

Geobim benchmark 2019: Design and initial results

Original

Geobim benchmark 2019: Design and initial results / Noardo, F.; Arroyo Ohori, K.; Biljecki, F.; Krijnen, T.; Ellul, C.; Harrie, L.; Stoter, J.. - In: INTERNATIONAL ARCHIVES OF THE PHOTOGRAMMETRY, REMOTE SENSING AND SPATIAL INFORMATION SCIENCES. - ISSN 1682-1750. - ELETTRONICO. - 42:(2019), pp. 1339-1346. [10.5194/isprs-archives-XLII-2-W13-1339-2019]

Availability:

This version is available at: 11583/2856622 since: 2020-12-11T09:38:25Z

Publisher:

Copernicus Publications

Published

DOI:10.5194/isprs-archives-XLII-2-W13-1339-2019

Terms of use:

This article is made available under terms and conditions as specified in the corresponding bibliographic description in the repository

Publisher copyright

(Article begins on next page)

GEOBIM BENCHMARK 2019: DESIGN AND INITIAL RESULTS

F. Noardo^{a,*}, K. Arroyo Ohori^a, F. Biljecki^b, T. Krijnen^c, C. Ellul^d, L. Harrie^e, J. Stoter^a

^a Department of Urbanism, Delft University of Technology, Delft, The Netherlands - (f.noardo, G.A.K.ArroyoOhori, J.E.Stoter@tudelft.nl)

^b Department of Architecture, National University of Singapore, Singapore - filip@nus.edu.sg

^c Department of the Built Environment, Eindhoven University of Technology, Eindhoven, The Netherlands - t.f.krijnen@tue.nl

^d Department of Civil, Environmental and Geomatic Engineering, University College London, London, UK - c.ellul@ucl.ac.uk

^e Department of Physical Geography, Lund University, Lund, Sweden - lars.harrie@nateko.lu.se

Commission IV, WG IV/10

KEY WORDS: GeoBIM, software support, interoperability, open standards, CityGML, IFC, 3D city models, BIM.

ABSTRACT:

GeoBIM, the integration of 3D geoinformation (Geo) with building information models (BIM), is a subject of increasing attention in both domains. A well-known practical challenge for this integration is the mixed state of software support for open standards in each domain that would ease the integration. This is often known by practitioners but poorly documented. In order to solve this problem, we devised the GeoBIM benchmark, in which we compile the experiences of volunteering participants, who perform a guided study to test the software they are most familiar with against a few provided datasets structured in open standards. The aim of the tests is to improve the knowledge of the state of the art in the software support for GeoBIM open standards and to identify points for improvement. In this paper, we present the design of the benchmark, especially explaining and discussing the chosen data to be used with their connected issues to be tested, and some initial results.

1. INTRODUCTION

In recent years, 3D information about cities, and the constructions in cities, has become more and more powerful and critical to assist users in a number of use cases (Biljecki et al, 2015). Some examples of users are: public entities (e.g. land description, development control, analysis and a number of smart city applications), practitioners (e.g. city planners, building designers, and asset managers), scholars (much research has a fundamental contribution by 3D information and 3D models).

To support these urban applications two main sources of data can be identified:

- 3D city models, which are used to represent city objects and substitute previous maps and other cartographic products, in order to support city analysis and management, city planning, navigation, and so on;
- Building information models (BIM), which are used in building and design of buildings, infrastructure and other constructions, and which also have features useful to project management and asset management.

Although both model the built environment, these have different characteristics, including: the kind of geometry which is used (usually solid parametric objects in BIM and surfaces through boundary representation in 3D city models), the semantics used to structure their entities (e.g. specific materials and building components for BIM and city objects and uses for 3D city models), and the need to georeference the models (essential in 3D city models and not so for BIM). These differences are also reflected in the respective open standards,

which are used to archive their information in interoperable and open formats, namely: CityGML (by the Open Geospatial Consortium) (OGC, 2012) and the Industry Foundation Classes (IFC) (by the buildingSMART consortium) (ISO16739, 2013), which are the most well-known standards to represent and exchange, respectively, 3D city models and BIM models.

Due to the overlapping interests in both fields, increasing attention is being paid to 3D city model-BIM integration (GeoBIM), where the exchange of information between geospatial (3D city models) and BIM sources (Ellul et al., 2018) enable the reciprocal enrichment of the two kinds of information with advantages for both fields (e.g. automatic updates of 3D city models with high-level-of-detail features, automatic representation of BIM in their context, automated tests of the design, and so on). Much research is currently ongoing in this field (Liu et al., 2017, Zhu et al., 2018, Stouffs et al., 2018). However, the connection with the world of practice and the availability of mainstream technical solutions is limited.

To address this challenge, a number of projects have been developed to investigate the issue and fill the interoperability gap between Geo and BIM, such as the OGC Future City Pilot^{1,2}, previous GeoBIM projects at the Delft University of Technology³, and on-going research performed and sponsored by EuroSDR⁴.

More recently, the GeoBIM benchmark⁵, which we discuss in this paper, was funded as an ISPRS scientific initiative 2019 (and co-funded by the European Spatial Data Research association, EuroSDR) to test the available software tools

* Corresponding author

¹ <http://www.opengeospatial.org/projects/initiatives/fcp1>

² <https://www.opengeospatial.org/projects/initiatives/fcp2>

³ <https://3d.bk.tudelft.nl/projects/geobim/>

⁴ <https://3d.bk.tudelft.nl/projects/eurosd-geobim/>

⁵ <https://3d.bk.tudelft.nl/projects/geobim-benchmark/>

against their support of open standards identifying points for improvements.

This paper presents the design of the benchmark, especially focusing on the provided data, and the initial results obtained.

2. THE GEOBIM BENCHMARK INITIATIVE

The aim of this project is to provide insight into the current state of the art of the implementation of open standards in the 3D geo and BIM domains, and to identify what are the remaining compatibility issues.

Some activities related to the software certifications are ongoing, especially a structured and affirmed procedure is used to certify software for IFC support⁶. However, since we experienced inconsistent imported and exported data even in certified software when working with data produced by designing practice, we deemed it important to assess this software too in a common framework.

The main objective of the benchmark is to provide a framework describing the present ability of existing software tools to use (read and visualise, import, manage, analyse, export) CityGML and IFC models and understand their performance while doing so, both in terms of information management functionalities, and, eventually, information loss, and in terms of ability to handle large datasets.

Areas to be explored within this project include the issues relating to existing software and tools for effectively managing CityGML and IFC, which sometimes do not effectively support features that a) are present in the native formats of the software (e.g. different types of geometries, topological links between objects or rich semantics), b) have limits in their representation (geometry, semantics, georeferencing), c) can generate ancillary information inconsistently (e.g. IFC spaces and their semantics), or d) can generate errors and wrong representations (e.g. Bazjanac, 2002). Three aspects of this problem are examined:

1. Firstly, software support for the individual standards (CityGML and IFC) within their respective domains – i.e. How well is CityGML supported by GIS (and other) tools and IFC by BIM (and other) tools?
2. Secondly, options for IFC geo-referencing is explored, as this is a fundamental step in enabling information exchange.
3. Thirdly, a critical test of the available conversion procedures between CityGML and IFC (in both directions) is conducted to objectively assess and compare potential for integrated data use, starting from the same premises.

Using a systematic testing approach will highlight the existence of these issues at least for the most widely used software packages, and will help to identify areas where improvements are needed. In particular, the four tasks investigated in the benchmark are:

Task 1 - Support for IFC within BIM (and other) software;

Task 2 - Options for geo-referencing BIM data;

Task 3 - Support for CityGML within GIS (and other) tools;

Task 4 - Options for conversion (software and procedural) (both IFC to CityGML and CityGML to IFC).

Voluntary participants are a very important part of the initiative. They perform one or more tasks with the tools they are familiar with, and deliver their results in the provided results template. This will allow joining the efforts and the individual experiences with specific (technical) tools to build a common list of tools that are compared following common criteria.

3. PREPARING THE DATA FOR THE BENCHMARK

One of the most critical challenges of the benchmark is the provision of adequate data to test software, tools and procedures without biases and effectively serve the purpose. Potentially, two kinds of data are available: the data modelled and generated in academic environments and data derived from real-world practice.

Both have advantages and disadvantages: the first ones are purposefully made to be clean and perfect, so that, in theory, they can be better managed by digital tools. Therefore, the results would be unrealistically positive for the software's behaviour in practice. At the same time, the second ones are generally developed without the aim of being exchanged using open standards, so that many aspects that are important for these models to be used in an interoperable environment are disregarded (e.g. the validity of geometries, the correctness of semantics and so on) (Arroyo Ohori et al., 2017). In this latter case, the issues with software support could be caused by the inaccuracies that are a result of exporting the models rather than the software itself, and the results would be negatively influenced without objective reasons.

To overcome this challenge, in order to obtain data suitable for the benchmark tests we have taken datasets modelled by practitioners and inspected and analysed them (with focus on geometry, semantics and georeferencing) using 3D viewers and validation tools (e.g. IfcViewer, Solibri Model Checker, FZK, Azul). This permits an understanding of the main limitations of such models, for example, inaccurate semantics and the inaccuracies in the modelling of geometries, prior to their use in the benchmarking activities. As part of this process, the better models were chosen among the available ones and the main issues resolved through specific procedures and operations, as it is explained in detail in Sections 4 and 5. This will ensure that, as far as possible, consistent results are produced from the benchmark and any anomalies can be linked to how specific software handles the data provided.

4. IFC DATA

4.1 IFC data by practitioners for designing buildings

For this and further on-going projects, we had the opportunity to consider a couple (approximately 10) of BIMs provided by some Municipalities in the Netherlands (Rotterdam, Den Haag, Almere) and for a further project working on GeoBIM adoption in Sweden⁷. As required by our benchmark, these are built by building designers for their construction aims. It can be challenging to obtain such kinds of data, since they usually involve very strict privacy and copyright issues.

⁶ <http://www.buildingsmart-tech.org/certification/details>

⁷ <https://www.smartbuilt.se>

We could inspect them in some of the most common 3D viewers (RDF IFC Viewer⁸, KIT FZK viewer⁹, Solibri Model Viewer¹⁰) and analysed them through the NIST STEP File Analyser and Viewer¹¹, which is one software able to open a STEP file and analyse it, generating a spreadsheet or CSV file summarising its characteristics (used entities, relationships between entities, geometric features and so on). The IFC files, encoded in STEP (Standard for the Exchange of Product models), and having the reference schema encoded in the EXPRESS modelling language, can be effectively processed with that tool. The current version of the software (3.40) cannot work with IFC 4 files. In addition, we also inspected the text format of the IFC files (through common text editors) in order to understand some more straightforward characteristics: e.g. the software used for modelling, the used IFC version, the use of georeferencing information.

Although IFC is a very complete but complex standard, providing a high number of entities for describing the buildings, and many solutions are available to model BIMs, interestingly we found that the models we analysed, coming from different sources having no reciprocal connections, have almost identical characteristics. For example, they are using the IFC version 2x3, even if the IFC version 4 has been released from 2013, they are all originally modelled in the software Autodesk Revit, and use for describing their semantics usually no more than (the same) approximately 40 IFC classes (e.g. *IfcProject*, *IfcSite*, *IfcBuilding*, *IfcBuildingStorey*, *IfcSpace*, *IfcOpening*, *IfcWall*, *IfcWallStandardCase*, *IfcColumn*, *IfcWindow*, *IfcSlab*, *IfcMember*, *IfcDoor*, *IfcFlowSegment*, *IfcBuildingElementProxy*).

From the sample to be tested, we chose to consider for the benchmark the ones having no serious geometry issues (e.g. not many intersections, etc.), the most complete semantics as possible, including the definition of attributes and relationships (hierarchies, groups, associated materials) and, for obvious reasons, the permission to be shared at least among the participants of the project. Among these, one describing a small building was selected in order to test the software functionalities in the most reliable way as possible, and another one representing a bigger project, in a heavier file was added to test also the connected software-and-hardware performances. Moreover, the origin of the models was also considered for the final choice, so that no specific national rules or best practices could affect the results of the benchmark. Taking all these considerations into account, the ‘Myran.ifc’ model, from Sweden and the ‘UpTown.ifc’ model from the Netherlands were finally selected (Sections 4.2.1 and 4.2.2).

We had to add another model to the proposed ones, in order to consider IFC version 4 in the benchmark. In the first moment it was not included, since many issues were already found when managing the previous and most widely spread version of IFC (2x3). Furthermore, it is very difficult to find models in IFC4. However, it is important, especially to test tools and functionalities which are programmed explicitly for IFC4. Also, to see the difference between the management of the two versions of IFC is worth. For this reason, we added the ‘Savigliano.ifc’ model. It could be slightly different from the others, since it was modelled in connection with research

environment (Politecnico di Torino, Department of Architecture and Design, prof. F. Rinaudo), by an architect specifically expert in BIM and IFC (L. Polia), so that many aspects (e.g. grouping of entities, semantics definition) were more controlled than in common models (Section 4.2.3). This last one is describing a building to be located in Savigliano (Italy).

It was not possible to completely analyse the geometries used for the building models, since the involved elements are many and very complex. So it was impossible to foresee all the possible configurations used in practice (e.g. for modelling beams and very specific components). Therefore, one more IFC file was modelled (Section 4.4.4). This one, provided in both IFC 2x3 and IFC 4, was conceived in order to test the geometries whose management is theoretically allowed by IFC, and which could be widespread in the practice of building modelling, but which often give errors when used by software.

4.2 The selected IFC models

In the next subsections 4.2.1-4.2.4 a description of the provided IFC files are summarised. The full description can be accessed from the project website¹².

4.2.1 The Myran IFC model: It is the IFC model of a 2-floor office building in Falun, Sweden. It was provided for this study by *MONDO arkitekter*, Falun, Sweden. The data represent the architectural model of the BIM (Figure 1). It was exported on 25/10/2017 from the base software Autodesk Revit 2018 (ENU) using the IFC version 2x3 (MVD: CoordinationView v.2.0).

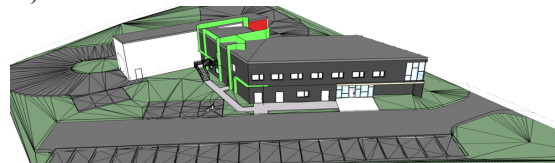


Figure 1. ‘Myran.ifc’ model.

The model was georeferenced through the attributes *RefLatitude*, *RefLongitude* and *RefElevation* in the *IFCSITE* entity. Although the coordinates of the reference point (blue in Figure 2) were expressed in the Swedish reference system EPSG: 3013 SWEREF 99 15 45, RH2000, the reference system was not cited in the model.

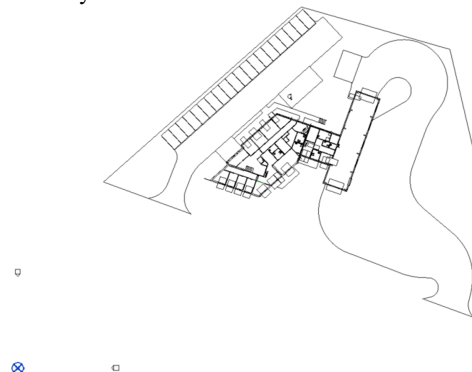


Figure 2. 2D view of the ‘Myran’ IFC model in Autodesk Revit; the blue crossed circle identifies the base point of the model, to which coordinates are associated.

⁸ <http://rdf.bg/product-list/ifc-engine/ifc-viewer/>

⁹ <https://www.iai.kit.edu/1302.php>

¹⁰ <https://www.solibri.com/solibri-model-viewer>

¹¹ <https://www.nist.gov/services-resources/software/step-file-analyzer-and-viewer>

¹² <https://3d.bk.tudelft.nl/projects/geobim-benchmark/data.html>

As we can see in Figure 2, the model is correctly oriented towards the True North, but the rotation is not part of the IFC file. We could therefore suppose that the building was modelled in an already correct orientation, or that it was rotated before the export to IFC, without using the georeferencing tools.

The semantics are employed accurately and many attributes are filled. However, the grouping of entities in 3 storeys is not consistent, as is shown in Figure 3.

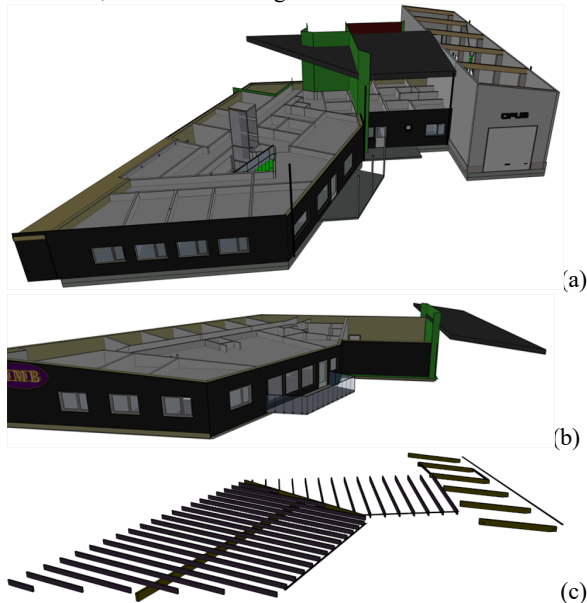


Figure 3. Grouping of entities in storey 1 (a), storey 2 (b) and storey 3 (c)

4.2.2 The UpTown Building IFC model: It is the IFC model of a large residential building under construction in Rotterdam, the Netherlands. It was provided for this study, and for the connected EuroSDR project regarding the use of GeoBIM for automating the building permission issuing, by the Municipality of Rotterdam.

The data represent the architectural model of the BIM (Figure 4). Also, the structural model was available, but we decided to use this one, having a higher number of IFC entities represented and also a major dimension (252,7 MB) which was useful to test the associated software and hardware performance. It was exported on 20/03/2017 from the base software Autodesk Revit 2015 (ENU) using the MVD CoordinationView v.2.0. In this case, no georeferencing details are provided in the model.

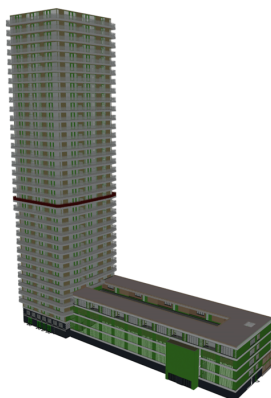


Figure 4. The architectural model of UpTown model.

In the UpTown model, a high number of `IfcBuildingElementProxy`, which is a generic IFC entity used to cover the semantics of objects not explicitly defined by other IFC entities, is used instead one of the many specific IFC classes which would have given more precise information of the element used.

4.2.3 Building in Savigliano, IFC model: This is an IFC model of a designed residential building in Savigliano (Italy) (Figure 5). It was modelled by Lorenzo Polia during a research in Politecnico di Torino, DAD. The data represent the architectural model of the BIM. It was exported on 14/03/2019 from the base software Autodesk Revit 2019 (ITA) in IFC 4 (DesignTransferView v.1.0). Also for this case, no georeferencing data are included in the file.



Figure 5. Savigliano.ifc BIM model.

4.3 A further model: IFC geometries

The IFC standard defines a comprehensive set of geometry definitions, ranging from parametric cross section profile definitions to extrusions and revolutions thereof, boolean operations and explicit boundary representations with arbitrary underlying surfaces. However, the actual geometries encountered in IFC building models in practice are known to often be lower order tessellated geometries, where semantically richer and more precise models could have been more appropriate (Jeong et al., 2009; Feringa and Krijnen, 2015).

In addition, the implementation of IFC in exporting applications has been an evolving process and as such files of a lower quality have been generated. On the other hand, applications that import the IFC data have been trying to accommodate such modelling errors and implement workarounds, to assure a stricter standard compliance. To give a concrete example the length of an extrusion (`IfcExtrudedAreaSolid.Depth`) has to be positive (it is of type `IfcPositiveLengthMeasure`), however some applications will accept negative values and extrude in the reversed direction. Some applications will also offer work-around solutions for self-intersecting face boundaries or shells, something also explicitly forbidden by the standard (Arroyo Ohori et al., 2017).

In order to specifically test definitions besides the prevalent ones and in order to assess the diverging compliance within software, a specific set of geometries was generated for this benchmark. It includes an esoteric profile definition (that has since been removed in IFC4, `IfcCraneRailAShapeProfileDef`), cases where the extrusion depth is negative, and cases where the extrusion direction is parallel to the base surface. Lastly, it includes a

sweep along a parametric range of a curve. The generated file also contains conventional geometries, such as Boolean operations and revolutions (Figure 6).

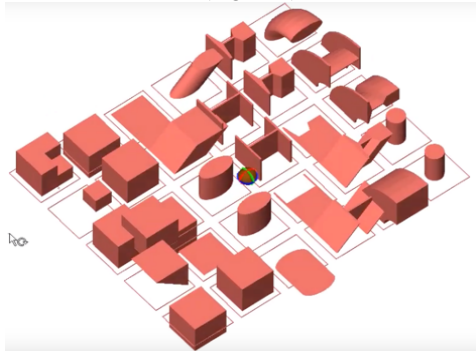


Figure 6. A view of the IFC geometries as visualised in the FZK viewer.

5. CITYGML DATA

Many open data about 3D city models are available in CityGML¹³. We selected some models allowing the test of different features of CityGML: they should cover the most part of available CityGML classes; one model was selected as covering one entire city (Amsterdam), which could be essential when running analysis on the whole city, even if the file is very heavy; another model was chosen including different levels of details (LoDs), which is an interesting opportunity in CityGML, but not always well supported; and finally we included LoD3 CityGML data, requiring whole management of 3D objects.

In this case, the version used for the data is CityGML v.2, without exceptions, because the next version of CityGML (v.3) is still under development, therefore software tools are not supposed to support it yet.

In order to have some control over the potential issues when working with the CityGML datasets, we have checked them using val3dity (Ledoux, 2018), which is an open source program that validates them against the ISO 19107 standard. Among other issues, it tests for planarity and proper nesting of rings, polygons, shells and solids

5.1 The selected CityGML models

As in the previous section, a summary of data characteristics is described in next subsections 5.1.1-5.1.3. The full description, and download, since these are open data, can be accessed through the project website¹⁴.

5.1.1 The CityGML model of Amsterdam in LoD 1: This model represents the entire city of Amsterdam (Figure 7), including all the CityGML entities for a seamless 3D city model (buildings, roads, vegetation, and so on) in LoD 1. As usual in the Netherlands, the used coordinate reference system is the Amersfoort / RD New (EPSG: 28992). It was produced based on the open datasets BGT (the large-scale topographic map of the Netherlands) and AHN3 (laser altimetry point cloud) using the software 3dfier developed by the 3D geoinformation group at the Delft University of Technology, and it is released as open data¹⁵. 3dfier produces already validated models.



Figure 7. CityGML model of Amsterdam in LoD 1.

It is a very large model (4.36 GB) and it was selected as part of the benchmark data in order to test whether software tools are able to deal with heavy CityGML datasets, such as those including one entire (medium-sized) city.

5.1.2 CityGML model of Rotterdam in LoD 1 and 2: This model represents the surrounding area of the UpTown building in Rotterdam (one of the provided IFC models for this benchmark, see section 4.2.2); it represents the *Maritiem* district in Rotterdam (Figure 8). It is open data produced and provided by the City of Rotterdam. Only the CityGML class Building is included, represented in two LoDs: LoD 1 (extruded building footprints), LoD 2 (more detailed external surfaces of the buildings, with realistic roof shapes and different semantics for walls and roofs). The dataset downloaded from the Rotterdam portal, also included the LoD 0, but it was actually a duplication of the LoD 1. Therefore we decided to remove it from the data. For doing this, a manual procedure was followed.

We validated the model using val3dity and found that it is of relatively high quality, but it does have errors on 3.5% of the primitives (non-planar polygons, open shells and self-intersections). Based on our experience, it is thus a cleaner model than most other 3D city models of that size and complexity (which tend to have substantially more errors) see Biljecki et al 2016. The imperfections were kept to test software behaviour against these errors.

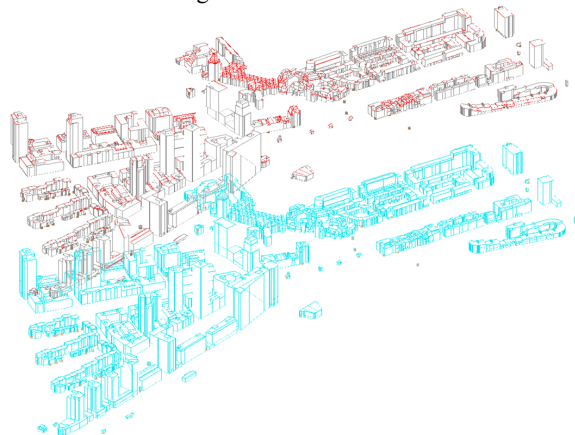


Figure 8. CityGML dataset representing a Rotterdam district in LoD 1 and LoD 2. Source of the data: City of Rotterdam.

¹³ <https://3d.bk.tudelft.nl/opendata/>

¹⁴ <https://3d.bk.tudelft.nl/projects/geobim-benchmark/data.html>

¹⁵ <https://3d.bk.tudelft.nl/opendata/3dfier/>

5.1.3 Buildings in LoD 3: Since few datasets are available in CityGML LoD 3, we used a dataset procedurally generated by the open source tool Random3Dcity (Biljecki et al., 2016b), see Figure 9. The reference system generated for this data is the Dutch one (Amersfoort/RD New EPSG:28992) because, even if they are fictional buildings it is important to carry out tests using georeferenced data. An advantage of using procedurally generated data is that it is usually free of geometric errors thanks to their sourcing nature.

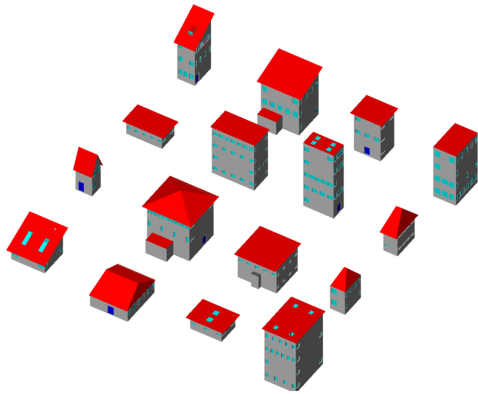


Figure 9. A CityGML LoD3 model representing buildings generated with procedural modelling.

6. INITIAL RESULTS

Although the initiative has only recently started (less than one month at the time of the writing of this paper), it has already generated some interest, and some tests were already performed by participants, including 1st year students of the MSc in Geomatics at the Delft University of Technology: 5 software were tested (3 BIM software for Task 1 and 2 GIS software, plus a 3D Viewer for Task 3). Moreover, few tests for Task 4 (conversion procedures) were delivered, but they are not considered here, since it is necessary to analyse such results in a wider sample in order to draw sensible conclusions (comparison between different procedures is still not possible). We present in this section some of the initial findings, considering the 3 BIM software and the 2 GIS software, tested with two tests each.

6.1 Initial results for Task 1 - Support of IFC within software

The tools tested during the initial activities were: Autodesk Revit 2018¹⁶, Graphisoft ArchiCAD 22.0.0¹⁷, which are among the most widespread BIM software, and Vectorworks Designer 2019 SP2¹⁸, which is another effective BIM software, including many functionalities.

They all have a lot in common: they all are BIM software, and offer both import and export functionalities, together with view, query and analysis tools. They have also been granted the IFC certification in the period between 2013 and 2015, in the certification program IFC2x3 'Coordination View 2.0', for import, and 'Coordination View 2.0 – Arch' for export¹⁹. They gave very similar results in testing the buildings models (Myran, Uptown and Savigliano).

Some main differences were found when managing the Uptown building model, which already pointed out the

performance of the tools in connection with the hardware requirements. Considering the same kinds of hardware for processing, in Revit and Vectorworks the system crashes without completing the operation, while ArchiCAD has no problems in importing it and managing it (import, zoom, pan, rotate, query) generally in less than 1 minute (up to 5 minutes approximately, for importing it).

The tools for reading the georeferencing details (which are not so complex, in this case) are similar in the three software, the coordinates of the base point and elevation (which is the information stored in the IFCSITE entity, with the attributes RefLatitude, RefLongitude and RefElevation) can be read correctly, and the rotation towards the True North does not change. In the data the TrueNorth is not explicit (in the specific attribute within IfcGeometricRepresentationContext). Therefore, we do not have the possibility to check how the software could read that information. We could have changed this by adding such a detail in the text file of the data, however, the IFC file could be corrupted by such intervention. Therefore we chose for the moment to avoid this.

The semantics was found to be consistent with the corresponding IFC entities, except in one case, for Revit, in which, for example, the entities 'IfcTank', 'IfcTankType', 'IfcTransportElement', 'IfcTransportElementType' were all converted in the software as 'Specialty Equipment'. Another example, with more common entities is when considering the wall: 'IfcWal', 'IfcWallStandardCase', 'IfcWallType' are all flattened to 'Wall'. It is possible to eventually set (manually) the corresponding Revit Categories. Moreover, the hierarchical relationships, and also other kinds of relationships (e.g. grouping) were also not consistent with the IFC ones in Revit. This could be due to the fact that, while importing the file, some error occurred, that required to separate elements to repair them. Therefore, some relationships were consequently solved. Another participant testing the same software answered that the elements appeared to be different from the original IFC-hierarchy, once imported, and ordered in the Revit families. This is much more interesting since the software used to model the original file (and from which the IFC data were exported) was actually Revit. When looking at the attributes, generally all respondents found them consistent with the ones present in the original IFC file. Very different answers were given about the relationships, since, apparently it is not a straightforward functionality of the tools to understand them.

They all allow the view of the model in 2D and 3D and they all allow the editing of the model (attributes, geometry, georeferencing, scale). Moreover, many and even complex analysis can be performed in the tested software, as usual for BIM software.

All the tested software also have an export capability. Whilst for Revit and ArchiCAD no processing is required (by default) to enable a consistent export, Vectorworks requires a layer mapping pre-process for defining storeys. It is possible to select which Model View Definition (MVD) and IFC version to use for exporting the model, and customise them in Revit and ArchiCAD. Vectorworks is at the moment more limited, since it only allows the export to IFC 2x2 and 2x3, and only Revit foresees the export to IFC 4.

¹⁶ <https://www.autodesk.it/products/revit/overview>

¹⁷ <https://www.graphisoft.com/archicad/>

¹⁸ <https://www.vectorworks.net/en>

¹⁹ <https://www.buildingsmart.org/compliance/certified-software/>

Something more specific should be said for the file ‘IFC geometries’, because software had some interesting behaviour in managing it. During the import, Vectorworks does not report any errors, whilst Revit and ArchiCAD do. Specifically, Revit reports on a possibly empty geometry and ArchiCAD reports on inconsistencies found during the reading of IFC, then 7 elements were not imported in one case because of missing/incorrect geometries, and 1 possible data loss was also cited. In another case, an error said that the IFC file is corrupt (but we are sure that it is not, since it was purposely generated without errors).

The tested tools are not able to read correctly all the geometries, especially some of them: none of the tested software could read the geometries similar to the one in Figure 10, which are extruded shapes having a footprint with oblique edges. The same for similar shapes, generated from the same footprint extruded through sweeping it along curved and oblique axis.

The shape in the position 44 in the grid (Figure 11), which is a swept H shape, could only be read as a footprint, or flattened on the plane, as also happens for other geometries, like the one in 13, in 35, in 42, in 44 and 53 (Figures 12). The curved geometries look smooth and are correctly discretized (Figures 12-13). Something interesting happens with the two small cylinders in the corner (positions 46 and 56 in the reference grid, Figure 11): in ArchiCAD the curved surfaces are discretized only in some cases, the two cylinders in the corner have a different behaviour than the others and look smooth in this case (Figure 12). The opposite behaviour is registered in Vectorworks, where they are roughly discretised, becoming octagons (Figure 13c). Moreover, those two cylinders are also supposed to have different heights, but actually, none of the software can read this. Similarly, the two H beams in positions 24 and 34 have different heights. This could be read in Revit and Vectorworks, while not by ArchiCAD, where they appear as having the same height.

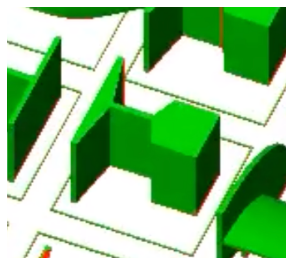


Figure 10. Shape in places 15 and 25 in IFC Viewer

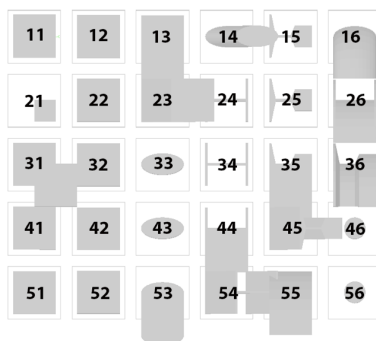


Figure 11. Reference grid for geometries description.

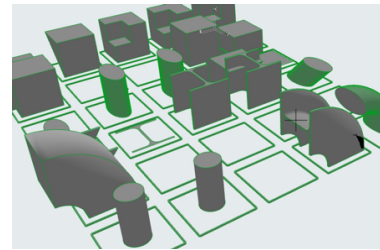


Figure 12. IFC geometries visualised in ArchiCAD, as an example.

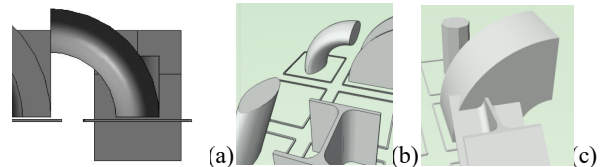


Figure 13. smooth surface in Revit (a) and Vectorworks (b, c). Again, something strange happens, since the surfaces look generally smooth, but (b) the small cylinders in the corner are converted into (c) octagons.

6.2 Initial results for Task 3 - Support of CityGML within software

Four tests have been made so far, two using ESRI ArcGIS and two using QGIS. In all the cases, CityGML format was not directly supported. Attempting to read the ‘Rotterdam’ CityGML file directly in QGIS and ArcGIS resulted in a set of read-only 2.5D points, representing the buildings (likely their centroids) with their associated attributes. This can be due to the misinterpretation of geometries, stored with the two joint LoDs. Instead, when importing the ‘BuildingsLoD3’ file, no geometry is visualised in QGIS (v. 3.0.2); only an attribute table is shown. This is, possibly due to high complexity of the data. Differently, importing ‘amsterdam.gml’ directly is possible, even if it takes some time (i.e. a couple of minutes). In this case, the geometries (triangulated surfaces) are imported as a QGIS layer for each entity, visualised in 2D in usual map and in 3D in the 3D map (new functionality in QGIS 3). However, the participants running the tests decided to use a specific translation tool to correctly import the 3D geometries consistently, with a same procedure for all the three files: the ‘Data Interoperability’ extension in the case of ArcGIS, and the conversion to CityJSON, through citygml-tools²⁰ and the CityJSON Loader plugin for QGIS²¹. During such a conversion, in one case one file was too big to be processed with citygml-tools for conversion to CityJSON (the Amsterdam.gml file) and it was not possible to continue. ArcGIS also struggled because of memory limitations. Apart from this model, processing times were fine for usual operations (e.g. opening a model in a few minutes).

As would be expected for GIS software, both QGIS and ArcGIS handled large numbers of features and georeferencing without problems. Displaying geometry and semantics was also fine.

In the case of QGIS, the conversion to CityJSON made it possible to view and query the models, but some limitations remained. There were problems with models with multiple LoDs, and there was a certain loss of semantics (for example, hierarchical relationships are not maintained and attributes are stored in different tables, connected through parent IDs). The

²⁰ <https://github.com/citygml4j/citygml-tools>

²¹ <https://github.com/tudelft3d/cityjson-qgis-plugin>

hierarchical relationships between objects were also lost. Moreover, since the 3D viewer in QGIS is still rather basic and detached from the main 2D view, operations apart from visualisation required some extra work. The 3D visualisation in ArcGIS through ArcGlobe or ArcScene actually presents more advanced features.

7. DISCUSSION AND CONCLUSIONS

This paper presents the design of the benchmark to test software support for open GeoBIM standards, especially the provided data, and the initial results obtained from it. The GeoBIM benchmark is shaping up to become a useful reference for the current capabilities of software tools regarding open 3D geoinformation and BIM standards.

The selection of the data to be used for this study was not straightforward: challenges were to obtain and suitably describe good test data coming from practice without changing them too much nor providing useless data. Privacy and copy right issues were also important to adhere to, especially for IFC data.

The test of the benchmark tasks with MSc students in geomatics was intended to have the materials (including the data) and some initial software tested by participants who are still new to open standards or specific software. This was useful to confirm that the benchmark can effectively point out the main difficulties encountered by mainstream users, as well as opportunities. However, this could have slightly affected the initial results about the software performance, which were checked by the authors against big inaccuracies. In the following part of the project, we plan to involve participants who are more experienced with the tested software. Notwithstanding, the initial testers have already contributed valuable results and have provided useful feedback to improve the clarity of the explanations and survey forms. Based on these findings the tests were improved before involving a larger team of testers.

Much more tests are still needed in order to obtain a comprehensive view of the state of the art. With this paper, we also would like to encourage prospective volunteers to participate in the benchmark by performing one or more tasks or to provide feedback on the design of the benchmark, see <https://3d.bk.tudelft.nl/projects/geobim-benchmark/>

ACKNOWLEDGEMENTS

This work was possible thanks to the collaboration of the whole GeoBIM benchmark team and advisory board (with their work as in-kind contribution to the project), all the data providers, the students of the course 'Datasets and quality' at the Delft University of Technology, MSc in Geomatics (2018-2019), who ran most of the initial tests, to the ISPRS and EuroSDR funding. This project has also received funding from the European Research Council (ERC) under the European Unions Horizon2020 Research & Innovation Programme (grant agreement no. 677312 Urban modelling in higher dimensions).

REFERENCES

Arroyo Ogori, K., Biljecki, F., Diakité, A., Krijnen, T., Ledoux, H., Stoter, J., 2017. Towards an integration of GIS and BIM data: What are the geometric and topological issues. In: *ISPRS Annals of Photogrammetry, Remote Sensing &*

Spatial Information Sciences, Proceedings of ISPRS 12th Geoinfo Conference, Melbourne, Australia. 26-27.

Bazjanac, V., 2002. Early lessons from deployment of IFC compatible software. In: *eWork and eBusiness in Architecture, Engineering and Construction*, Proc. fourth Euro. conf. product process modelling, Portorož, SLO, pp. 9-16.

Biljecki, F., Ledoux, H., Du X., Stoter, J.E., Soon, K., Khoo, V. 2016. The most common geometric and semantic errors in CityGML datasets. *ISPRS Ann. Photogramm. Remote Sens. Spatial Inf. Sci.*, IV-2/W1: 13-22, 2016.

Biljecki, F., Ledoux, H., Stoter, J., 2016b. Generation of multi-LOD 3D city models in CityGML with the procedural modelling engine Random3Dcity. *ISPRS Ann. Photogramm. Remote Sens. Spatial Inf. Sci.*, IV-4/W1, 51–59.

Biljecki, F., Stoter, J., Ledoux, H., Zlatanova, S., and Çöltekin, A. 2015 Applications of 3D city models: State of the art review. *ISPRS International Journal of Geo-Information*, 4(4):2842–2889, December 2015.

Ellul, C., Stoter, J., Harrie, L., Shariat, M., Behan, A., Pla, M., 2018. Investigating the state of play of GeoBIM across Europe. In: *International Archives of the Photogrammetry, Remote Sensing & Spatial Information Sciences*, 42.

ISO16739:2013. *Industry Foundation Classes (IFC) for data sharing in the construction and facility management industries*. International Organization for Standardization, March 2013.

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

Liu, X., Wang, X., Wright, G., Cheng, J., Li, X., Liu, R., 2017. 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.

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

Stouffs, R., Tauscher, H., & Biljecki, F. (2018). Achieving Complete and Near-Lossless Conversion from IFC to CityGML. *ISPRS International Journal of Geo-Information*, 7(9), 355.

Zhu, J., Wright, G., Wang, J., Wang, X., 2018. A critical review of the integration of geographic information system and building information modelling at the data level. *ISPRS International Journal of Geo-Information*, 7(2), 66.

Jeong, Y.-S., C. Eastman, R. Sacks, and I. Kaner (2009). "Benchmark tests for BIM data exchanges of precast concrete". In: *Automation in construction* 18.4, pp. 469–484. doi: 10.1016/j.autcon.2008.11.001.

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