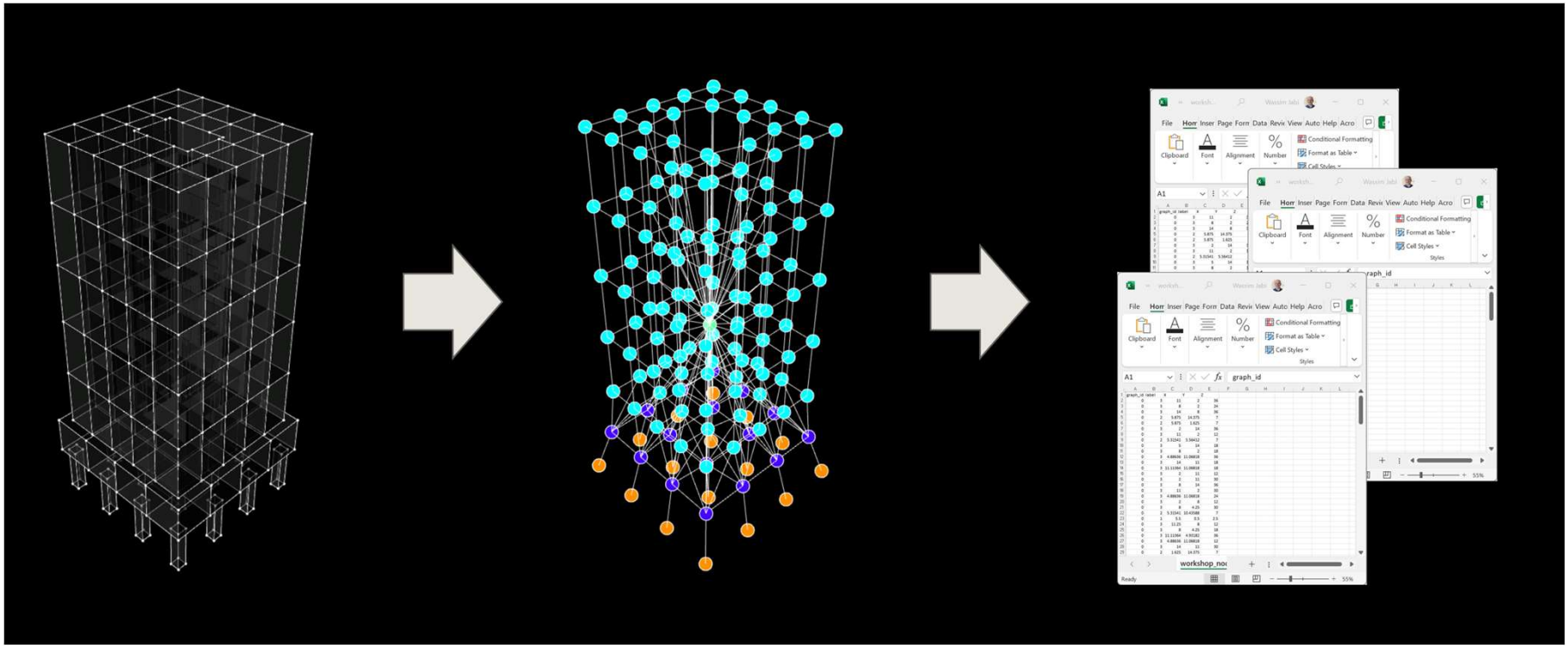
Lateral Topology (direct)

In topologicpy, entities can be queried for their neighbours.



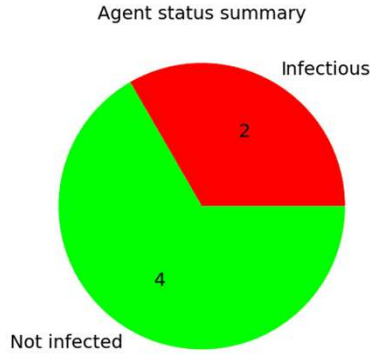
topologicpy

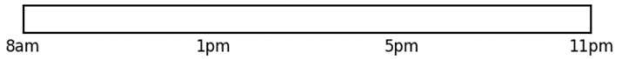
CARDIFF UNIVERSITY
PRIFYSGOL CAERDYDD

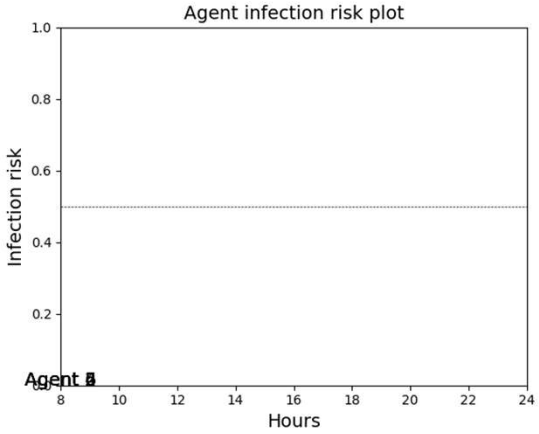


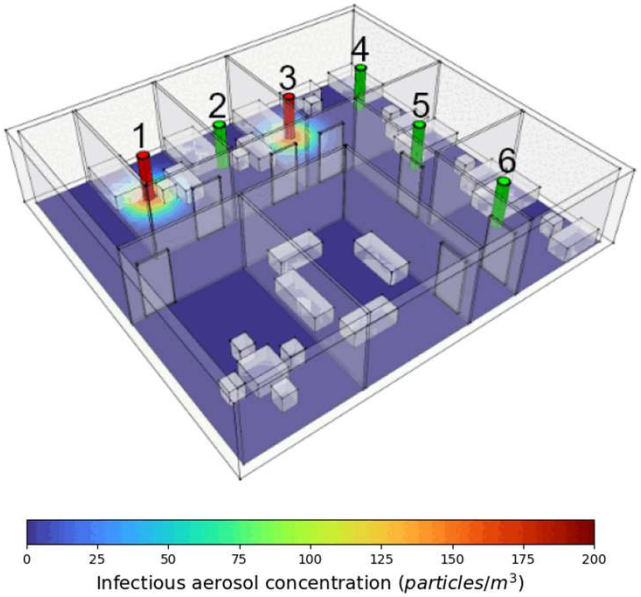
topologicpy

CARDIFF UNIVERSITY
PRIFYSGOL CAERDYDD







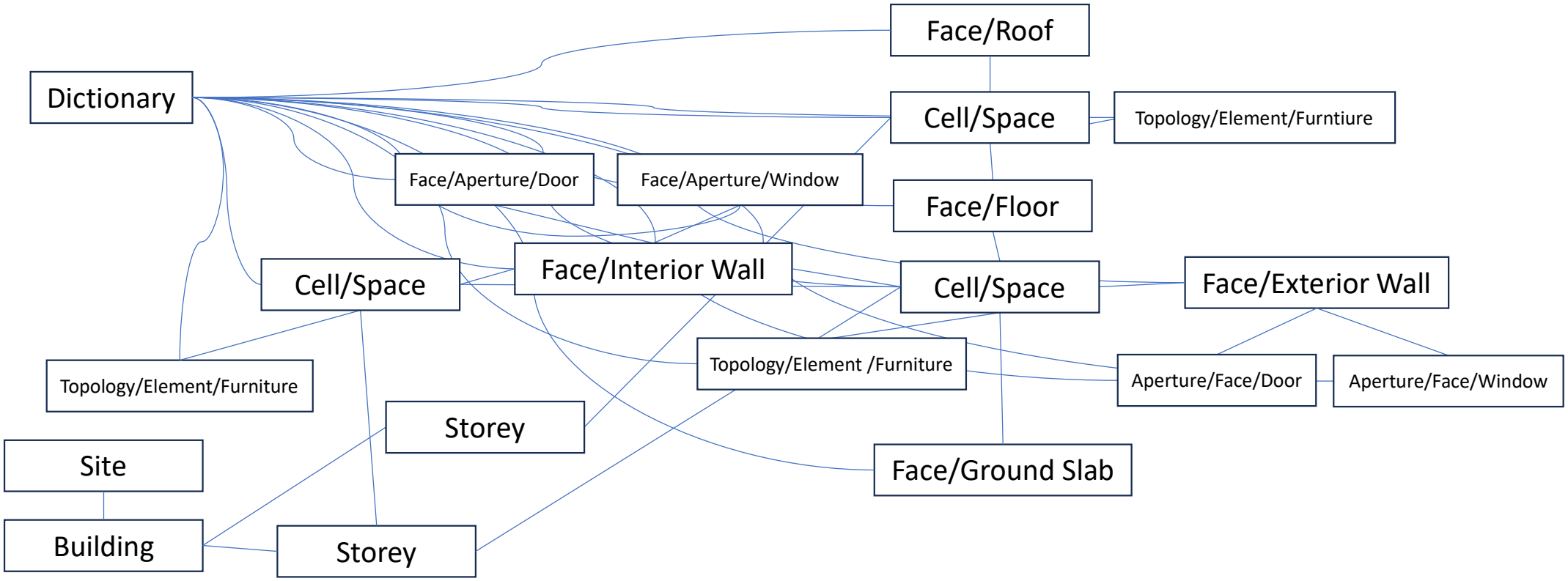


With Katerina Kaouri, Thomas Wolley, and Yidan Xue, Cardiff University

topologicpy

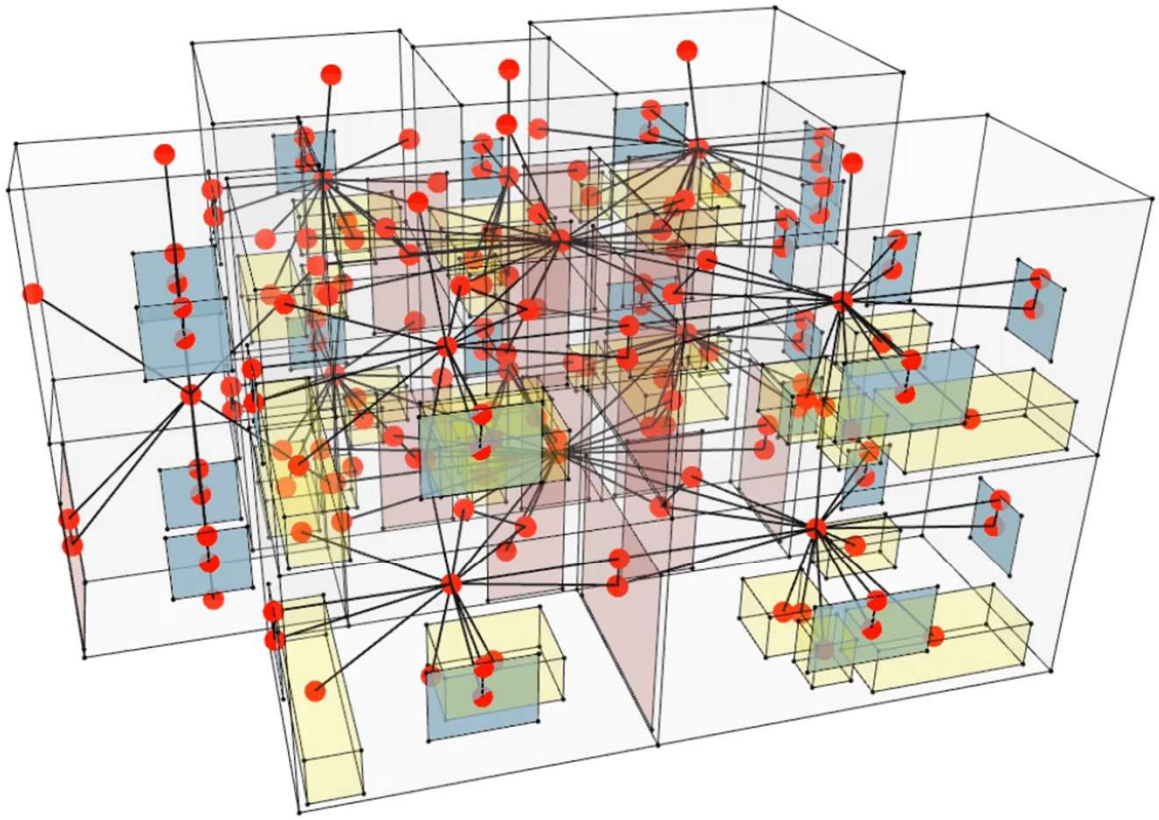
CARDIFF UNIVERSITY
PRIFYSGOL CAERDYDD

Basic Construction



topologicpy

CARDIFF UNIVERSITY
PRIFYSGOL CAERDYDD



topologicpy

CARDIFF UNIVERSITY
PRIFYSGOL CAERDYDD

Advice from Mads Rasmussen (Thanks!)

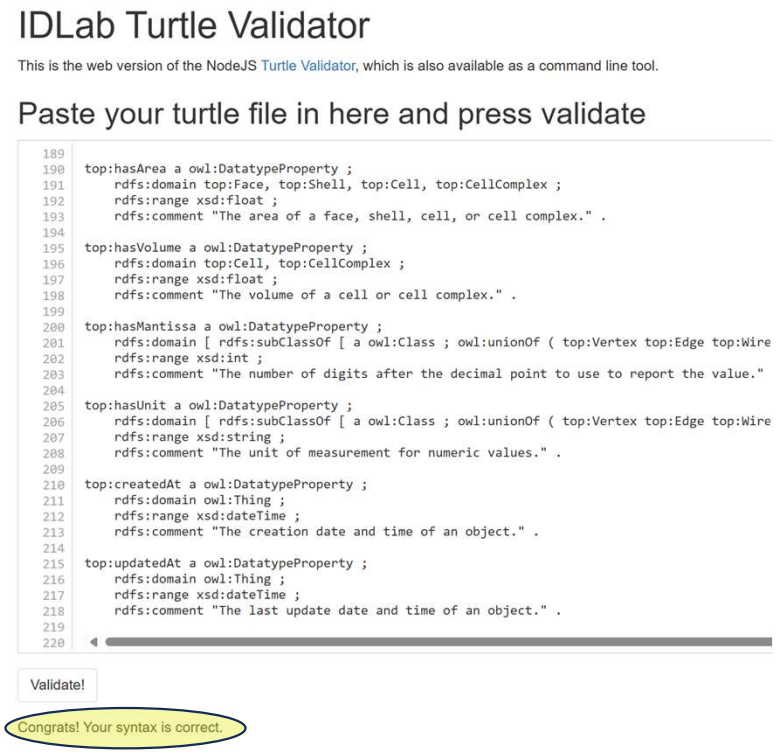
- "You should use a namespace for the triples":
  - @prefix top: http://github.com/wassimj/topologicpy/resources
- "I would recommend that you just use a hash as the id of instances instead of something human readable and I would use a hash that I derive from the content itself":
  - An element's UUID is based on its BREP string:
    - predefined_namespace_dns = uuid.UUID('6ba7b810-9dad-11d1-80b4-00c04fd430c8')
    - namespace_uuid = uuid.uuid5(predefined_namespace_dns, namespace)
    - brep_string = Topology.BREPString(topology)
    - return uuid.uuid5(namespace_uuid, brep_string)
- "BOT doesn't include specific subtypes of bot:Element, so I would suggest that you use something like ifc:IfcWall":
  - While I have code for IFC -> RDF. The generic BOT export is not related to IFC specifically. This will probably become top:Wall (sub-class of top:Face)
- "Properties":
  - Properties are now:
    - top:brepType 8 ;
    - top:brepTypeString "Face"^^xsd:string ;
    - top:x "3.0"^^xsd:float ;
    - top:y "8.5"^^xsd:float ;
    - top:z "4.5"^^xsd:float ;
- "For geometry I suggest taking a look at FOG/OMG".
  - We will. We now have the option to use hasSimpleGeometry, but this will also depend on what we do with IPFS (Theo will talk about this).
- "For the things that are unique for Topologic I suggest that you make a minimal ontology":
  - http://github.com/wassimj/topologicpy/resources/topologicpy.ttl

topologicpy

CARDIFF UNIVERSITY
PRIFYSGOL CAERDYDD

Advice from Mads Rasmussen (Thanks!)

- “To validate your Turtle, use a tool like this: http://ttl.summerofcode.be/”



Topologicpy / BOT Mapping

- Cell:
  - top:Space -> bot.Space, bot.Zone
- Face:
  - "externalVerticalFaces": "External_Vertical_Wall", bot.Wall
  - "internalVerticalFaces": "Internal_Vertical_Wall", bot.Wall
  - "topHorizontalFaces": "Roof_Slab", bot.Slab
  - "bottomHorizontalFaces": "Ground_Slab", bot.Slab
  - "internalHorizontalFaces": "Internal_Floor_Slab", bot.Slab
  - "externalInclinedFaces": "External_Inclined_Wall", bot.Wall
  - "internalInclinedFaces": "Internal_Inclined_Wall", bot.Wall
  - "externalVerticalApertures": "External_Vertical_Window", "External_Vertical_Door", bot.Door, bot.Window
  - "internalVerticalApertures": "Internal_Vertical_Window", "Internal_Vertical_Door", bot.Door, bot.Window
  - "topHorizontalApertures": "Skylight_Roof_Window", bot.Window
  - "bottomHorizontalApertures": "Floor_Window", bot.Window
  - "internalHorizontalApertures": "Floor_Window", bot.Window
  - "externalInclinedApertures": "External_Inclined_Window", bot.Window
  - "internalInclinedApertures": "Internal_Inclined_Window", bot.Window
- Content Topology (e.g. Furniture):
  - top:element -> bot.Element

topologicpy

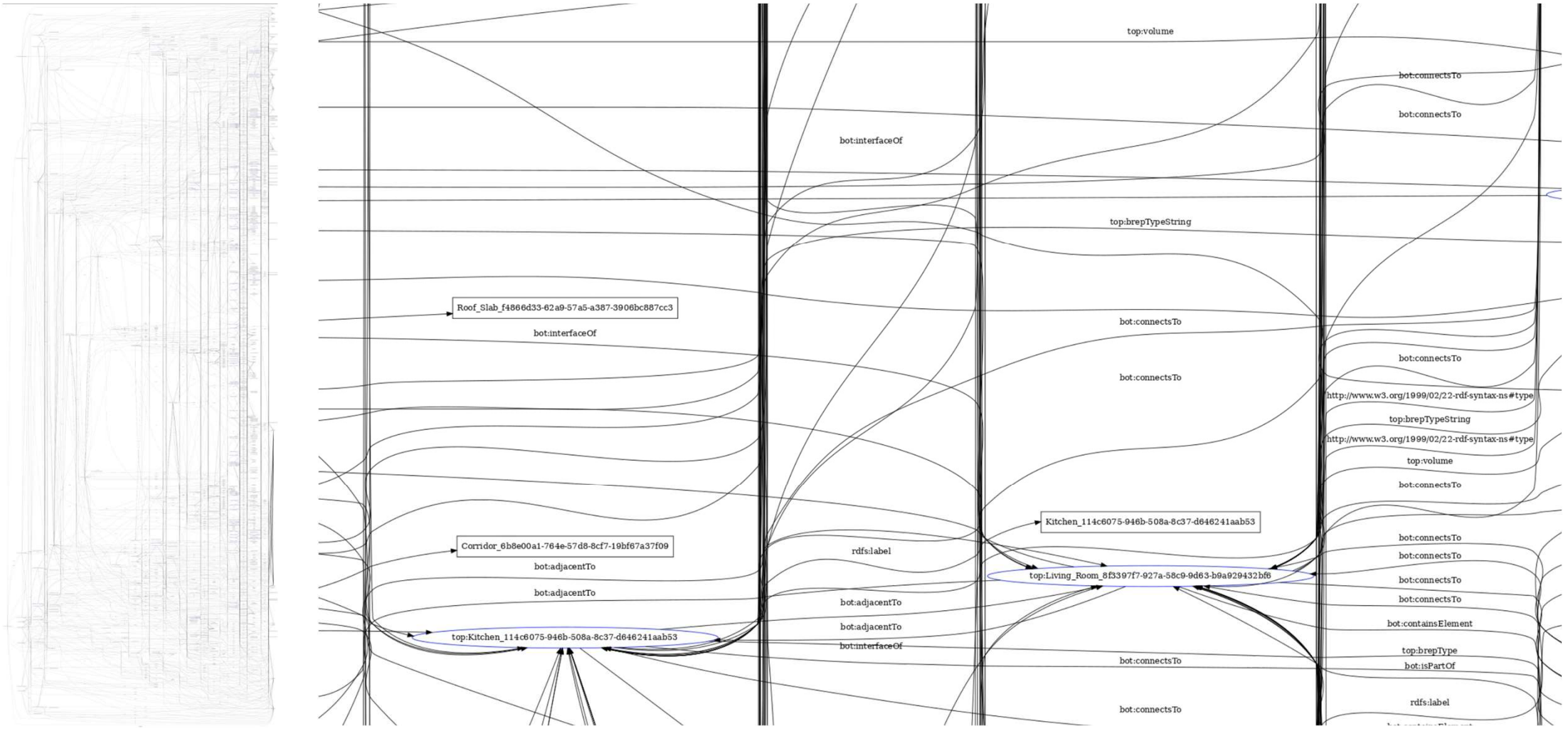
CARDIFF UNIVERSITY
PRIFYSGOL CAERDYDD

Relationships

• Space	• AdjacentTo	• Space
• Wall	• InterfaceOf	• Space
• Slab	• InterfaceOf	• Space
• Space	• ContainsElement	• Content Topology
• Anything Else	• connectsTo	• Anything Else

topologicpy

CARDIFF UNIVERSITY
PRIFYSGOL CAERDYDD



topologicpy

CARDIFF UNIVERSITY
PRIFYSGOL CAERDYDD

The element name before the UUID was temporary for debugging. Has been removed.

```
@prefix bot: <https://w3id.org/bot#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix top: <http://github.com/wassimj/topologicpy/resources> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .

top:External_Vertical_Wall_0ac4f3d4-ef43-57e6-8387-6e84f0001ac8 a bot:Wall ;
    rdfs:label "External_Vertical_Wall_0ac4f3d4-ef43-57e6-8387-6e84f0001ac8" ;
    top:area "3.0"^^xsd:float ;
    top:brepType 8 ;
    top:brepTypeString "Face"^^xsd:string ;
    top:x "8.5"^^xsd:float ;
    top:y "4.0"^^xsd:float ;
    top:z "1.5"^^xsd:float ;
    bot:interfaceOf top:Bedroom_a6a9f788-ecfd-509e-ba5a-b93913a7cfb1 .

top:External_Vertical_Wall_45fe65de-9e19-5bf8-a83d-08d1c43b8cd5 a bot:Wall ;
    rdfs:label "External_Vertical_Wall_45fe65de-9e19-5bf8-a83d-08d1c43b8cd5" ;
    top:area "3.0"^^xsd:float ;
    top:brepType 8 ;
    top:brepTypeString "Face"^^xsd:string ;
    top:x "8.5"^^xsd:float ;
    top:y "4.0"^^xsd:float ;
    top:z "4.5"^^xsd:float ;
    bot:interfaceOf top:Bedroom_eba94237-7647-5499-b059-84ce95bf2162 .

top:External_Vertical_Wall_4c01de3a-15a2-5ea7-9590-6bb01094ecc7 a bot:Wall ;
    rdfs:label "External_Vertical_Wall_4c01de3a-15a2-5ea7-9590-6bb01094ecc7" ;
    top:area "3.0"^^xsd:float ;
    top:brepType 8 ;
    top:brepTypeString "Face"^^xsd:string ;
    top:x "5.0"^^xsd:float ;
    top:y "8.5"^^xsd:float ;
    top:z "4.5"^^xsd:float ;
    bot:interfaceOf top:Home_Office_32201248-48c1-5b1b-aa7d-f2233b7bba72 .

top:External_Vertical_Wall_60b22c8f-044f-5e1c-b72b-6e85ac1afb8f a bot:Wall ;
    rdfs:label "External_Vertical_Wall_60b22c8f-044f-5e1c-b72b-6e85ac1afb8f" ;
    top:area "3.0"^^xsd:float ;
```

Live Demo!

topologicpy

CARDIFF UNIVERSITY
PRIFYSGOL CAERDYDD

Thank You.

Wassim Jabi

jabiw@cardiff.ac.uk

https://pypi.org/project/topologicpy/

https://github.com/wassimj/topologicpy/blob/main/resources/topologicpy.ttl

https://pypi.org/project/topologicpy/