Automated analysis and quantity calculation of balcony elements in IFC

Hesselink, Geert; Krijnen, Thomas; Pannekoek, Geert

Publication date
2021

Document Version
Final published version

Published in
Proceedings of the 38th International Conference of CIB W78

Citation (APA)
Hesselink, G., Krijnen, T., & Pannekoek, G. (2021). Automated analysis and quantity calculation of balcony elements in IFC. In *Proceedings of the 38th International Conference of CIB W78* (pp. 272-281). (CIB W78 conference series). https://itc.scix.net/paper/w78-2021-paper-028

Automated analysis and quantity calculation of balcony elements in IFC

Geert Hesselink, geert.hess@gmail.com
Schöck Nederland, Apeldoorn, The Netherlands

Thomas Krijnen, t.f.krijnen@tudelft.nl
Luxembourg Institute of Science and Technology, Luxembourg, Luxembourg

Geert Pannekoek, g.pannekoek@schock.nl
Schöck Nederland, Apeldoorn, The Netherlands

Abstract
In this paper, we discuss an automated procedure to automatically analyse and extract domain specific construction information from IFC building models. More specifically, we were interested in extracting location potential placement sites for thermal bridges between balconies and their neighbouring floors. For this aim, we developed a web-based platform where balconies could be manually selected to be analysed at a platform which was built upon open-source computational frameworks such as OpenCascade and IfcOpenShell. Moreover, steps to automate classification of balconies and adjacent floor, without manual selection, are discussed along with shortcomings and possible solutions regarding these classifications. The output consisted of elementary attributes such as geometrical coordinates, mass and volume as well as more sophisticated attributes such as the cantilever direction and the equality between balconies. These attributes can then be embedded in practical, day to day, operations.

Keywords: Geometry, BIM, Building Information Modelling, IFC, IfcOpenShell, OpenCascade, Balconies

1 Introduction

The usage of Building Information Modelling (BIM) in the workforce has increased greatly in the last years as it offers structured exchange of information on physical and functional characteristics of construction works of information on construction works. BIM and the commonly used collaboration format, IFC, are predominantly used in the engineering phase of a construction project, but the connection of BIM to manufacturing and Enterprise Resource Planning is also investigated (Babič et al., 2010). Specifically for off-site manufacturing there is a detailed overview presented by Abanda et al. (2017). In a somehow related fashion, we see interest in BIM adoption in tendering (Ciribini et al., 2015) and cost estimation (Zhiliang et al., 2011).

However, certain analysis tasks are hindered by a lack of standardization. In particular, there is in current practice no explicit classification of balcony parts and data pertaining to the exact construction of the balcony is not explicitly provided. More specifically, balconies are classified as either a ‘IfcSlab’, ‘IfcBeam’, 'IfcBuildingElementProxy', another classification or no classification whatsoever. The (at times, lack of) visual representation is likewise varying. For instance, representation of balconies is sometimes basic (see Figure 1[a]) as is the case in elementary architectural models. They can be displayed in a more complex fashion as is the case in more extensive models, such as in structural models (see Figure 1[b]) in which the enclosing elements such as columns and beams are also represented. Lastly, representation of balconies are at times very extensive (Figure 1[c]). This is the case in IFC models that display many relevant enclosing elements, e.g. the neighbouring floor consisting of an isolation layer and two concrete layers, as well as very detailed information about the balcony such as fencing and gutters (Figure 1[d]). Complexity anywhere within the range of these four examples is possible and, in general, uniformity in balcony representation would lead to increasingly successful and informative analysis.

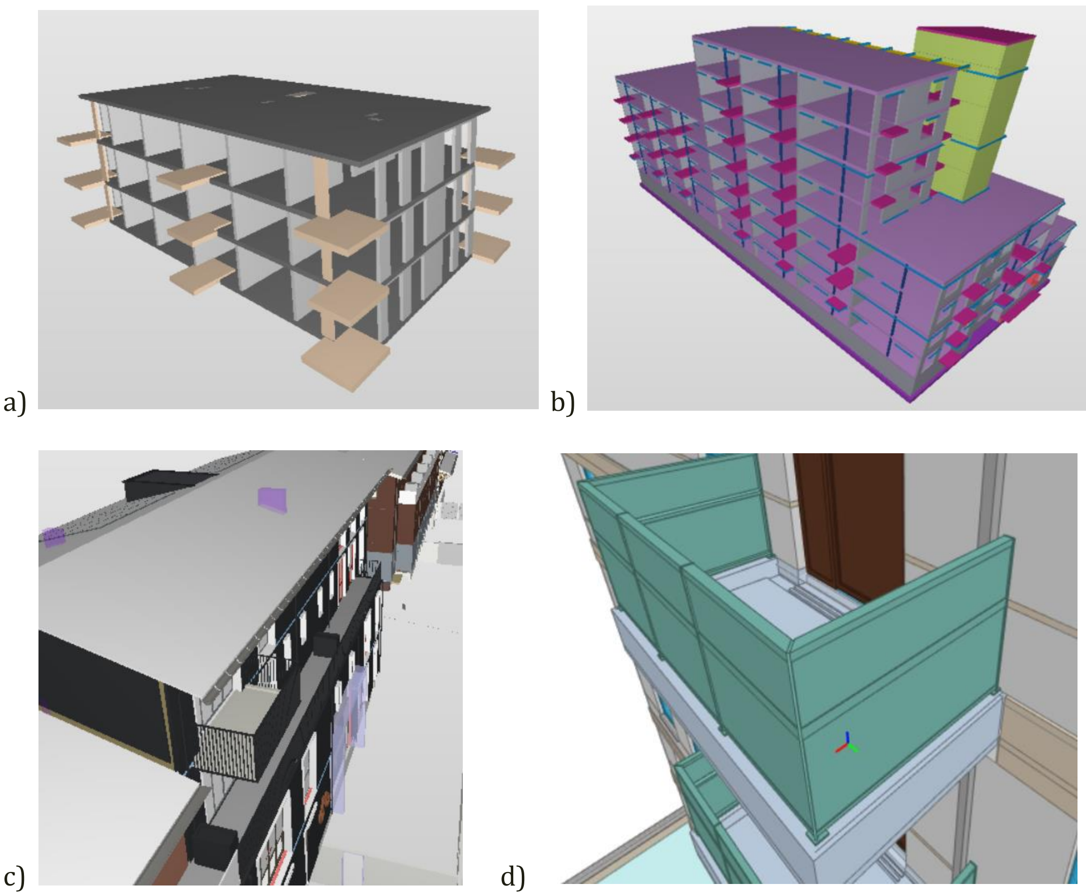


Figure 1. Basic balcony representation in elementary IFC model (a), structural model (b) elaborate architectural model (c) and balcony representation including gutters where slabs are not convex boxes (d).

This lack of standardization is, arguably, increasingly problematic in the later stages of the construction process, as there is more dependency on the quality of the information of earlier stages. In addition, there is the matter of information reliability of the non-geometric meta-data. It would be tempting to rely in the 'IsExternal' Boolean (true or false) property that can be associated to a wide set of IFC element types by means of the property set mechanism, for example PSet_SlabCommon for IfcSlabs. However, research has indicated that his information is rarely reliable (Luttun and Krijnen 2020).

1.1 Current state of the art

As part of this research, carried out by Schöck Netherlands, a company that sells thermal break connections for balconies, an automated procedure has been developed to automatically extract coordinates of potential placements sites for such thermal break connections and to calculate necessary quantities for analysis. Thermal break connections are thermal insulators used between two connections to prevent thermal or cold bridging. In the case of the products sold by Schöck Netherlands, they are used between balconies and their adjacent floors, preventing undesired consequences of thermal bridges such as moisture penetration in building components and mould growth. For this goal, current available tools and technologies did unfortunately not suffice. BIM-related software capable of manipulation and extraction of information of IFC files - such as Revit[1], BimVision[2] or Blender[3] - did not provide tools nor additional plugins for this aim. Another approach was to use machine learning to obtain the required implicit information. In Krijnen & Tamke (2015), both supervised and unsupervised machine learning methods were used to differentiate elements based on their geometrical appearance. Unfortunately, for the current research these methods did not suffice. In Figure 2 we outline one of the difficulties with using (machine) learning approaches. On the one hand, for simple models, there is indeed a clear boundary between what constitutes a main floor and a balcony, for example by looking at volume and number of neighbours. On the other hand though, for models with a higher level of detail, the number of relationships increase dramatically. This increases the number of relationships dramatically, and a large variance of volume is seen due to slabs being used to articulate the fabrication detail. Hence, there is a less clear topological view on the functional connectivity graph of slabs and no longer a clear separation between floor and balcony elements.

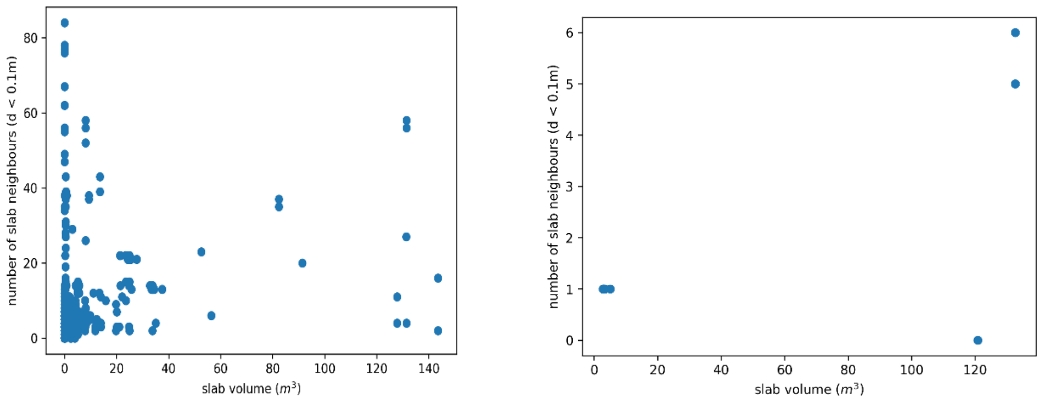
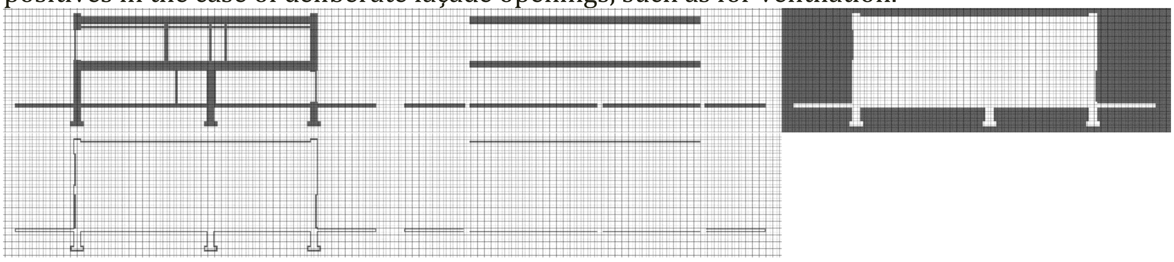


Figure 2. Scatter plots of slab volume and neighbours. In the case of a simple model (a) we see a clear division with smaller slabs with a single neighbour as balconies, a large roof slab with zero neighbours, and a large main floor slab with many neighbours. In the case of a more complex, architectural, model (b) there is a higher level of detail and therefore the distinction is a lot less clear.

Additonaly, in Figure 3 we show some experiments we did with voxelization of IFC building models to come to a segmentation of interior and exterior elements. This experiment was based on an open source library for voxelization[4]. While technically the solution worked as intended, this approach was not useful for structural aspect models without the architectural façade. This is however a significant part of the models delivered to the company. It also resulted in false positives in the case of deliberate façade openings, such as for ventilation.



[1] Revit [Computer software]. Retrieved from https://www.autodesk.com [accessed April, 2021]
[2] BimVision [Computer software]. Retrieved from https://www.bimvision.eu [accessed April, 2021]
[3] Blender [Computer software]. Retrieved from https://www.blender.org [accessed April. 2021]
[4] Voxelization toolkit [Computer software]. Retrieved from: https://github.com/opensourceBI/voxelization_toolkit [accessed April. 2021]

Figure 3. (a) voxelized representation of the IFC building model, (b) voxelized representation of only the IfcSlab elements, (c) exterior space obtained by a flood fill over unset voxel elements starting from a corner outside of the model extends, (d) the voxelized exterior shell as the neighbours of the voxel set in 'c' and (e), the intersection of exterior shell and slab voxels: interior slabs are absent, external floor and roof slabs are present with one of their horizontal faces and balcony slabs are present with both horizontal faces.

Furthermore, Wu and Zhang (2019) describe certain geometric signatures of BIM elements, but do not differentiate between more granular sub types of building elements and do not include the contextual surroundings of building elements necessary to differentiate a balcony slab from a normal indoor floor slab. Augmenting the used classifications as information requirements was not an option as it may turn away clients.

Ultimately, computational geometry methods were applied by using various open source software, such as the widely used CAD kernel Open CASCADE Technology (OCCT)[6] for boundary representation modeling, IfcOpenShell to convert the implicit geometry into explicit geometry, PythonOCC[5] to access the geometry in Python specifically and ifc-pipeline[4] for web-based processing and visualization.

2 Methods and Assumptions

As discussed, we have considered using automated analysis to identify balconies in the model using supervised learning and geometric and topological analysis, but due to the wide variety of input this currently did not give satisfying results. This paper focusses on the analysis and extraction of geometric quantities once an identification of balconies has been made manually(Figure 4). Furthermore, there is an option to only extract elementary data from the balcony (such as mass, volume and geometrical coordinates) if more sophisticated analysis is not required. The output of the analysis was an xml file in a format compatible with the software used by Schöck to determine a location suitable for thermal bridge placement or other further analysis. The content of the output is moreover not limited to this format, which will be further discussed in the 'exchange requirements' chapter.

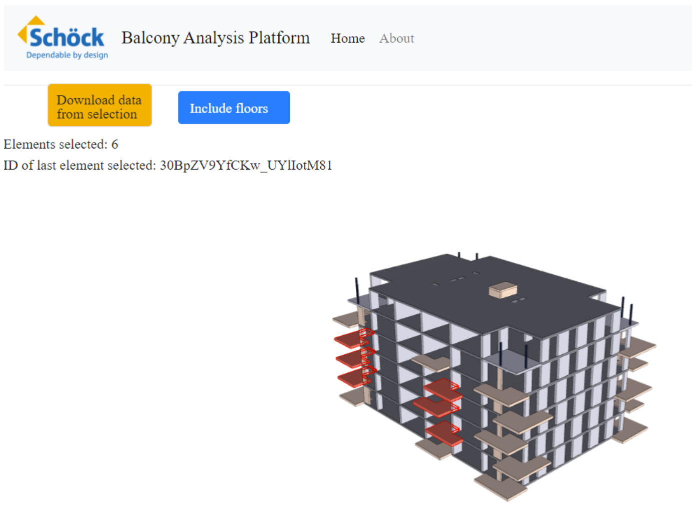


Figure 4. Web-based interface developed for this research project with manually selected balconies

Technically the algorithm is as follows. First, as the connection between balconies and floors is crucial for the structural connection, pairs of neighbouring floors are identified using a lookup in a spatial hierarchy. A first selection of candidates using bounding boxes is used for performance (see Figure 5[a]). These pairs are filtered based on mass, footprint area and topological constraints. Balconies are always the leaves of the slab connectivity graph. Faces of the boundary representation are subdivided into top and side faces.

[5] ifc-pipeline https://github.com/AECgeeks/ifc-pipeline [accessed May 2021]

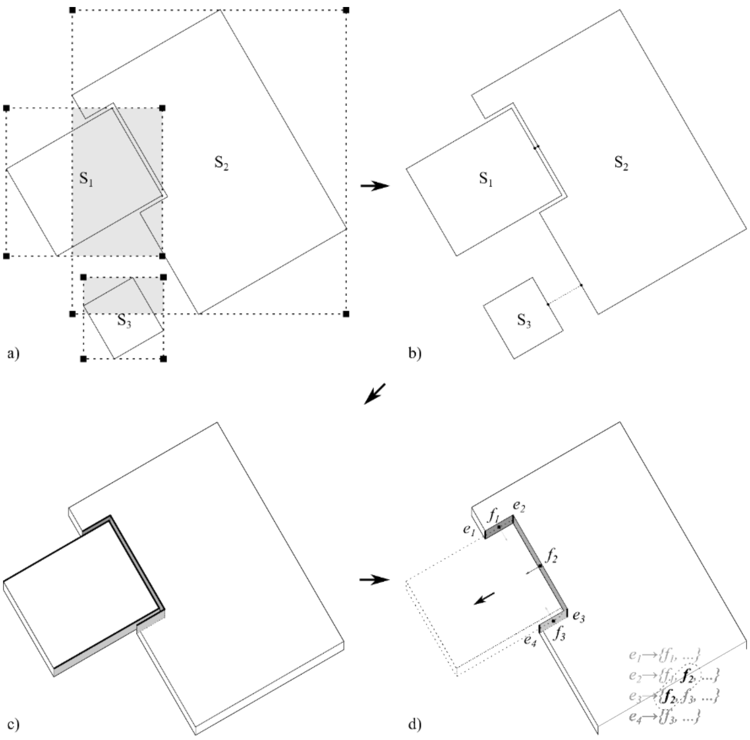


Figure 5. *Extracting and analysing the balconies in four steps; a) gathering pairs of floor-balcony candidates by means of bounding box overlap in a spatial tree b) further filtering candidates based on solid to solid distance c) on the level of sub-shapes (edges and faces) find corresponding pairs between floor and balcony d) analysing the cantilever direction, which in this case follows from the topology and face surface normal of the face that is an ancestor of multiple edges in the selected set of overlapping entities.*

A Boundary Representation solid model CAD library such as OpenCascade[5] uses hierarchy of topological entities: Solid > Shell > Face > Loop > Edge > Vertex. Geometrical entities are associated to Face, Edge and Vertex, namely an underlying surface, curve and point. Face adjacency is analysed between balcony and main floor with a geometric tolerance. To place a thermal bridge connection, distance between two neighboring elements cannot be greater than 12.5cm. Furthermore, it is important in the analysis to discard elements that do not overlap or touch, for example elements in different storeys. This also leads to a reduction in calculation time as additional advantage. When faces are parallel, the projected geometric overlap is sufficient and the distance not larger than the above mentioned distance, the pair of faces is a potential placement site for the thermal break connection. From this connection point (or multiple connection points) follows the cantilever direction, length and balcony width, with the length being in the cantilever direction (see Figure 5[b]).

Specifically, the projection direction is often difficult to calculate. When there is a single pair of parallel faces passing the earlier mentioned criteria, the projection direction is perpendicular to the face of the floor. For now. the requirement for automated extraction is that these faces have planar underlying surfaces. In situations where there are two or more pairs of faces that pass the criteria (see Figure 5[c]), the face perpendicular to the balcony can be filtered out by looking at the faces of the floor from a topological view. This will be further specified in the implementation section (3.2).

Further parameters of interest were coordinates of the two-dimensional corner points for both balconies and the adjacent floor as well as parameters suck as the thickness, mass, support type, centre of mass, hmin and the quantity. The hmin is defined as the vertical overlap of the balcony and the adjacent floor at the point of connection. This can be used to determine potential points for thermal break placements. Lastly, these attributes were furthermore used to determine equality between balconies. To account for balconies that are equal in shape, but not in the global

[6] OpenCascade [Computer software]. Retrieved from https://www.opencascade.com [accessed April 2021]

coordination system of the IFC (such as when balconies with an equal shape are on opposite sides of the building) we developed a local coordination system.

3 Implementation

The IFC Balcony analyser platform presented in this paper is built mainly on top of OpenCascade for the boundary representation modelling and analysis. The interface with IFC and the conversion of implicit procedural geometry definitions is handled by IfcOpenShell. PythonOCC offers a convenient Python binding to OpenCascade that facilitates rapid prototyping of the algorithm. Visual representation and loading of IFC files was accomplished thanks to ifc-pipeline. These software and tools allowed us to further concretize the steps described in the previous chapter.

3.1 Bounding box and topological constraint

The initial step was to develop an algorithm to filter elements into pairs of two elements; balconies and their respective neighbouring floors. An axis aligned bounding box models the geometric extent of an element by means of 2 points. Using bounding box overlap as a first test for element adjacency is an efficient method to quickly discard elements that do not overlap or touch. In this case the ifcopenshell.geom.tree structure is used which in turn depends on the OpenCascade NCollection_UBTree.

The gap between the elements was determined by computing the minimum distance between points on the two corresponding shapes by the OpenCascade BrepExtrema_DistShapeShape class. These criteria filter out elements into possible balcony-floor pairs. By means of additional constraints the relationships are further specialized. For example, a balcony always has less mass and volume than their neighbouring floor and a balcony can only have one neighbouring floor, whereas a floor can have multiple neighbouring balconies.

3.2 Orientation

To determine positions of faces of the neighbouring floor relative to the balcony the corresponding edges were grouped with the Map_Shapes_And_Ancestors functionality. In OpenCascade, topological entities have a set of children stored in a member attribute, but not the reverse; parent topology is not directly stored. This is particularly helpful to select faces suitable for support regarding thermal bridging. In Figure 5[c], the adjacent floor consists of three faces within a short distance (calculated in the previous step) from the balcony and, moreover, these three faces have four edges in common. The distinctive feature of the perpendicular face is that the two edges of the face perpendicular to the balcony each borders to another face, and therefore can be filtered out to calculate the projection direction.

3.3 Cantilever direction

Determining the cantilever direction (the direction that extends from the connection point) of the balcony relative to the neighbouring floor is not something that can be done by looking at the elements in isolation, as typically both floors and balconies are defined as two-dimensional footprints that are extruded vertically. Therefore, a geometric and topological analysis was conducted.

First, the face perpendicular to the balcony as determined in the previous steps were selected. Then, a normal vector was calculated by extracting the cross product from the projection of the vertex points of the balcony on the face of the neighbouring floor. From this normal and the related dot product, the cantilever direction and cantilever distance from the face of the neighbouring floor could be calculated.

3.4 Elementary attributes

Vertices of both balconies and the neighbouring floor of the global coordination system were outputted in a xml file. From these global coordinates, the thickness of the floor at the boundary point between the balcony and floor was extracted (see Listing 1). The mass and centre of mass were then calculated: a shape of the balcony or neighbouring was created with IfcOpenShell and used with the BRepGProp_VolumeProperties to extract the mass and centre of mass. At last, the support type was extracted from the semantics of the IFC file. For the models we tested this was the IfcProperty Set, specifically.

Listing 1. Showing a code snippet of calculating the mass, centre of mass and support type

```
# Obtain mass and centre of mass
import OCC.Core.GProp
import ifcopenshell.geom
shape = ifcopenshell.geom.create_shape(
   ifcopenshell.geom.settings(), balcony).geometry
props = OCC.Core.GProp.GProp_GProps()
OCC.Core.BRepGProp.brepgprop_VolumeProperties(shape,props)
mass = props.Mass()
centre_of_mass = props.CentreOfMass()

# Obtain support type
for p in balcony.IsDefinedBy:
 if p.is_a('IfcRelDefinesByProperties') and \
      p.RelatingPropertyDefinition.is_a(ÍfcPropertySet'):
   for props in p.RelatingPropertyDefinition.HasProperties:
     if props.Name == 'MATERIAL':
       supportType = str(props.NominalValue.wrappedValue)
```

3.5 Quantity calculation

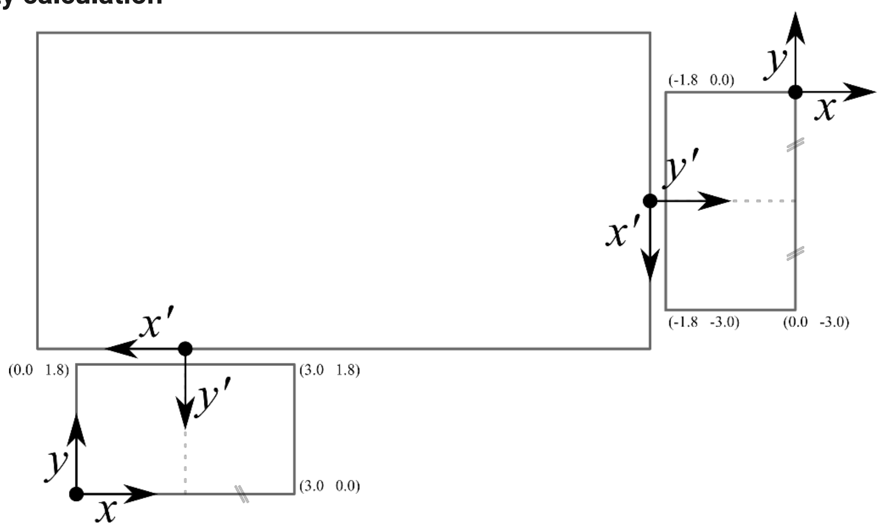


Figure 6. Two identical balconies with different IFC representations. Even with the ObjectPlacement mechanism in IFC that defines geometries in a local coordinate system, identical balconies in IFC might have different local footprint coordinates. Walls and Slabs in IFC do not typically use the IfcMappedItem mechanism of establishing reuse of geometry definitions. There is no consistency in how local placements are defined for Slabs. For that purpose we transform the Slab coordinates to a new coordinate system (x' y') defined by the center of the overlap at the supporting face oriented along the surface normal.

To determine the amount of occurrences of a balcony within an IFC model we had to check for equality between two balconies (see Figure 6) by looking at basic attributes such as mass, volume and geometrical coordinates. We sorted balconies into buckets in a hash table using these basic attributes as a hash function and checked for equality subsequently. Additionally, the two-dimensional corner points had to be transformed to a local coordinate system constructed from the adjacent face and normal to work around inconsistencies in the local placements in the IFC building model. In the case of equality, balcony elements and engineering decisions by the structural engineer can be re-used, saving time and resources. Specifically, we transferred the points of the balcony from a 3D global coordination system to a 2D with the gp_Trsf2d Class and compared the new coordinates from the gp_Ax2d Class.

4 Results

The predominant usage of geometric information allows for reliable and consistent feedback about potential placement sites. Additionally, the input can be easily inspected and edited through the web-based visualization interface used in the prototype. We tested eight models (see Krijnen & Hesselink, 2021) and looked at the time it took to load the IFC file, the time it took to analyse the file and the quality of the analysis (see Table 1). The first observation is that an increase in file size leads to a subsequent increase in analysis time, which was expected. Furthermore, most models were fast to analyse. The loading and analysis time only increased when the IFC models were increasingly complex. The analysis successfully lead to a complete dataset to output in the xml file including all the attributes of the balcony and neighbouring floor that are discussed in the exchange requirements chapter. Obtaining attributes of balconies was successful in all models except for [a3] in which the balcony were modelled as a compound of several layers, with each layer modelled as a distinct slab. The other compounds were, however, not passing the predefined filters (e.g. based on mass and volume); unlike the neighboring floor, which could be pointed out correctly by the algorithm. Manual selection of the different compounds could solve this problem relatively easily. Conversely, attributes of the neighbouring floor were correctly outputted in all models but for one exception, [a2], in which the neighbouring floor consisted of multiple layers (an isolation and concrete layer) causing correct topological data but missing volume. However, this extra layer could also be selected manually.

Table 1. Results of testing platform performance

Model	Size (in kb)	Num. balconies	Load time (in ms)	Analyse time (in ms)	Succes of analysis balcony/floor
[a1]	388	15	11 453	290	Analysis successful
[a2]	255 779	8	203 408	7 392	Balcony data incomplete
[a3]	8 261	40	34 621	15 212	Floor data incomplete
[a4]	1 285	30	24 844	1384	Analysis successful
[a5]	35 347	35	408 907	13 291	Analysis successful
[a6]	38023	39	349592	10394	Analysis successful
[a7]	1536	57	17430	4730	Analysis successful
[a8]	1831	23	25545	3952	Analysis successful

5 Discussion

In addition to the automatic quantity extraction, we have also attempted to automate classification of balconies and adjacent floors, but this often proved to be unreliable due to factors such as misattributions in the entity classification (e.g. a balcony classified as IfcBeam instead of IfcSlab), decomposition of balconies into multiple entities, overly complex balcony geometries, absence of division between internal and external elements and the variability in IFC input files. In some structural discipline models the neighbouring floors were absent, some models included the facade, others did not. The results of this classification steps are provided and discussed in this paper, but for this reason, currently, users are required to visually select balconies beforehand after which the automated extraction and analysis happens.

A possible solution to the classification problem is to make prerequisites for analysing IFC in this particular tool. This is certainly a possibility in, for example, early stages of construction progresses.
In this context, in a 'perfect IFC file' balconies and neighbouring floors are represented as one single element classified as an IfcSlab and do not contain additional details that are irrelevant for calculating potential placement sites.

Additional checks for balconies consisting of more elements were out of the scope of the current paper. The difficulties encountered with layered elements where the layers are separate superimposed slab elements partly stems from the topological nature of our analysis. In this purely topological view only connectivity and proximity is considered, by adding geometric predicates, such as above and below, the analysis can perhaps be made more robust with respect to layered elements.

5.1 Exchange Requirements

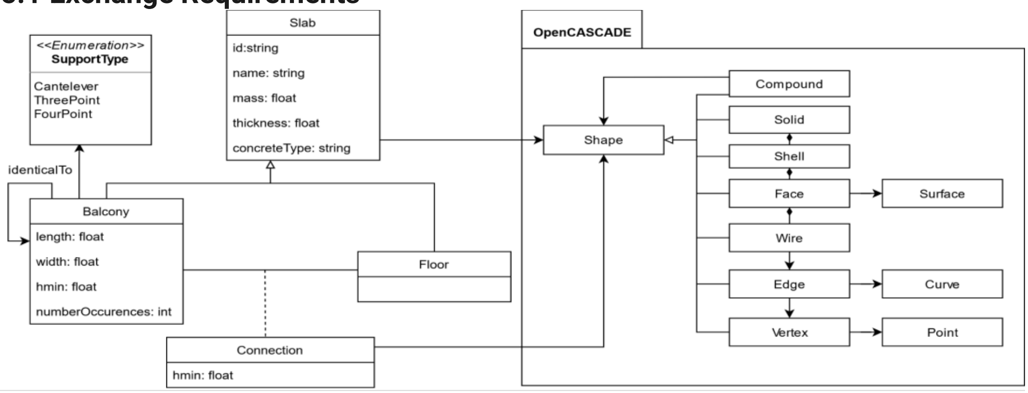


Figure 7. Conceptual data model of the extraction. Balcony and Floor are both subtypes of a type Slab that roughly corresponds to IfcSlab (but can be other types of IFC elements). Connection describes the geometric overlap between Balcony and Floor. In the vertical sense this is summarized as hmin.

IFC is envisioned in the community as part of a framework with other standards and concepts such as the Information Delivery Manual (IDM) and Model View Definitions (MVD). In principle, the IDM would contain process maps that define the information exchanges between stakeholders and MVDs impose then additional constraints on the exchanges such as the necessary inclusion of certain entity type, relevant property data or dictate the usage of certain geometric constructs. From Figure 7b it can be read that in theory there is a lot of overlap with what the IFC standard supports, including the IfcRelConnectsElements objectified relationship with the option for ConnectionGeometry.

In practice though, the development of such IDMs and MVDs is an elaborate process that is not often undertaken. MVDs are not typically implemented in software in a way that end-users are able to specify user-defined views during export. In addition, specifically for this project, the suppliers or manufacturers tend to receive models that were primarily intended as a means for coordination during design and engineering. Even more so, imposing additional requirements on this exchange would potentially hurt the opportunity for sales, so in this project the aim is to be as lenient as possible when it comes to receiving data, which results in an additional amount of variability in quality and content. It is envisioned though that over time more semantic data related to manufacturing will find its way into the BIM models. The recently started buildingSMART IDS project which aims again at enabling project stakeholders to formulate exchange requirements will be relevant to that. Contrary to mvdXML which embodies model constraints as explicit graph queries with somewhat weak semantics (Moult and Krijnen 2020; Bus et al. 2019), the IDS project chooses a limited set of facets (e.g property, material) using a higher level schema disconnected from the IFC instance graph to reduce the implementation effort of full graph query and matching and make specifications more agnostic of the IFC schema version.

6 Conclusion

The IFC schema offers an attractive means to automatically analyse and extract domain specific construction information. The open-source computational frameworks such as OpenCascade and IfcOpenShell are at a maturity level where they can be readily applied in an industrial setting. The complexity of the IFC schema and these programming tools do result in a considerable. In early 2020 this research project was conceptualized, one and a half year later the implementation is at a level that an evaluation is possible to what extent it can be embedded in the day to day operations of the company.

The variability of IFC data proved to be prohibitive for automated identification of balconies and in some cases also led to errors in the analysis output, in particular the layering of elements without proper decomposition relationships.

At first glance, a Model View Definition or Exchange Requirement for this use case seems feasible as with IsExternal, IfcRelConnects and IfcSlab the basic building blocks for a successful exchange are provided. With all necessary information modelled during export, the analysis task would be reduced to simply extracting explicit information. However, especially the requirement to identify identical balcony slabs poses detailed requirements on how local coordinates of balconies are modelled with respect to their placement, which can likely not dictated in a MVD or even complied with in an authoring tool. In that light, geometric analysis remains necessary.

References

Abanda, F. H., Tah, J. H. M. & Cheung, F. K. T. (2017). BIM in off-site manufacturing for buildings. *Journal of Building Engineering,* 14, 89-102

Babič, N. C., Podbreznik, P. & Rebolj, D. (2010). Integrating resource production and construction using BIM. *Automation in Construction,* 19(5), 539-543

Bus, N., Roxin, A., Picinbono, G., & Fahad, M. (2019). Towards French Smart Building Code: Compliance Checking Based on Semantic Rules. arXiv preprint arXiv:1910.00334.

Ciribrini, A. L. C., Bolpagni, M. & Oliveri, E. (2015). An innovative approach to e-public tendering based on model checking. *Procedia Economics and Finance.* 21, 32-39

Krijnen, T. & Hesselink, G. (2021). Ifc Building Models for Automated Extraction of Data from Balconies [Data set]. *Zenodo.* http://doi.org/10.5281/zenodo.5047709

Krijgen, T. & Tamke. M. (2015). Assessing Implicit Knowledge in BIM Models with Machine Learning. *Conference: Design Modelling symposium: Modelling Behaviour.* Volume: Modelling Behaviour: Design Modelling Symposium 2015. DOI: 10.1007/978-3-319-24208-8_33

Krijnen, T., Noardo, F., Ohori, K. A., Ledoux, H., & Stoter, J. (2020). Validation and Inference of Geometrical Relationships in IFC. In Proceedings of the 37th International Conference of CIB W78, Sao Paulo.

Luttun, J., & Krijnen, T. (2020). An Approach for Data Extraction, Validation and Correction Using Geometrical Algorithms and Model View Definitions on Building Models. In International Conference on Computing in Civil and Building Engineering (pp. 529-543). Springer, Cham.

Moult, D., & Krijnen, T. (2020). Compliance checking on building models with the Gherkin language and Continuous Integration. In Proceedings of the EG-ICE 2020 Workshop on Intelligent Computing in Engineering. Technische Universität Berlin.

Zhiliang, M.,, Zhenhua, W., Wu. S. & Zhe, L. (2011). Application and exclusion of the IFC standard in construction cost estimating for tendering in China. *Automation in Construction,* 20(2), 196-204