Validation and Inference of Geometrical Relationships in IFC

Krijnen, T.F.; Noardo, F.; Arroyo Ohori, G.A.K.; Ledoux, H.; Stoter, J.E.

Citation (APA)
Krijnen, T. F., Noardo, F., Arroyo Ohori, G. A. K., Ledoux, H., & Stoter, J. E. (2020). Validation and Inference of Geometrical Relationships in IFC. In *Proceedings of the 37th International Conference of CIB W78, Sao Paulo* (pp. 98-111). (ITC Digital Library). https://doi.org/10.46421/2706-6568.37.2020.paper008

Important note
To cite this publication, please use the final published version (if applicable).
Please check the document version above.

VALIDATION AND INFERENCE OF GEOMETRICAL RELATIONSHIPS IN IFC

Thomas Krijnen[1], Francesca Noardo[2], Ken Arroyo Ohori[3], Hugo Ledoux[4], and Jantien Stoter[5]

Abstract: The Industry Foundation Classes are a prevalent open standard to exchange Building Information Models. In such a model, geometric representations are provided for individual building elements along with semantic information, including a significant amount of properties related to geometry and explicit topological relationships. These relationships and quantities introduce redundancies and often inconsistencies as well. Moreover, they introduce complexity in downstream processing. Combining multiple aspect models into a single model has non-trivial consequences for the connectivity graphs. Programmatic mutations are complicated because of the relationships that need to be updated as a result of changes.

In order to alleviate these issues, this paper provides a theoretical framework and implementation for both validating and inferring semantic and topological constructs from the geometric representations, rooted on Egenhofer spatial predicates and extended with the IFC modelling tolerance. Combining these two concepts, wall connectivity is equivalent to the intersection of the wall representation boundaries, where a boundary is not a surface, but rather a hollow solid with a thickness derived from the modelling tolerance.

The algorithms presented in this paper are implemented in fully open source software based on the IfcOpenShell software library and the CGAL computational geometry library using Nef polyhedra. We provide a formalization of space boundaries, spatial containment and wall connectivity relationships. The validation and inference rules are applied to a public set of building models. We conclude that exported models have geometric flaws and that several relationships can indeed be inferred by means of generic geometric intersection logic.

Keywords: BIM, IFC, Geometry, Validation

1 INTRODUCTION

1.1 Industry Foundation Classes

The Industry Foundation Classes (IFC) are a prevalent open standard to exchange Building Information Models (BIM). In such a model a geometric representation is provided for individual building elements along with user-extensible and multi-disciplinary semantic information. The geometrical definitions by which representations

---

[1] Postdoctoral researcher, Delft University of Technology, Delft, The Netherlands, t.f.krijnen@tudelft.nl
[2] Postdoctoral researcher, Delft University of Technology, Delft, The Netherlands, f.noardo@tudelft.nl
[3] Postdoctoral researcher, Delft University of Technology, Delft, The Netherlands, k.ohori@tudelft.nl
[4] Associate professor, Delft University of Technology, Delft, The Netherlands, h.ledoux@tudelft.nl
[5] Professor, Delft University of Technology, Delft, The Netherlands, j.e.stoter@tudelft.nl

can be constructed are derived from existing standards for exchanging Computer Aided Design (CAD) geometries and enriched with domain specific parametric cross section profile definitions and more. Coming to a full evaluation of the building element geometries requires a considerable software implementation effort and is computationally intensive due to complex sweep operations and Boolean operations.

In order to make those models useful to a high number of applications, a significant amount of properties and explicit topological relationships are provided by the IFC authoring applications (see Figure 1 for a hierarchical overview of all IFC relationships). These include spatial containment relationships, wall connectivity relationships, space bounding interfaces and quantities for surface areas and volume. It is essential these to be explicitly specified, in particular for importing data into applications that are unable to evaluate computational geometry, for example spreadsheet-based facility management applications. On the other hand, simulations and interoperability with Geographical Information Systems (GIS) may pose strict requirements on geometric and topological validity. For example, boundary interfaces are vital for thermal simulation (Bazjanac 2010) and connectivity information is vital for evacuation analysis. Without this information these analyses cannot be executed directly on BIM data.

However, these explicit relationships and quantities by their nature introduce redundancies between the semantic and geometrical constructs and often introduce inconsistencies as well. Moreover, they introduce further complexity in downstream processing of IFC building models. In particular, combining multiple aspect models into a single model has non-trivial consequences for the connectivity graphs if such relationships are not well stated and understood. There is also a fundamental problem with these relationships and the file-based nature of data exchanges in the industry. First of all, IFC (due to the EXPRESS legacy) are essentially information silos, references can be made within a file, but not easily to other files. The usage of aspect models (subsets of the overall model relevant to a specific discipline) is essential in industry to allow practitioners to use bespoke software for their expertise. Therefore, the amount of relationships that can be expressed when using aspect models is limited to only the elements in scope of that discipline. To exemplify: when one aspect model contains the spaces and another aspect models the walls and slabs, it is impossible for an authoring tool to provide the space boundary relationships.

While traditionally information exchanges in the construction industry have been predominantly file-based, more and more initiatives arise that aim to exchange incremental model subsets, partial exchanges or do content negotiation based on the possibilities of the importing and exporting side. These more interactive and collaborative exchanges are considered part of the higher BIM levels 1 and 2 in industry. Examples of this are the BIM Collaboration Format (BCF) BIM Snippet (see Lindhard and Steinmann (2014) for example, that advocates transferring isolated elements because of the dense relationship graphs otherwise) and the OpenCDE API[6]. These kind of incremental exchanges, additions and modifications are much easier on a "bag of elements" model, similar to a dataset of GIS features, where relationships are not stored explicitly but computed on the fly where needed.

---

[6]   https://github.com/buildingSMART/OpenCDE-API

Figure 1. Hierarchical overview of relationships in the schema. IfcRel- prefixes have been removed for brevity. Many are related semantic constructs such as IfcRelDefines subtypes. A significant amount of the IfcRelConnects subtypes are related to proximity, containment and neighborhood and can likely be geometrically inferred.

1.2 BIM and computational geometry

Geometry is not the only aspect conveyed in BIM models, but it's an important part and one of the most complex parts of the schema. Many issues with interoperable IFC usage stem from geometrocal issues, as indicated in literature (Feringa and Krijnen 2015; Arroyo Ohori et al. 2018; Solihin et al. 2015). Also, many of the challenges on multidisciplinary usage of BIM models are related to geometry, such as conversion to CityGML and thermal analysis.

The geometry resources are one of the largest and more complicated subschemas of the IFC specification. In the current IFC4X1 schema there are 153 subtypes of IfcRepresentationItem and 22 subtypes of IfcProfileDef. This indicates the implementation effort required to implement full geometric support for the IFC schema. Most of these geometric definitions are derived from ISO 10303 42, but over time bespoke extensions have been added, such as the domain-specific parametric profile definitions, tapered extrusions and revolutions and more efficient ways to encode tessellations and piecewise connected curves.

In addition to the amount of geometry definitions, for the purpose of this research, the most influential characteristic is the implicitness of the geometry definitions. In

contrast, many GIS standards use explicit representations where (maybe besides a transformation for the Coordinate Reference System) the eventual location and shape of the feature can be read from the data without much computation. Instead, IFC has a placement hierarchy that matches the project decomposition structure so that every element geometry is defined within its own Local Coordinate System. Furthermore, there is an extensive amount of processing required for sweeps and Boolean operations, defined in a typical IFC file, to be visualized.

A typical IFC file contains individual solid volume geometries for the respective building elements. However, for other use cases, such as thermal simulation, another view on geometric data is required where pairs of neighboring spaces are connected by means of a single thin-walled interface. Provisions for this are incorporated in the IFC schema by means of the IfcRelSpaceBoundary (a subtype of IfcRelConnects).

There are various geometric models that can be used underlying an implementation of the geometric resources in IFC. A semantically good match is found in the Boundary Representation (BRep) model. It consists of a hierarchy of topological entities: Solid > Shell > Face > Loop > Edge > Vertex. Face, Edge and Vertex have associated geometric components: an underlying Surface, Curve and Point. At every level a tolerance factor is applied so that, for example, a Vertex point can be some small distance away from the Edge curve that is connected to the Vertex. This tolerance is necessary because floating point arithmetic on modern computer hardware is intrinsically imprecise. The tolerance value is also necessary because some solutions to geometric intersections are inherently imprecise. When operations are applied to BReps the tolerance increases as the uncertainty of the operands is added together. This can result in situations where the tolerance value is on the same scale as the geometric detail and a consistent separation between interior and exterior is no longer provided by the boundary. The shape has then become invalid. This will likely happen when chaining many operations or when using many operands. An open source implementation of the BRep model and associated algorithms can be found in Open CASCADE[7]. IfcOpenShell[8] is a software library that implements support for the geometrical resources in IFC and is based on Open CASCADE.

Other implementations of IFC operate internally on triangle meshes or polyhedra. CGAL (Fabri et al. 2000) is a software library for computational geometry that offers higher precision number types that are implemented in software. In addition, by restricting operands to polyhedra, it's possible to guarantee exact constructions. Nef polyhedra (Bieri and Nef 1988) are closed under Boolean set operations, which is not the case for the BRep model described above. Hence, further processing on Nef polyhedra in CGAL is much more robust and precise. This at the expense of less direct implementation available for the kind of advanced sweep primitives in IFC and curved surfaces can be represented only as faceted approximations.

1.3   Structure of this paper

In the following section, the current state of the art with respect to validation and geometrical processing is outlined. Following that, we provide a formalization of spatial containment, wall connectivity and space boundary relationships. The validation and inference rules are tested on a public set of building models and the results are listed. A

---

[7]   https://www.opencascade.com/
[8]   http://ifcopenshell.org/

conclusion is provided that reflects on the presented work and provides an outlook for further research.

2   STATE OF THE ART

A theoretical framework for assessing the validity of an IFC file is provided in Solihin et al. (2015). It explains how errors are introduced at human modelling, in exporting to IFC and in the importing application. Indeed, geometrical and topological errors are referenced as a main source of errors in IFC files and the research includes several examples encountered in models. It does not provide a geometrical formalization of these errors.

Within the context of a reinterpretation of IFC data as CityGML models, Arroyo Ohori et al. (2018) discusses how geometrical and topological errors in IFC models impact further automated handling. The focus here was on errors within individual element geometries that are clear violations of the informal propositions in the IFC schema such as non-planar faces and self-intersections of face boundaries and shells.

A framework for validation of GML primitives is provided in Ledoux (2018). The set of geometric primitives in GML is rather different from those in IFC. However, what is included is a check on element disjointedness and containment similarly to how spatial containment is modeled in IFC and geometrical interference is tested on BIM models in practice, using the "clash detection" algorithms widely implemented in BIM-based coordination tools.

An algorithm to automatically calculate second level space boundaries is provided in Lilis et al. (2017) consisting of four stages: (a) parsing, (b) geometry interpretation into a BRep, (c) boundary overlap and (d) boundary projection. Detection and fixes for a set of geometric modelling issues including interfering geometries (clashes), space definition errors where space geometry does not fill the complete bounded area, surface orientation and incomplete shells is provided in Lilis et al. (2015).

Another algorithm to compute second level space boundaries and a conversion to the file format of Open Studio for thermal analysis is provided in El-Diraby et al. (2017). The steps of the algorithm are different from Lilis et al. (2017). In El-Diraby et al. (2017) the solid volumes of the bounding elements are first flattened into thin-walled surfaces and then broken into segments so that a manifold composite solid of space boundaries is obtained.

Egenhofer operators (Egenhofer and Franzosa 1991) find their origin in the GIS domain. They provide a formal approach to qualify the spatial relationship between two geometries as a 3x3 matrix of intersections of the Interior, Boundary and Exterior of both operands. By incorporating the dimensionality (point, line, area) of the resulting intersection this is extended this into the Dimensionally Extended 9 Intersection Model (DE-9IM) (Clementini et al. 1993). While typically applied to two-dimensional geometries, these can be applied to three-dimensional operands as well (Zlatanova 2000) and have been applied to a BIM query interface (Daum and Borrmann 2014).

3   CONTRIBUTION

In this paper we present a generic formalization and corresponding implementation of several geometric predicates to both validate IFC data and infer relationships when missing. The implementation of all predicates is based on the geometric intersection of

two operand shapes. In the paper, these are listed based on the increasing order of complexity. The implemented predicates are:

- Containment in building storeys; operates on IfcBuildingStorey × IfcElement. Operand interiors are intersected and the calculated volume of the intersection is computed and compared.

- Wall connectivity; operates on all unique unordered pairs ∈ IfcWall × IfcWall. Intersection is computed on a thin volume around the boundary of the elements and the intersection geometry is projected onto a linear representation of the wall to qualify the connection type (at start, at end, at path).

- First level space boundaries; operates on IfcSpace × IfcElement. Intersection is computed on a volume around the boundary of the elements. The intersection result is flattened into a thin-walled surface and written to the resulting IFC file.

The checks and inferences above are implemented on a novel version[9] (branch) of IfcOpenShell where CGAL is used for geometry interpretation in addition to Open CASCADE. Work on this alternative version started as part of the project described in Arroyo Ohori et al. (2018). The checks are implemented using a Minkowski sum (minkowski_sum_3()) between the interpreted IFC geometries and a padding volume (the current prototype uses a cube with the size of the IFC modeling tolerance, but this is configurable) that is used to convert the surface boundary into a solid. The element to element intersection operation is sped up by finding candidate pairs by an initial bounding box search (box_self_intersection_d()).

3.1   Spatial decomposition

The spatial decomposition structure is currently restricted to a tree in IFC. The requirement of a single containing element implies that elements need to be broken up when they are part of multiple building storeys. This does not always match the construction method on site, creates discrepancies between the native and exported model and adds additional complexity to the IFC export process. Therefore, more adaptive assignment to storeys based on preferences by the importing user might be beneficial. The work in this paper can be explained in that regard.

An IfcBuildingStorey does not typically have an associated representation, so a solid volume has to composed based on the 'IfcBuildingStorey.Elevation' attribute. A solid volume (represented as a point set $\langle x, y, z \rangle$ defined in the Cartesian coordinate system of the model) representing the storey envelope is composed, infinite in X and Y directions and bounded over the Z-axis according to definition (1) below:

$$v_{storey} := \langle x, y, z \rangle \; if \; (z \geq storey.Elevation - 0.3m \lor is\_first(storey)) \land$$
$$(z < next(storey).Elevation + 0.3m \lor is\_last(storey)) \#(1)$$

Where next(), is_first() and is_last() are based all IfcBuildingStoreys in the model, sorted based on their Elevation attribute.

Since storey elevation is typically set at the top of the finished floor height and the floor slab underneath is to be included in that storey we have set 0.6m vertical overlap

---

[9]   https://doi.org/10.5281/zenodo.3633653

between the deduced building storey volumes. Storey containment relationships are marked invalid when there is another building storey shape with a larger total overlap with the element shape.

3.2   Wall connectivity

Wall connectivity information in BIM models describes how walls can move with respect to each other and how the wall layers in case of a compound structure are folded to provide a watertight shell. Wall connectivity can be useful to find cycles of connected loops and the spaces that bound them, but since spaces can be modelled explicitly and metadata can be attached only when modelled explicitly, there is little use for deriving space geometry from bounding wall cycles. See Fig. 2 for a graphical representation of how the spatial predicate is implemented.

Fig. 2. Graphical depiction of an example wall connectivity intersection operation with two wall "Body" geometries (rectangular); their respective wall "Axis" (a line from square to pointer arrow head); the computed boundary (dotted); boundary intersections (gray); and projections of boundary intersection indicated as narrow rectangle over extended wall "Axis".

Wall connectivity in IFC is only provided in the Z-plane. The generic intersection approach in this paper will also detect pairs of walls stacked vertically. In that case the intersection volume is a horizontally oriented narrow slab. These cases are filtered by looking at the vertical extent of the intersection volume. Similarly, sometimes wall elements are modelled as a form of covering of other wall elements so that they are touching along their longitudinal face. Also in this case no connection relationship should be written. This second exception is filtered by comparing the surface area of the intersection with the surface area of the operands. When area of intersection is close to area of smallest wall the pair of elements is considered not to be connected.

Note that the current implementation of this check assumes that the wall axis begin and end points are aligned with the body representation. An additional check can be in place to verify this.

3.3   Space boundaries

Similar to the wall connectivity check, space boundaries are relationships between elements where there is a geometric intersection between the boundaries of the element volumes. In case of the wall connectivity the candidates were of the same type, in this case however, the check still operates on pairs of elements, but exactly one of the elements need to be a space. In addition, since doors and windows are often a bit inset into the wall surface, they do not directly touch the neighboring space. Because of that, for windows and doors, the opening geometry is to be used instead. Since the opening geometry is often offset a bit from the wall surface to prevent precision issues when applying the Boolean operation, the intersection between wall and opening has to be used to remove this offset. This is currently not implemented yet in the validation prototype. For this paper only wall and slab boundaries are checked and inferred.

In this version of the prototype only first-level space boundaries are implemented. These are relationships between spaces and the elements that bound them. Second-level space boundaries are relationships between pairs of neighboring spaces with a uniform boundary element. In Fig. 3 a graphical sketch is provided of how the functionality in this paper can be extended for second-level space boundaries by using a variable padding derived from wall thicknesses.

The inferred intersection shapes can be written to the file, but have to be flattened into a single surface. The result from the Nef polyhedron intersection is a solid volume.
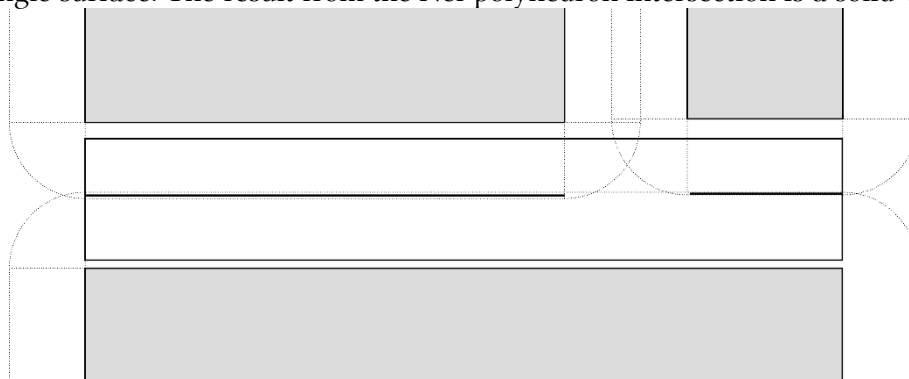

Fig. 3. Inference of second level space boundaries by including half of the thickness of wall elements into the padding operation and computing overlap of padded IfcSpace pairs. Space geometry in solid gray; offset boundaries (dotted); and inferred second level boundaries in black line; Note that this is somewhat speculative as this has not been implemented yet as part of this paper.

4   EVALUATION

4.1   Models

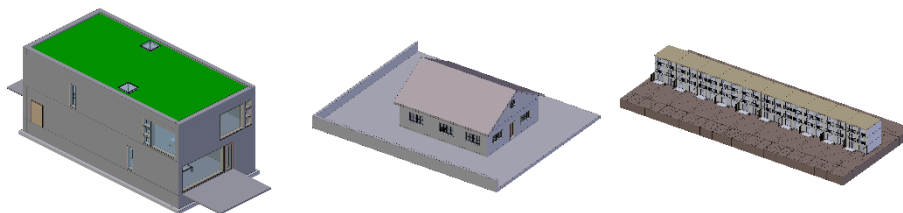A limited set of IFC building models is assessed for this research, presented in Fig. 4.

Fig. 4. Three models assessed in this paper: Duplex[1], Fzk[2], and Smiley[3]; [1] IFC2X3; Autodesk Revit Architecture 2011; U.S Army, Corps of Engineers [2] IFC2X_FINAL; Autodesk Architectural Desktop; Automation and Applied Informatics (IAI) / Karlsruhe Institute of Technology (KIT) [3] IFC4; Archicad 20; IAI / KIT

4.2 Results

<u>Storey containment</u>. The storey containment check is the most elementary in this paper and also fairly trivial to verify manually by toggling the visibility of building elements storey children in an IFC viewer. If elements are wrongfully located it will likely also negatively impact how the model is handled in the original BIM authoring application, given that these tools often provide a plan-based modelling view to the user. Only the Duplex model reports issues for this check and actually in Solihin et al. (2015) this issue on this exact model is reported as an indication of IFC model quality. The misplaced walls are shown visually in Fig. 5.
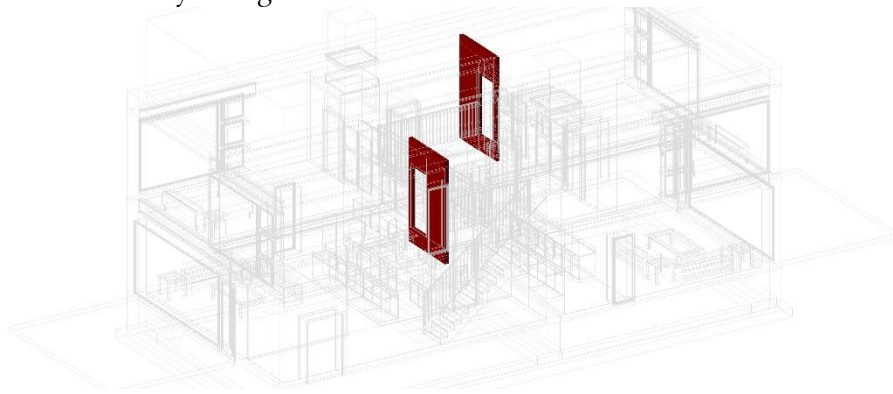

Fig. 5. Only the duplex model faces issues with elements assigned to the wrong storey. The two elements are identified by GlobalId 2O2Fr$t4X7Zf8NOew3FL8v and 2O2Fr$t4X7Zf8NOew3FL8v.

<u>Wall connectivity</u>. In Table 1 the results of the wall connectivity check on three models are presented. Three kinds of errors can be detected (a) a connection relation can be missing for neighbouring elements (b) a connection relation can be present but qualified with the wrong connection type and (c) a connection can specified for non-neighboring elements. All these three cases are detected, only the Smiley model was free from errors of this kind. Visual depictions of the cases are presented for the Duplex model in Fig. 6, Fig. 7 and Fig. 8.

Table 1. Wall connectivity errors reported for the three assessed models.

	Duplex	Fzk	Smiley
missing relation	2	2	0
invalid relation: wrong connection type	4	6	0
invalid relation: walls not touching	8	0	0

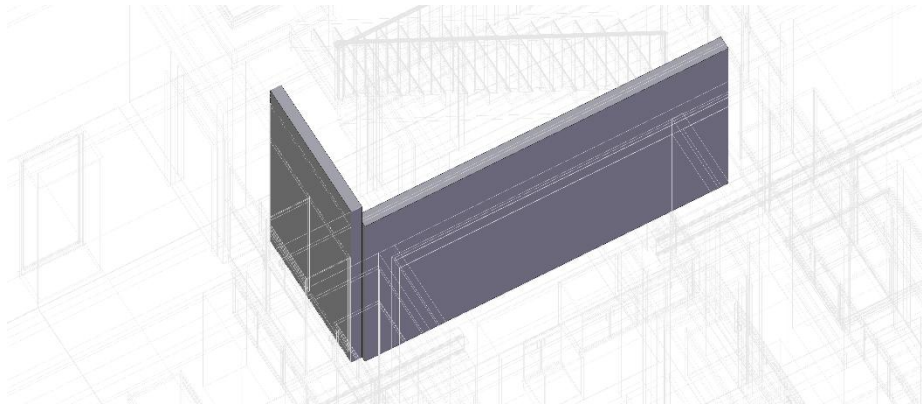Thomas Krijnen, Francesca Noardo, Ken Arroyo Ohori, Hugo Ledoux and Jantien Stoter

Fig. 6. Example of wall connectivity information in the duplex model where the wrong connection type is supplied. One of the walls is said to be connected ATPATH where both walls clearly touch at their respective begin or end points. The two walls are identified by GlobalId 2O2Fr$t4X7Zf8NOew3FLKI and 2O2Fr$t4X7Zf8NOew3FLKI.
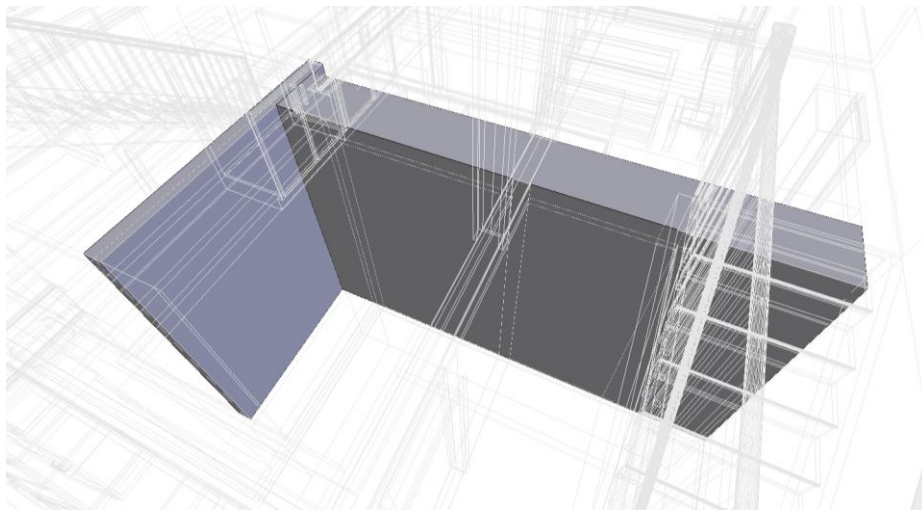

Fig. 7. Example of two touching walls in the Duplex models for which no connectivity information is provided in the model. These two walls are identified by GlobalId 2O2Fr$t4X7Zf8NOew3FKRi and 0iEHWY1$XA8eQeeULq4jDb.
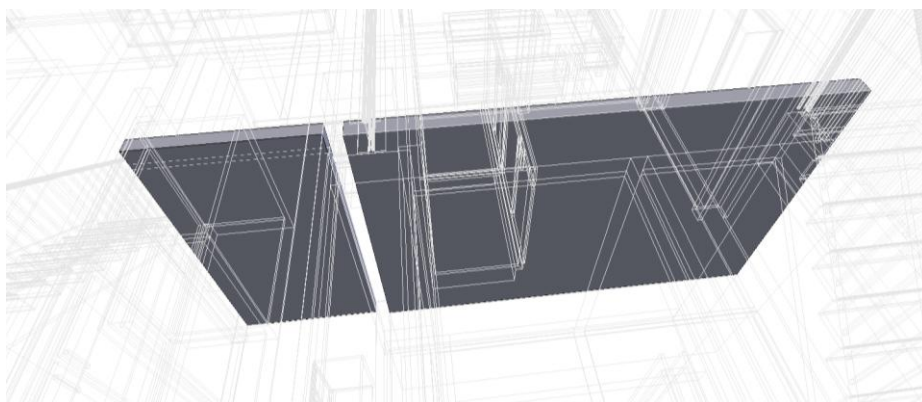
Fig. 8. Example of two walls in the Duplex model that are not touching but for which connection information is provided regardless. These two walls are identified by GlobalId 2O2Fr$t4X7Zf8NOew3FKIu and 0iEHWY1$XA8eQeeULq4jE6

Space boundaries. Space boundary geometry is not typically directly visible in common BIM visualization or coordination platforms. Wrong space boundaries are shown in Fig. 9 for the Duplex model and in Fig. 10 some of the missing space boundaries are located in the Smiley model. The version of the Fzk model used did not have any space boundaries.

Fig. 9. Wrong space boundary connection geometries in red for the duplex model. Some boundaries are misplaced. Some boundaries do not have the appropriate geometry. Strangely, some space boundary geometry consists of wall footprints.
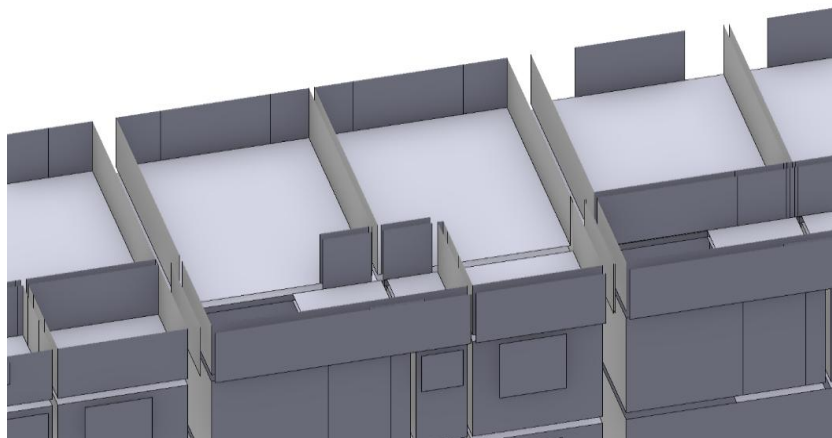

Fig. 10. Visualization of the space boundaries embedded in the Smiley model. The tool developed for this paper reports 10 missing space boundaries, with some of these are located on the exterior (right side of the image); some located on the interior (center)

4.3   Performance

Performance optimization has not been part of this initial research prototype. Also, the IFC software library used, IfcOpenShell, favours reliability over performance so quite a bit intermediate validity checks and fixes are applied at the expense of performance on valid input. To give an indication of the processing time in the current state the following times are listed, broken down into the performed steps.

New in the version of IfcOpenShell used in this paper is a two-phase geometry processing routine, where first the IFC instance types are mapped to generic geometrical and topological definitions agnostic of IFC data and afterwards the reinterpretation in CGAL or Open CASCADE happens. These are steps (b) and (c).

Results presented in Table 2 are averages of CPU time on five runs of the wall connection check on the Duplex model on a HP Spectre x360 Convertible 13-ae0xx laptop with an Intel(R) Core(TM) i7-8550U CPU and 16.0 GB RAM on an executable compiled with Microsoft Visual Studio 2017 version 15.9.11 with optimization level /O2.

Table 2. Processing times for the wall connectivity check on the Duplex model.

Task	Time (s)
a. IFC parsing (tokenization and storage)	0.7174
b. Geometry mapping	0.0364
c. Geometry creation	0.5732
d. Conversion to Nef polyhedron	1.4116
e. Minkowski sum with small cube	7.0064
f. 3D Box intersection pairs	0.0006
g. Boolean intersection on Nef	16.5032
h. Conversion of result back to polyhedron	1.5342

While performance has not been a focus of this initial implementation the timings do show that the current approach of validation adds a significant additional processing time on top of normal visualization workflows where, after (c) and probably a triangulation step, the results can be visualized to an end-user. Since especially the usage of Nef polyhedra adds a significant time penalty, it can be investigated whether regular CGAL polyhedra can be used for the distance query to further reduce false negatives reported by the bounding box intersection. Also, we can investigate whether we can make use of the CGAL simple Cartesian kernel with regular floating-point numbers with full implementation in hardware and only switch to the exact predicates kernel, which is required by Nef, when needed. The Minkowski sum can be eliminated for many geometrical instances when taking the padding into account during construction. For example, an IfcExtrudedAreaSolid of an IfcRectangleProfileDef - which is a very prevalent definition for wall elements - can be constructed with padding by augmenting the rectangle dimensions and extrusion depth.

Note that the geometry generation in (c) only includes wall elements and has opening subtractions disabled (as window and door openings will not impact connectivity). Wall elements are typically simple extrusions that are efficient to process. When evaluating elements with more complicated geometry and openings are included,

step (c) will be more significant with respect to the overall processing time of the check and inference computation.

5   CONCLUSION

The aim of this paper was twofold. On the one hand, concretely, a set of validation algorithms has been presented that can be applied to IFC models from practice. On the other hand, we also aim to show that, by providing functionality to infer relationships, a fair amount of the explicit relationships in the IFC schema can be eliminated as they can be inferred using computational geometry. This leads to a lighter schema and a much less dense instance graph in population models. That last notion will help in establishing transactional exchanges of data subsets and interfaces for data extraction and modification.

The validation algorithms show that indeed a fair amount of geometrical relationships are invalid on the limited set of example models. This will not come as a surprise to end-users that are often affected by the limited amount of trust that importing and exporting parties put in the exchanges. This leads to frustration and costly rework. A thorough analysis of building models using a wider set of validation checks will be provided as part of ongoing research.

Further research will be directed at inferring more diverse relationships in IFC. It is believed that the generic approach used in this paper can be applied at additional relationships.

The implementation using CGAL proved to be a useful counterpart to the existing implementation of geometry handling in IfcOpenShell based on Open CASCADE. Further research will be directed at creating a hybrid implementation that can select the most appropriate implementation based on geometry forms and end-user preferences. The performance overhead of running the checks is currently significant, but, by incorporating padding during construction and only use Nef polyhedra and the exact kernel where necessary, likely a lot of the overhead can be eliminated.

6   REFERENCES

Arroyo Ohori, K., Diakité, A., Krijnen, T., Ledoux, H., Stoter, J. (2018). Processing BIM and GIS models in practice: experiences and recommendations from a GeoBIM project in the Netherlands. ISPRS International Journal of Geo-Information, 7(8), 311.

Bazjanac, V. (2010). Space boundary requirements for modeling of building geometry for energy and other performance simulation. In CIB W78: 27th International Conference.

Bieri, H., Nef, W. (1988). Elementary set operations with d-dimensional polyhedra. In Workshop on Computational Geometry, pp. 97-112. Springer, Berlin, Heidelberg.

Clementini, E., Di Felice, P., Van Oosterom, P. (1993). A small set of formal topological relationships suitable for end-user interaction. In International Symposium on Spatial Databases, pp. 277-295. Springer, Berlin, Heidelberg.

Daum, S., Borrmann, A. (2014). Processing of topological BIM queries using boundary representation based methods. Advanced Engineering Informatics, 28(4), pp. 272-286.

Egenhofer, M. J., Franzosa, R. D. (1991). Point-set topological spatial relations. International Journal of Geographical Information System, 5(2), pp. 161-174.

El-Diraby, T., Krijnen, T., Papagelis, M. (2017). BIM-based collaborative design and socio-technical analytics of green buildings. Automation in Construction, 82, 59-74.

Fabri, A., Giezeman, G. J., Kettner, L., Schirra, S., Schönherr, S. (2000). On the design of CGAL a computational geometry algorithms library. Software: Practice and Experience, 30(11), pp. 1167-1202.

Feringa, J., Krijnen, T. (2015). BIM for robotic manufacturing. In: Proceedings of the International Association for Shell and Spatial Structures (IASS) Symposium, 17-20 August 2015, Amsterdam, The Netherlands.

Ledoux, H.: val3dity (2018). Validation of 3D GIS primitives according to the international standards. Open Geospatial Data, Software and Standards, 3(1), 1.

Lilis, G. N., Giannakis, G. I., Rovas, D. V. (2015). Detection and semi-automatic correction of geometric inaccuracies in IFC files. In 14th International Conference of IBPSA-Building Simulation 2015, BS 2015, Conference Proceedings, pp. 2182-2189.

Lilis, G. N., Giannakis, G. I., Rovas, D. V. (2017). Automatic generation of second-level space boundary topology from IFC geometry inputs. Automation in Construction, 76, pp. 108-124.

Lindhard, K., Steinmann, R. (2014). BIM-collaboration processes-from fuzziness to practical implementation. In: European Conferences On Product And Process Modeling In The Building Industry (ECPPM), pp. 27-32.

Solihin, W., Eastman, C., Lee, Y. C. (2015). Toward robust and quantifiable automated IFC quality validation. Advanced Engineering Informatics, 29(3), 739-756.

Zlatanova, S. (2000). On 3D topological relationships. In: Proceedings 11th international workshop on database and expert systems applications, pp. 913-919.