Towards an integration of GIS and BIM data: what are the geometric and topological issues?

Ken Arroyo Ohori   Filip Biljecki   Abdoulaye Diakité
Thomas Krijnen   Hugo Ledoux   Jantien Stoter

This is an author's version of the paper. The authoritative version is:

Towards an integration of GIS and BIM data: what are the geometric and topological issues?. Ken Arroyo Ohori, Filip Biljecki, Abdoulaye Diakité, Thomas Krijnen, Hugo Ledoux and Jantien Stoter. In M. Kalantari and A. Rajabifard (eds.), *12th 3D Geoinfo Conference*, ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences IV-4/W5, ISPRS, Melbourne, Australia, October 2017, pp. 1–8. ISSN: 2194-9042 (Print), 2194-9050 (Internet and USB).
DOI: 10.5194/isprs-annals-IV-4-W5-1-2017

Related source code is available at:
`https://github.com/aothms/IfcOpenShell_CGAL`

Geographic information and building information modelling both model buildings and infrastructure, but the way in which they are modelled is usually complimentary and BIM-GIS integration is widely considered as a way forward for both domains. For one, more detailed BIM data can feed more general GIS data and GIS data can provide the context that is necessary to BIM data. While previous studies have focused on the theoretical aspects of such an integration at a schema level, in this paper we focus on explaining the geometric and topological issues we have found while trying to develop software to realise such an integration in practice and at a data level. In our preliminary results, which are presented here, we have found that many issues for such an integration remain: handling the geometric and topological problems in BIM models, dealing with bad georeferencing and the figuring out the best way to convert data between IFC and CityGML are all open issues.

1 Introduction

The geographic information science (GIS) domain describes information about the environment, mainly as it currently exists, while the building information modelling (BIM) domain focuses on information about the design and construction of building sites. Both domains however overlap when their data concerns information about infrastructure, buildings, floors, and rooms. Because of the overlap, it is widely acknowledged that the integration of data from both domains is beneficial and a crucial step forward for future 3D city modelling as well as for facing the multidisciplinary challenges of our built environment. More detailed BIM data can feed more general GIS data and GIS data can provide the context that is usually missing in BIM data. Liu et al. [2017] describe seven applications for which the integration is a necessity including: 3D cadastres, heritage management and urban environment analysis.

The studies on the exchange of BIM and GIS data often focus on the two most prominent open standards in the two domains: the *Industry Foundation Classes* (IFC) for the BIM domain [ISO, 2013; Building SMART International, 2013], and the OGC standard *CityGML* for the GIS domain [Open Geospatial Consortium, 2012]. IFC models represent the physical elements of single constructions in great detail, and CityGML models represent entire cities in a simpler format that is usable for exchange, dissemination and spatial analyses, such as solar potential and energy consumption estimations.

The two modelling paradigms embodied by IFC and CityGML are representative of BIM and GIS data in general, and they are both widely used in their respective domains. However, the two standards are still broadly incompatible as they originally targeted different applications and different professionals, and they usually cannot be used in the same tools.

The ambitions of integration are getting bigger now that the two domains are increasingly intersecting: BIM methodologies are applied to infrastructural works, city models are getting more detailed; smart city concepts ask for an integrated reasoning on our city infrastructure; and objectives towards sustainability urge for an approach on multiple levels of detail supporting the complete life cycles of objects. Yet, the disciplines of GIS and BIM are disconnected by their modelling paradigms and software tools, which differ fundamentally with respect to their semantics, geometry and level of detail.

Many researchers and practitioners have studied how to best share information between BIM (IFC) and GIS (CityGML) and how to address all the differences, but it is still very hard to share 3D information among different users throughout the life cycle of urban and environmental processes, i.e. from plan, design and construction to maintenance. In addition, because of the different modelling approach of both information models, there is not one optimal nor uniform conversion.

For a successful integration of GIS and BIM data, we consider that two aspects are important. First, data experts from both domains should be involved in the integration endeavour in order to deeply understand the integration issues. Most researchers are experts in one of the domains; rarely in both. The second aspect is to have a close involvement of users and use cases to assure that the solution is more than an academic or theoretical standardisation exercise. Applications from practice should define which BIM data is exactly needed in GIS applications and vice versa, and finally the integration should be implemented accordingly. To make further steps in the BIM and GIS integration, we started a joint research project called *GeoBIM* which takes these two aspects into account.

The partners in the project are: GIS/CityGML experts from TU Delft, BIM/IFC experts from TU Eindhoven, Geonovum as the Dutch national organisation for GIS data standardisation, BIM-loket as the Dutch national organisation for BIM standardisation, and several users: the Cities of Rotterdam and The Hague as well as *Rijkswaterstaat*, the Dutch government entity responsible for public works and water management.

The project is an experiment- and use case-driven scoping study and has two aims: (1) to develop a CityGML/IFC interface for reusing GIS data in the BIM domain and vice versa and (2) to formulate recommendations for further integration, such as modelling guidelines for bidirectional integration based on the main issues identified and the preferred solutions to these.

The CityGML/IFC interface to be developed is an open source library[1] that can read and write both IFC and CityGML, integrate multiple models into one, and can then perform operations (e.g. Boolean set union or intersection operations) to manipulate and analyse the models. The interface should be able to represent the generalised IFC structure in CityGML for GIS applications and the GIS data restructured into IFC for BIM applications. It should also cover the most important geometric and semantic classes of IFC and CityGML.

The developed interface is first applied to three use cases. These are:

Use case 1 The process of submitting an IFC model to a building permit-application portal by citizens and companies; checking the IFC design against the existing physical world (represented in a 3D city model) and against a 3D zoning plan; and finally updating the 3D city model by integrating the 3D building model.

Use case 2 The process of supporting the life-cycle of objects with a continuous information chain: using information about complete urban areas in the design process (i.e. using geoinformation in a BIM application) and, at a later stage, converting plan, design and construction data to maintenance data. The focus of this second use case is on large infrastructure projects.

Use case 3 Integration of sub-surface information in the BIM design process.

Based on the three use cases, recommendations will be formulated for both developments of techniques and standardisation for further integration.

The project started in March 2017 and has until now focused on developing code to read and process IFC models, experimenting with the IFC models available from the users in the project and making an inventory on how designers and constructors need 3D geoinformation in their applications and vice versa, i.e. how GIS professionals can be best served with BIM data.

This paper presents the intermediate results of the project which provided already some detailed insights into the issues of integrating GIS and BIM data in practice.

2 Modelling buildings in GIS and BIM: CityGML and IFC

Although BIM and GIS have a common interest in modelling certain object types, they fundamentally differ in their encoding, their use of geometry and semantics, as well as their level of detail. An integration between BIM and GIS that is useful in different application should acknowledge these differences, which are highlighted in the following sections.

2.1 CityGML

CityGML [Open Geospatial Consortium, 2012] is the most prominent standard to store and exchange 3D city models with semantics in the GIS domain. It presents a structured way to describe the geometry and semantics of topographic features such as buildings and roads. CityGML as a data format is implemented as an application schema for the Geography Markup Language (GML)[2] [OGC, 2004].

CityGML contains a small number of classes structured into 12 modules, most of which are meant to model different

[1] The source code is published at `https://github.com/aothms/IfcOpenShell_CGAL/`

[2] CityGML uses version 3.1.1 of GML

types of objects (e.g. Building, Bridge, WaterBody). These classes differ in the way objects are structured into smaller parts and the attributes that are expected for each. However, CityGML geometries are essentially the same for all classes: objects are represented as surfaces embedded in 3D and consist of triangular and polygonal faces.

CityGML supports five levels of detail (LODs). Figure 1 illustrates the five LODs (LOD0 to LOD4):

LOD0 is non-volumetric and is an horizontal footprint and/or roof surface representation for buildings;

LOD1 is a block-shaped model of a building (with an horizontal roof);

LOD2 adds a generalised roof and installations such as balconies;

LOD3 adds, among others, windows, doors, and a full architectural exterior;

LOD4 models the interior of the building, potentially with pieces of furniture (CityGML does not mandate which indoor features need to be modelled, in practice resulting in models with a different granularity [Goetz, 2013; Boeters et al., 2015]).

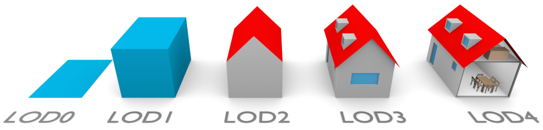


Figure 1: A building represented in LOD0 to LOD4 (image from Biljecki et al. [2016b]).

2.2 IFC

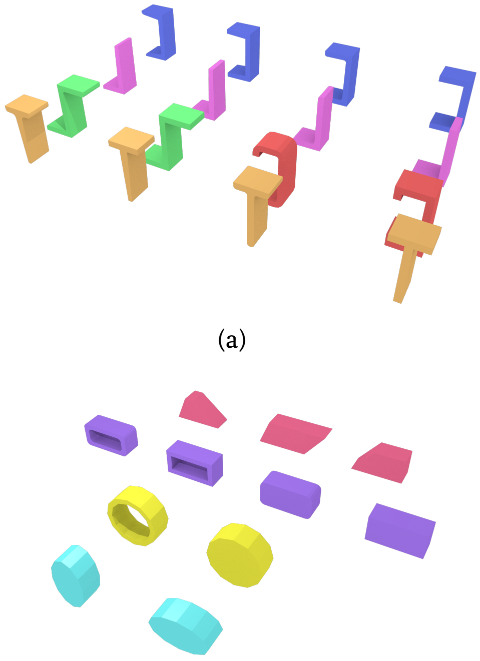
The Industry Foundation Classes (IFC)[3] standard is an open data model used in the Building-information modelling (BIM) domain for the exchange of construction models, often including 3D models of buildings. It has also been adapted as the ISO 16739 international standard [ISO, 2013]. Its geometric aspects are however mostly defined or derived from a different standard, ISO 10303 [ISO, 2014], which also specifies the STEP Physical File (SPF) encoding that is most commonly used in IFC files (`.ifc`).

IFC files can contain many types of classes (130 defined types, 207 enumeration types, 60 select types, 776 entities, 47 functions, and 2 rules in IFC 4 Addendum 2) The geometries in them can use several different representation paradigms which can be combined more or less freely. However, in practice, most IFC objects are built using sweep volumes, explicit faceted surface models and CSG [El-Mekawy and Östman, 2010]. The representation paradigms include:

- **Primitive instancing:** an object is represented based on a set number of predefined parameters. IFC uses this paradigm to define various forms of 2D profiles (Figure 2), as well as volumetric objects such as spheres, cones and pyramids.
- **CSG:** an object is represented as a tree of Boolean set operations (union, intersection and difference) of volumetric objects (see Requicha [1982] for more details). Half-spaces are often used to cut out the undesired parts of surfaces or volumes.
- **Sweep volumes:** a solid can also be defined by a 2D profile (a circle, a rectangle or an arbitrary polygon with or without holes) and a curve [Wang and Wang, 1986] along which the surface is extruded.
- **B-rep:** an object is represented by its bounding surfaces, either triangulated meshes, polygonal meshes or topological arrangements of free-form surfaces.

While GIS data usually rely on geographic coordinate systems that precisely locate the data in the real world, BIM tools often only use local coordinate systems. This is because the latter usually focus on the design aspect of the architectural components at the scale of a single building and such components are usually designed one by

[3] `http://www.buildingsmart-tech.org/specifications/ifc-releases`



(a)

(b)

Figure 2: IFC defines various types of parametric curved profiles such as (a) those based on the characters U, L, Z, C and T and (b) those based on trapezia, (rounded) rectangles, circles with/without holes and ellipses. Note the various types of tapered and curved parts of the profiles. These are most commonly used in extrusions such as those shown here.

one. The direct consequence of such an approach is that BIM data is often not properly georeferenced—something that is only evident when BIM models are loaded into GIS tools. This concern is not new, as it has been pointed out among others in Lapierre and Cote [2007].

In order to consider georeferencing parameters in the exchange process of IFC files, the IFC standard proposed the possibility to store geographic coordinates attached to the `IfcSite` class. The `IfcSite` entity is intended to describe the area where construction works are undertaken. Therefore its geographic location is important and may stand as the link between the local coordinates of a building and a global coordinate system. In that sense, the `IfcSite` class comes along with `RefLatitude`, `RefLongitude` and `RefElevation` attributes [Building SMART International, 2007]. As their names suggest, the two former ones correspond to the latitude and longitude coordinates with respect to the world geodetic system WGS84, and the last corresponds to the datum elevation relative to sea level. There are also attributes dedicated to land title number (designation of the site within a regional system) and postal address of the site (`LandTitleNumber` and `SiteAddress`). The local engineering coordinate system is established by the `IfcGeometricRepresentationContext` entity, which contains a `TrueNorth` attribute that relates the axes of the coordinate system to the geodetic North.

3 Previous integration efforts

The integration of BIM and GIS data is a complex topic that has been tackled in different ways. As Liu et al. [2017] state, the integration can involve semantics and/or geometry, and it can involve the conversion of BIM and GIS data to either a unified model or to the standards of each other (either bidirectional or only one-way). We focus in this paper on realising the integration in practice and at the *data* level. By contrast, Amirebrahimi et al. [2016] identifies how the integration can also be performed at the *process* and *application* levels.

Most of the related work on this topic is concerned with converting IFC models to a GIS model like CityGML because that implies simplifying and removing details and extraneous information in the data. The inverse operation, from GIS data to BIM, is rarely discussed as it is deemed less useful and it might involve adding more details to the data. Nonetheless, there are methods to create BIM from existing models, which can be considered as GIS to BIM; see Volk et al. [2014] for an overview.

El-Mekawy et al. [2011] and El-Mekawy et al. [2012] propose to combine information from both domains and create a unified model in which all the semantic properties of IFC and CityGML are present, and they propose using bidirectional mappings for all the semantic classes relevant to IFC and CityGML. However, they extract the geometries separately from the two models and need manual editing to enrich the result. The resulting unified model does not acknowledge the particularities of the two different communities, and their focus is mostly LOD4 since all the features are mapped. Amirebrahimi et al. [2016] extends the data model of GML (and not CityGML) to create a unified model supporting two specific applications (visualisation and flood damage assessment to specific buildings).

When the geometry is considered, most existing conversion algorithms from BIM to GIS convert *all* the geometries (from one of the 4 paradigms listed in Section 2.2), which yields GIS models having poor usability in practice since using these geometries in software for simulations or spatial analyses requires a huge amount of manual work [**?**]. Little attention has been paid to a more meaningful conversion which requires not only selecting the appropriate classes, but also performing spatial operations (such as Boolean operations) to select only the features, or part of these, that are appropriate in the other model. One concrete example is how in CityGML the buildings in LOD2 and LOD3 should be modelled using only their surfaces that are visible from the outside, while a simple conversion of all the BIM classes (in most cases) will yield walls that have a thickness and thus interior and exterior surfaces. As Benner et al. [2005] remarks, the surfaces forming the exterior of a building are not explicitly marked as such in an IFC file and cannot be deducted directly from the semantics of the objects.

A few programs offer the possibility to convert IFC models into CityGML models. Three examples are: the Building Information Modelserver[4], IfcExplorer[5] and Safe FME[6]. All of them allow the users to convert IFC models to CityGML at different LODs. The users can in some cases choose which IFC objects should be used. However, all the features are converted, without any selection or post-processing to keep only the relevant ones. Different projects use such approach to obtain integrated datasets useful for visualisation-based analysis, e.g. Rafiee et al. [2014]. Observe here that for visualisation converting all the geometries is not a major hindrance since the ones that are indoor will simply be not displayed. The size of the dataset will however increase and slow down the visualisation process.

Hijazi et al. [2010] built an open-source web-GIS in which IFC objects can be imported after having been converted to b-rep models. Unfortunately, only sweep-volume objects can be converted and all the geometries are kept (thus resulting in non-manifold buildings). De Laat and van Berlo [2011] developed an CityGML ADE (application domain extension) called GeoBIM[7], so that new semantic classes defined in IFC are added in a CityGML model. However, no geometric manipulation is performed.

There are different attempts at converting IFC models to CityGML LOD2/3 models by processing the geometries.

Benner et al. [2005] describe the general steps needed to convert an IFC file to an alternative data model closely related to CityGML (the QUASY model). They first map the semantics from IFC to QUASY and select the relevant boundary objects, and then the outer visible surfaces are extracted by selecting a subset of the input objects. For the (equivalent of) the LOD3 model, they discard geometries inside the building by projecting each floor of a building to horizontal and vertical 'footprints' and keeping only those touching the envelope. This technique may yield building having holes/gaps in the exterior envelope. Moreover, while the output models appear to be LOD3 models, the walls and the roof are volumetric. Deng et al. [2016] converts IFC to the different LODs of CityGML. To obtain the exterior envelope of each building, they use a ray-tracing algorithm: they define a few points-of-view and determine whether

[4] http://bimserver.org

[5] http://iai-typo3.iai.fzk.de/www-extern/index.php?id=1566&L=1

[6] http://www.safe.com

[7] http://www.citygmlwiki.org/index.php/CityGML_GeoBIM_ADE

a given surface is visible from them, if so it is assumed to form part of the exterior boundary. This method will however yield buildings with several holes as several surfaces (e.g. those for a roof overhang, or small ones near a window sill) will not be visible from the finite set of points of view. Donkers et al. [2016] convert IFC models to LOD3 models by selecting a subset of the objects and then extracting the exterior envelope by using a series of Boolean set operations in 3D. Their algorithm does not yield holes/gaps if the input does not contain any. However, in their implementation the semantics of the objects can be stored and thus different tricks are used to imply it in the final model.

4 Our data and methodology

We have focused our experiments on three IFC files from the City of The Hague (Figures 3–5). These are complex models with several thousand objects each and which use all the main representation paradigms that are possible in IFC.

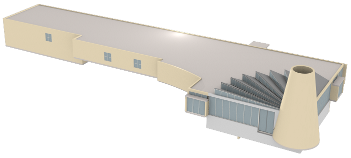


Figure 3: CUVO Ockenburghstraat KOW.

We parse and process every object in an IFC file independently using a modified version of IfcOpenShell[8], where we substituted its openCASCADE[9]-based kernel for a new one based on CGAL. By doing so, we bring geometry from the GIS and BIM domain into a common data environment while maintaining a strict correspondence to the original IFC and CityGML instances.

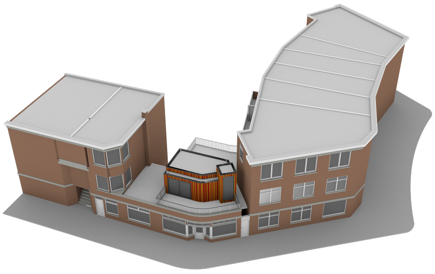


Figure 4: Rabarberstraat 144.

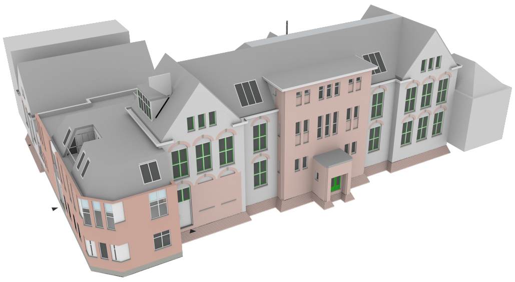


Figure 5: Witte de Withstraat.

As such, we aim to use spatial analysis algorithms from the GIS domain and the robust Boolean set operations on Nef polyhedra available in CGAL in order to solve various use cases [Bieri and Nef, 1988; Nef, 1978; Hachenberger, 2006][10] (Figure 6). For instance, compliance with height regulations in 3D zoning maps can be checked using Boolean intersections of the model and the polyhedra generated from zoning maps.

The main IFC entity types are thus converted into appropriate CGAL-based representations for points, polygonal curves, polygons with holes, planes, etc. In this process, implicit and parametric curves, surfaces and volumes are interpreted into explicit boundary representations and discretised into polygonal curves and polygonal meshes (Figure 2). All placements and transformations in IFC are converted into 3D affine transformations defined by a matrix, which can then recursively applied to each object as necessary. In this manner,

[8] http://ifcopenshell.org

[9] https://www.opencascade.com

[10] Our previous experience shows that the Boolean set operations in openCASCADE are not as robust as those available in CGAL

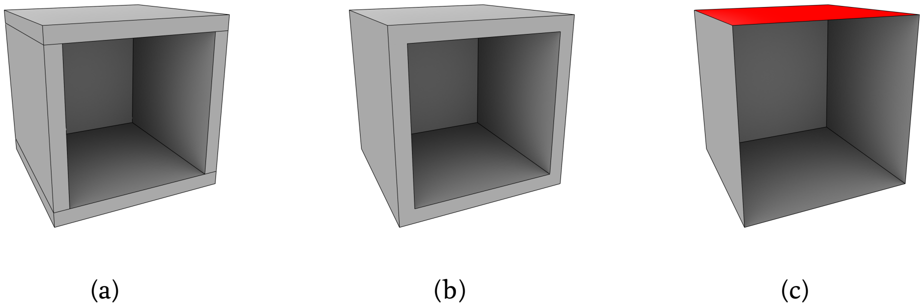


Figure 6: A method to get the semantically labelled outer surfaces of (a) a simple IFC model in a manner suitable for CityGML LOD3 could involve (b) computing the Boolean set union of certain IFC entities (e.g. IfcSlabs) and (c) extracting the outer envelope and classifying surfaces using their normals [Boeters et al., 2015; Donkers et al., 2016].

we obtain a polyhedral representation of every volumetric object and store it as a CGAL `Polyhedron_3`.

Some IFC entity types however require additional processing, such as the CSG solid, half-space and Boolean result representations which are obtained by first converting a polyhedron to a CGAL `Nef_polyhedron_3` and then performing Boolean set union, intersection or difference operations.

5 Issues encountered

In this section we describe the integration issues we have so far encountered and identified.

5.1 Geometric and topological processing issues

We have found that invalid objects are widespread in the IFC models we received as part of this study, which is consistent with our previous experience with BIM and GIS data; see for instance Biljecki et al. [2016a]. Self-intersections and intersections between objects are the most common errors (Figures 7 and 8), but there are also uneven surfaces that are supposed to be planar and disconnected objects that are modelled as one. One of the most interesting errors has been finding objects that are seemingly valid and form topological 2-manifolds (as checked by making sure that surfaces join in pairs along common edges), but in reality contain self-intersections (Figure 9). Note that this is something explicitly disallowed by the IFC standard[11], but not enforced by most of the current implementations.

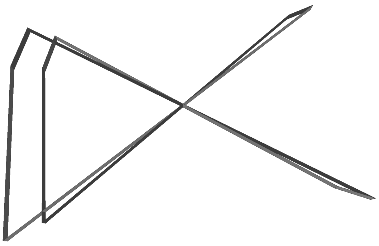


Figure 7: A prismatic polyhedron with an obvious self-intersection. The self-intersecting top and bottom faces of the polyhedron are not shown.

In order to make sure that every face is perfectly planar, we triangulate the non-triangular faces of every object whenever

[11] `http://www.buildingsmart-tech.org/ifc/IFC2x3/TC1/html/ifcgeometricmodelresource/lexical/ifcfacebasedsurfacemodel.htm`

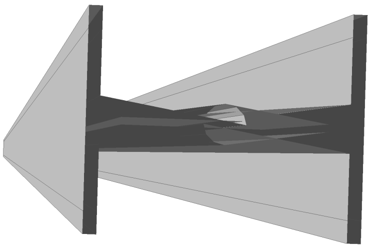


Figure 8: A self-intersecting representation of a beam.

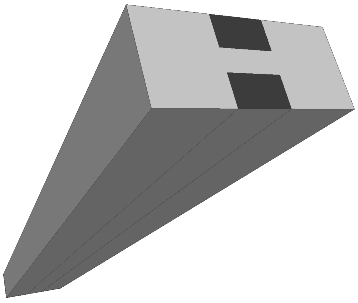


Figure 9: A topological manifold that contains non-obvious geometric intersections. The bottom of the polyhedron, seemingly composed of three rectangular faces, actually has only two rectangular faces that overlap along the middle third. The top of the polyhedron (not visible) has the same problem. Typical validation tests commonly return that such a shape is valid.

we need to create a Nef polyhedron. This ensures that the conversion from a CGAL `Polyhedron_3` to a `Nef_polyhedron_3` is able to compute a plane passing through every face. Another possibility would be to compute the best fitting plane per face and then snap a vertex to the intersection of the planes of its incident faces [Arroyo Ohori, 2016].

In order to catch and correct as many errors as possible, we perform a series of validation tests on every object and to its openings (if any). These are done at appropriate places during the construction of every object, before the conversion of a CGAL `Polyhedron_3` to a `Nef_polyhedron_3`, and before it is triangulated for the generation of a file used for visualisation (the simple OBJ format). We test for:

- combinatorial validity (2-manifoldness),
- surfaces that enclose a space,
- crashes/failures of CGAL's triangulation algorithm (e.g. when a surface self-intersects),
- self-intersections,
- CGAL crashes/failures when converting to a `Nef_polyhedron_3`.

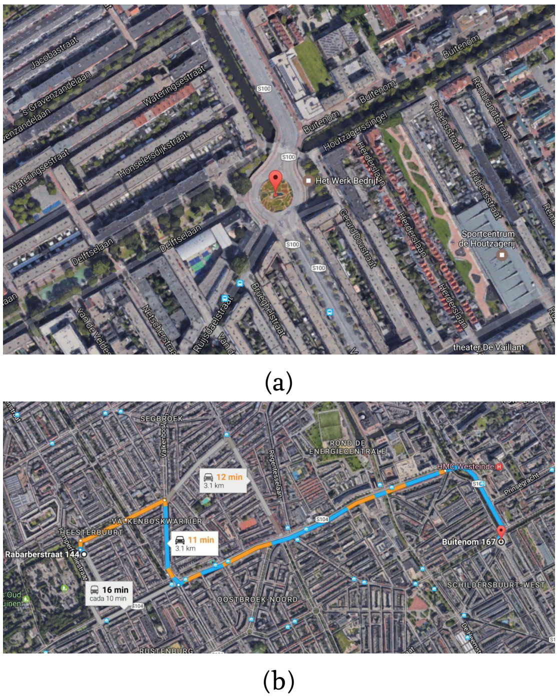
Another issue is related to the generation of discrete and explicit b-rep linear geometries (e.g. polygons and polygonal meshes) from the implicit and curved geometries in IFC. Mainly, explicit geometries will yield different b-rep representations according to the chosen discretisation method and its parameters. For instance, we discretise ellipses into closed polygonal curves using a customisable number of equal-angle intervals, and we discretise spheres into icosahedral approximations with a customisable number of refinements. However, alternative methods could involve discretising ellipses using equal-length line segments and spheres using equal angle rectangular patches.

Finally, we have found that the available features of CGAL are not enough to comfortably model all the complex features of IFC—something that is also true for many other geometric processing libraries. For instance, CGAL Nef polyhedra do have support for half-space representations as long as an extended kernel is used, which incorporates polynomial representations of various classes such as planes. However, this CGAL kernel appears to be incompatible with various parts of the Polygon mesh processing package, which we use for triangle-triangle intersection tests, 2-manifoldness tests, stitching the faces of a polyhedron together, and reversing normals among other functions. Another problem is that Nef polyhedra do not offer a quick way to construct meshes that do not enclose a volume or to create a `Polyhedron_3` from such a mesh stored in a Nef polyhedron. While we have found workarounds around these problems, we are aware that these are slow or do not cover all the cases present in IFC files other than the ones in our use cases.

5.2 Bad georeferencing of IFC models

The IFC standard provides dedicated entities and attributes for georeferencing models, as discussed in Section 2.2. By combining the georeferencing attributes of the `IfcSite` (corresponding to the latitude, longitude and elevation) and the orientation of true North in the coordinate system of the model, it is clearly possible to properly locate a model on a geographic coordinate system. However, we have found that those attributes are often not filled in correctly by practitioners. For example, Figure 10(a) illustrates the provided position of the reference point (latitude and longitude) given in the attributes of the Rabarberstraat 144 model, compared to its real one in Figure 10(b). We can see that the provided point corresponds to the centre of a roundabout in the city of The Hague and therefore appears to be chosen manually and on purpose.

There is a distance of approximately 3 km between the provided and actual positions, which is definitely problematic. However, it is negligible compared to the case of the other files we have studied, which have coordinates corresponding to a different city of the same country, and to a different continent even! This can be explained on one hand by the fact that the georeferencing issue is not the first concern of BIM practitioners in the design phase, and on the other hand, when they may need it, it can be just for indicative purposes. Furthermore, there are no rules or specifications to force BIM practitioners to fill in the proper information at the design phase or at the generation of IFC files. Consequently, such type of information is not always present in IFC, where the single geographic reference point for the construction site may specify the exact position of the origin of the local placement, or an approximation of it [Building SMART International, 2007].



(a)

(b)

Figure 10: Wrong georeferencing of IFC models. (a) Provided position of the Rabarberstraat 144. (b) Distance comparison between the provided and the real positions. Images and map data taken from Google Maps.

With the growing interest of the BIM community regarding the surrounding environment of architectural structures Liu et al. [2017], the georeferencing issue is being considered more and more. For instance, the latest major version of the IFC standard (IFC4) includes new classes intended to provide support for exact georeferencing: `IfcCoordinateReferenceSystem` and `IfcMapConversion`. The former class allows us to define a coordinate reference system based on a specified geodetic datum (WSG84, ED50, etc.), and the latter makes it possible to convert the local origin of the coordinate system to its location on the Earth (easting, northing, orthogonal height) and to rotate the x-axis of the local engineering coordinate system within the horizontal (easting/westing) plane of the map [Building SMART International, 2013].

6 Results, conclusions, and discussion

This paper presents the *GeoBIM* project, our integration project for GIS and BIM data in the Netherlands, and describes its intermediate results. Our experiments have until now focussed on reading the geometry and topology of every object in an IFC file independently, processing them into appropriate CGAL structures and in converting these to .obj files with materials in order to have a visual output of the models. The next phases of the project will involve using operations on the CGAL structures (e.g. Boolean set operations) in order to perform spatial analyses and to do a better conversion to CityGML LOD 2/3.

Our preliminary prototype, which is based on IfcOpenShell and CGAL, is open source and freely available at `https://github.com/aothms/IfcOpenShell_CGAL`. Due to the requirements of CGAL, it is available under the GPLv3 licence.

Based on the preliminary results of the project, it can already be concluded that a full integration of GIS and BIM data is far from straightforward. From a data perspective, this is mainly because existing BIM models contain many geometrical and topological errors which need to be properly handled and often fixed. These may not be problematic when used in a BIM environment because of a few reasons: many more geometry types are usually natively supported in BIM software than in GIS software; geometric errors (e.g. self-intersections) are often found in places that are inconspicuous or invisible from the outside; data is often only used for visualisation purposes, which does not require geometric and topological correctness; or the errors only show after the implicit and parametrised types have been converted into explicit geometries. However, the errors are very problematic in GIS applications because we need to perform spatial analyses which often involve complex operations such as Boolean set operations. Therefore, functionalities to validate the geometry of a BIM model are required; see Ledoux [2013] for an equivalent for 3D GIS datasets.

Another conclusion that can be drawn is that it is unrealistic to develop a transformation for all IFC geometry classes within a small project. In our experiments, we have so far developed a transformation solution for only a subset of the IFC standard. It comprises a combination of entities that we have found frequently and those which are relatively easy to convert to polyhedral representations. We currently support about 60 of the roughly 100 IFC entities that are mainly about geometry. It would take years to extend these to all entities while in practice some of them are rarely used. Therefore it would help to select with BIM experts the most common remaining geometry classes and to focus on these first. A related conclusion is that with IFC there are many different ways to model an object, and that supporting all of them is a problem in practice; and thus this hinders standardisation. Ideally, standards that are intended for exchange should define only one way to model something—this both fosters data exchange and makes it easy to create compliant implementations of the standard.

The history of data exchange and its accompanying problems is also much longer in the GIS domain. As also concluded by Liu et al. [2017] from Cerovsek [2011], the relatively younger concept of BIM and its standard IFC have not satisfied the requirements of standards yet: competitiveness, conformity, and connectivity. BIM has to further develop in this respect, including paying attention to formal definitions for correct geometries, adhering to these and developing functionality to validate models.

Also, the transformation from IFC to CityGML needs to be standardised as there are currently many different interpretations of how to best transform an IFC file into CityGML. From a geometric perspective, this is mainly due to the many classes of IFC that need to be converted into the relatively few classes of CityGML. The processing needed to make explicit geometries from implicit geometries and to change geometry classes (like conversion of volumes in IFC into surfaces in CityGML for walls) can result in different outcomes in different implementations. For a sustainable information chain that supports the life cycle of objects, this is unwanted, and therefore there is a need to define one uniform and standardised transformation. Our project will aim to define such a transformation for the Netherlands.

From our experiments we can also conclude that some awareness of the specificities of the transformation can help to formulate recommendations for the data modelling process in order to better support the transformation, i.e. to create CityGML/GIS prepared IFC data and, vice versa, to create BIM/IFC-prepared CityGML data. Examples are adding the appropriate information in a BIM model so that they can always be precisely located in the real world and limiting the data modelling to the most common geometry classes. Also `IfcSpaces` and `IfcSlab` are two key layers between the conversion of CityGML and IFC [Liu et al., 2017] since they represent IFC concepts that usually have simple geometries that can usually be easily converted to an equivalent in CityGML. Therefore BIM-designers may be encouraged to include these concepts in their models.

More understanding is needed in order to find out how BIM data are used in GIS applications and vice versa. For example, we wonder about how IFC designs are checked against the existing physical world and against a 3D zoning plan (both GIS data): is this done in BIM software, and thus GIS data is imported and localised in BIM software; or is this done in GIS software, and thus BIM design data needs to be imported in GIS software and properly georeferenced.

BIM data is usually much more detailed than what is expected from GIS data. It is thus very unlikely that all the details of a BIM dataset will be integrated into GIS data. Instead, a generalised version of the BIM model (with relevant attributes for the GIS world) will be converted into a GIS model. A 3D city model may serve as a connection between the two, with unique identifiers and update mechanisms in order to keep the separated BIM models consistent with their generalised counterparts.

Future work will focus on formulating recommendations to BIM practitioners and to the relevant standardisation bodies for better integration possibilities, extending our work by identifying which semantics should be preserved in the transformation process, and further developing the IFC/CityGML interface accordingly. We will apply the developed solution to the use cases with close involvement of the users to assure that it does align with the actual needs from practice.

Acknowledgements

This research is supported by the Dutch Technology Foundation STW, which is part of the Netherlands Organisation for Scientific Research (NWO), and which is partly funded by the Ministry of Economic Affairs (Project code: 11300). This project has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement No 677312 UMnD).

References

Sam Amirebrahimi, Abbas Rajabifard, Priyan Mendis, and Tuan Ngo. A BIM-GIS integration method in support of the assessment and 3D visualisation of flood damage to a building. 61(2):317–350, April 2016.

Ken Arroyo Ohori. *Higher-dimensional modelling of geographic information*. PhD thesis, Delft University of Technology, apr 2016.

J Benner, A. Geiger, and K. Leinemann. Flexible generation of semantic 3D building models. In G. Gröger and T. H. Kolbe, editors, *Proceedings 1st International Workshop on Next Generation 3D City Models*, pages 17–22, Bonn, Germany, 2005.

H. Bieri and W. Nef. Elementary set operations with d-dimensional polyhedra. In Hartmut Noltemeier, editor, *Computational Geometry and its Applications*, volume 333 of *Lecture Notes in Computer Science*, pages 97–112. Springer Berlin Heidelberg, 1988.

Filip Biljecki, Hugo Ledoux, Xin Du, Jantien Stoter, K. H. Soon, and V. H. S. Khoo. The most common geometric and semantic errors in CityGML datasets. volume IV-2/W1 of *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, pages 13–22, Athens, Greece, 2016a. doi: http://dx.doi.org/10.5194/isprs-annals-IV-2-W1-13-2016.

Filip Biljecki, Hugo Ledoux, and Jantien Stoter. An improved LOD specification for 3D building models. *Computers, Environment and Urban Systems*, 59:25–37, 2016b.

Roeland Boeters, Ken Arroyo Ohori, Filip Biljecki, and Sisi Zlatanova. Automatically enhancing CityGML LOD2 models with a corresponding indoor geometry. *International Journal of Geographical Information Science*, 29(12):2248–2268, December 2015.

Building SMART International. Industry Foundation Classes (IFC), IFC2x3, 2007. Available from: http://www.buildingsmart-tech.org/specifications/ifc-releases/summary.

Building SMART International. Industry Foundation Classes (IFC), IFC4, 2013. Available from: http://www.buildingsmart-tech.org/specifications/ifc-releases/summary.

Tomo Cerovsek. A review and outlook for a 'building information model' (BIM): A multi-standpoint framework for technological development. *Advanced Engineering Informatics*, 25:224–244, 2011.

Ruben de Laat and Léon van Berlo. Integration of BIM and GIS: The development of the CityGML GeoBIM extension. In T. H. Kolbe, G. Köning, and C. Nagel, editors, *Advances in 3D Geo-Information Sciences*, Lecture Notes in Geoinformation and Cartography, pages 211–225. Springer-Verlag, Berlin Heidelberg, 2011.

Yichuan Deng, Jack C P Cheng, and Chimay Anumba. Mapping between BIM and 3D GIS in different levels of detail using schema mediation and instance comparison. *Automation in Construction*, 67:1–21, July 2016.

Sjors Donkers, Hugo Ledoux, Junqiao Zhao, and Jantien Stoter. Automatic conversion of IFC datasets to geometrically and semantically correct CityGML LOD3 buildings. *Transactions in GIS*, 20(4):547–569, 2016.

Mohamed El-Mekawy and Anders Östman. Semantic mapping: an ontology engineering method for integrating building models in IFC and CityGML. *Proceedings of the 3rd ISDE Digital Earth Summit*, pages 12–14, 2010.

Mohamed El-Mekawy, Anders Östman, and Khurram Shahzad. Towards interoperating CityGML and IFC building models: A unified model based approach. *Advances in 3D Geo-Information Sciences*, pages 73–93, 2011.

Mohamed El-Mekawy, Anders Ostman, and Ihab Hijazi. A unified building model for 3D urban GIS. *ISPRS International Journal of Geo-Information*, 1:120–145, July 2012.

Marcus Goetz. Towards generating highly detailed 3D CityGML models from OpenStreetMap. *International Journal of Geographical Information Science*, 27(5):845–865, May 2013.

Peter Hachenberger. *Boolean Operations on 3D Selective Nef Complexes Data Structure, Algorithms, Optimized Implementation, Experiments and Applications*. PhD thesis, Saarland University, 2006.

Ihab Hijazi, Manfred Ehlers, and Sisi Zlatanova. BIM for geo-analysis (BIM4GEOA): set up of 3D information system with open source software and open specifications (OS). In *Proceedings 5th International 3D Geoinfo conference*, pages 45–49, Berlin, Germany, 2010.

ISO. *Industry Foundation Classes (IFC) for data sharing in the construction and facility management industries.* International Organization for Standardization, March 2013.

ISO. *Industrial automation systems and integration - Product data representation and exchange.* International Organization for Standardization, August 2014.

A Lapierre and P Cote. Using open web services for urban data management: A testbed resulting from an ogc initiative for offering standard cad/gis/bim services. In *Urban and Regional Data Management. Annual Symposium of the Urban Data Management Society*, pages 381–393, 2007.

Hugo Ledoux. On the validation of solids represented with the international standards for geographic information. *Computer-Aided Civil and Infrastructure Engineering*, 28(9):693–706, 2013. ISSN 1467-8667. doi: http://dx.doi.org/10.1111/mice.12043.

Xin Liu, Xiangyu Wang, Graeme Wright, Jack Cheng, Xiao Li, and Rui Liu. A State-of-the-Art Review on the Integration of Building Information Modeling (BIM) and Geographic Information System (GIS). *ISPRS International Journal of Geo-Information*, 6(2):53, February 2017.

Walter Nef. *Beiträge zur Theorie der Polyeder.* Herbert Lang, Bern, 1978.

OGC. Geography markup language (GML) encoding specification. Open Geospatial Consortium inc., 2004. Document 03-105r1, version 3.1.1.

Open Geospatial Consortium. OGC City Geography Markup Language (CityGML) Encoding Standard 2.0.0. Technical report, April 2012.

Azarakhsh Rafiee, Eduardo Dias, Steven Fruijtier, and Henk Scholten. From BIM to geo-analysis: View coverage and shadow analysis by bim/gis integration. *Procedia Environmental Sciences*, 22:397–402, 2014. ISSN 1878-0296. doi: http://dx.doi.org/10.1016/j.proenv.2014.11.037. URL `http://www.sciencedirect.com/science/article/pii/S1878029614001844`.

A. A. G. Requicha. Representation of rigid solids—theory, methods and systems. *ACM Computing Surveys*, 12(4):437–464, 1982.

Rebekka Volk, Julian Stengel, and Frank Schultmann. Building information modeling (BIM) for existing buildings—literature review and future needs. *Automation in Construction*, 38:109–127, 2014.

WP Wang and KK Wang. Geometric modeling for swept volume of moving solids. *Computer Graphics and Applications, IEEE*, 6(12):8–17, 1986.