RESEARCH ARTICLE

Reference study of IFC software support: The GeoBIM benchmark 2019—Part I

Francesca Noardo[1] | Thomas Krijnen[1] | Ken Arroyo Ohori[1] |
Filip Biljecki[2,3] | Claire Ellul[4] | Lars Harrie[5] | Helen Eriksson[5] |
Lorenzo Polia[6] | Nebras Salheb[1] | Helga Tauscher[7] |
Jordi van Liempt[1] | Hendrik Goerne[8] | Dean Hintz[9] | Tim Kaiser[7] |
Cristina Leoni[10] | Artur Warchol[11,12] | Jantien Stoter[1]

[1]3D Geoinformation, Delft University of Technology, Delft, The Netherlands

[2]Department of Architecture, National University of Singapore, Singapore, Singapore

[3]Department of Real Estate, National University of Singapore, Singapore, Singapore

[4]Department of Civil, Environmental and Geomatic Engineering, University College London, London, UK

[5]Department of Physical Geography and Ecosystem Science, Lund University, Lund, Sweden

[6]Architect, Turin, Italy

[7]Faculty of Spatial Information, Dresden University of Applied Sciences, Dresden, Germany

[8]GMX, Munich, Germany

[9]Safe Software, Surrey, Canada

[10]Department of Civil, Constructional and Environmental Engineering, Sapienza Univerisity of Rome, Rome, Italy

[11]Institute of Technical Engineering, PWSTE Bronislaw Markiewicz State University of Technology and Economics in Jaroslaw, Jaroslaw, Poland

[12]ProGea 4D sp. z.o.o., Kraków, Poland

Correspondence
Francesca Noardo, 3D Geoinformation, Delft University of Technology, Delft, The Netherlands.
Email: f.noardo@tudelft.nl

Abstract

Industry Foundation Classes (IFC), the buildingSMART open standard for BIM, is underused with respect to its promising potential, since, according to the experience of practitioners and researchers working with BIM, issues in the standard's implementation and use prevent its effective use. Nevertheless, a systematic investigation of these issues has never been carried out, and there is thus insufficient evidence for tackling the problems. The GeoBIM benchmark project is aimed at finding such evidence by involving external volunteers, reporting on

various aspects of the behavior of tools (geometry, semantics, georeferencing, functionalities), analyzed and described in this article. Interestingly, different IFC software programs with the same standardized data sets yield inconsistent results, with few detectable common patterns, and significant issues are found in their support of the standard, probably due to the very high complexity of the standard data model. A companion article (Part II) describes the results of the benchmark related to CityGML, the counterpart of IFC within geoinformation.

1 | INTRODUCTION

In the architecture, engineering and construction (AEC) fields, as well as in other disciplines, interoperability is of increasing importance, in order to enable reuse and exchange of data and information, including in the strictly related asset and facility management field. Furthermore, it is essential for reciprocal integration of different types of data. One of the current research topics is, for example, the integration of building information models (BIMs) with 3D city models, for supporting several use cases (e.g., building permits issuing, 3D cadastre, complex assets and facility management). International standards are conceived as a solution for fostering such interoperability. The most popular standard data models, considered for obtaining such integration, are usually the OGC CityGML for 3D city models and buildingSMART Industry Foundation Classes (IFC) for BIMs.

However, although some building designers and professionals, supporting openBIM, are IFC enthusiasts, many others working with BIMs still seldom use IFC as a first choice for exchanging their models, and only export them to IFC when required explicitly by law, when trying to integrate with software from different vendors, or when no other solutions are available, as emerged from several personal communications. This often happens because they are aware of some limitations in the IFC format—as documented, for example, more than 10 years ago by Pazlar and Turk (2008), or in Barbini, Malacarne, Massari, Monizza, and Matt (2019)—although they might acknowledge its potential as an open exchange format.

In practice, limited support for IFC might not been considered as a major issue, since some very widespread software, such as the *Autodesk Revit* format, is often used as a *de facto* standard for exchange of BIMs among designers, as well as for integrating the BIM with further systems (see, for example, Aksamija, 2018; Baik, Yaagoubi, & Boehm, 2015; Farid Mohajer & Aksamija, 2019; Kamari, Laustsen, Petersen, & Kirkegaard, 2018; Kensek, 2014; Lamartina & Sa, 2019; Lv, 2018; Papadopoulos, Sotelino, Martha, Nascimento, & Faria, 2017; Petersen, Bøving, Nielsen, & Purup, 2018), while the actual use of BIMs by authorities, is still limited for the moment, apart from for visualization. This solves the most immediate interoperability issues, even though the *Revit* format is a proprietary binary format, which hinders reusability in an open way and across time.

A specific study of the limitations of IFC was part of the GeoBIM benchmark project (https://3d.bk.tudelft.nl/ projects/geobim-benchmark/, see Section 2) and is reported in this article. Within the project, the approach to the study of the support for the two standards involved in the GeoBIM integration (IFC and CityGML) was conceived

in parallel, also with the aim of understanding whether one of the two offered more effective solutions that could possibly be borrowed by the other one in future developments. However, the final outcomes of the two different tasks are very specific for each standard and deserve to be presented and discussed separately, considering the specifics of each case. For these reasons, this article, which focuses on the results about the benchmark Task 1 (support for IFC), is written in tandem with Noardo et al. (2020c), which describes Task 3, covering the support for CityGML. In order to allow each article to be read on its own, the two articles share some information (i.e., Section 2 explaining the general context and motivation of the study; Section 3. 1 covering the initial part of the methodology about the entire GeoBIM benchmark set-up, and Section 3. 3 concerning some similarities in the methodology). One further article explores the parts of the project more directly related to the subject of integration, namely, conversion procedures and useful tools to georeference IFC models Noardo, Harrie, et al. (2020).

2 | THE GeoBIM NEEDS AND THE IDEA BEHIND THIS STUDY

Two kinds of 3D information systems have been developed, studied and used in recent times, revealing their potential in related fields:

- *3D city models*, which are used to represent city objects in three dimensions and advance previous 2D maps and other cartographic products, in order to support city analysis and management, city planning, navigation, and so on (e.g., Bartie, Reitsma, Kingham, & Mills, 2010; Biljecki, Stoter, Ledoux, Zlatanova, & Çöltekin, 2015; Egusquiza, Izkara, & Gandini, 2018; Jakubiec & Reinhart, 2013; Liang, Gong, Li, & Ibrahim, 2014; Kumar, Ledoux, Commandeur, & Stoter, 2017; Peters, Ledoux, & Biljecki, 2015; Nguyen & Pearce, 2012);
- *building information models*, which are used in the architecture, engineering and construction fields to design and manage buildings, infrastructure and other construction works, and which also have features useful to project and asset management (e.g., Azhar, 2011; Haddock, 2018; Petri, Kubicki, Rezgui, Guerriero, & Li, 2017).

Several international standards exist to govern the representation of the built environment in a shared manner, to foster interoperability and cross-border analysis and, consequently, actions, or to reuse tools, analysis methods and data themselves for research and, possibly, government. Some examples of international standards are: the European Directive for an infrastructure for spatial information in Europe (INSPIRE) (https://inspire.ec.europa. eu), aimed at the representation of cross-border territories in Europe, for common environmental analysis; the Land and Infrastructure standard (LandInfra) (https://www.ogc.org/standards/landinfra), by the Open Geospatial Consortium (OGC), aimed at land and civil engineering infrastructure facilities representation; and the green building data model (gbXML) (https://www.gbxml.org), aimed at the representation of buildings for energy analysis.

Nonetheless, the two dominant reference open standards for those two models are CityGML (citygmlwiki. org), by the OGC, focusing on urban-scale representation of the built environment, and the Industry Foundation Classes (ISO, 2013, https://technical.buildingsmart.org/standards/ifc/), by buildingSMART, aimed at the very detailed representation of buildings and other construction works for design and construction objectives, first, but also intended to enable project management throughout the process, and asset and facility management in a following phase. Those standards are both intended to be very comprehensive and are therefore very wide and articulated. They both use complex data models allowing for a wide variety of models using object-oriented representations, even if that comes at the cost of slower and more inconsistent implementations.

Due to the overlapping interests in both fields (meeting in the building-level representation), 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 enables 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 (Aleksandrov, Diakite, Yan, Li, &

Zlatanova, 2019; Arroyo Ohori, Diakite, Krijnen, Ledoux, & Stoter, 2018; Fosu, Suprabhas, Rathore, & Cory, 2015; Kang & Hong, 2015; Kumar, Labetski, Arroyo Ohori, Ledoux, & Stoter, 2019; Lim, Tauscher, & Biljecki, 2019; Liu et al., 2017; Niu, Yang, & Pan, 2019; Noardo, Ellul, et al., 2020; Stouffs, Tauscher, & Biljecki, 2018, Sun, Olsson, Eriksson, & Harrie, 2020). The GeoBIM subject can be divided into several sub-issues:

1. The harmonization of data themselves, which have to concretely fit together, with similar (or harmonizable) features (e.g., accuracy, kind of geometry, amount of detail, kind of semantics, georeferencing).
2. Interoperability, which is a fundamental key in the integration. It is important to note here, that before enabling the interoperability among different formats (e.g., GIS and BIM), which is the theme of point 3 below, the interoperability GIS-to-GIS and BIM-to-BIM itself is essential. That means that the formats of data have to be understood and correctly interpreted uniquely by both any person and any supporting software. Moreover, an interoperable data set is supposed to remain altogether unchanged when going through a potentially infinite number of imports and exports by software tools, possibly converting it to their specific native formats and exporting it back.
3. The effective conversion among different formats, that is, transforming one data set in a (standardized) format to another one in compliance with the end format specifications and features.
4. The procedures employing 3D city models and the ones based on BIM should be changed in order to obtain better advantages by the use of both, integrated, since those systems enable processes which are usually more complex than just the simple representations.

The many challenges implied by the points above are still far from being solved, and one of the essential initial steps is actually to outline such challenges more sharply.

In particular, the second point (interoperability and involved standards) is often considered to be solved by standardization organizations. It is desirable to rely on open standards for this, because the well-documented specifications of open standards would enable longer-term support, as well as their genericity with respect to different software vendors, as opposed to closed point-to-point solutions that merely connect one proprietary system to another (and might be discontinued or stop working at any moment). However, our previous experiences suggest that the support for open standards in software often includes shortcomings.

The researchers promoting this study (as users of data, advocates of open standards and developers of tools adopting such standards) have noticed, over their research and work activities, how the use of those standards in data and their implementations in software were not always straightforward and not completely consistent with the standard specifications either. Many tools, when managing standardized data, do not adequately support features or functionalities as they do when the data is held in the native formats of the software. In addition, software tools have limitations with respect to the potential representation (geometry, semantics, georeferencing) of data structured following these standards, or can generate errors and erroneous representations by misinterpreting them.

The standards themselves are partly at fault here, since they often leave some details undefined, with a high degree of freedom and various possible interpretations. They allow high complexity in the organization and storage of the objects, which does not work effectively towards universal understanding, unique implementations and consistent modeling of data. This is probably due to the fact that such standards often originate as amalgamations of existing mechanisms and compromises between the various stakeholders involved. These experiences have been informally confirmed through exchanges within the scientific community and especially with the world of practitioners, who are supposed to work with (and have the most to gain from) those standardized data models and formats. However, more formal evidence on the state of implementation of these open standards and what problems could be connected to the standard themselves have not been compiled so far.

For this reason, the GeoBIM benchmark project (https://3d.bk.tudelft.nl/projects/geobim-benchmark/, https://www.isprs.org/society/si/SI-2019/TC4-Noardo_et_al_WG-IV-2-final_report.pdf) was proposed and

funded in 2019 by the International Society for Photogrammetry and Remote Sensing (ISPRS, https://www.isprs.org) and the European association for Spatial Data Research (EuroSDR) (http://www.eurosdr.net). The aim of the benchmark was to get a better picture of the state of software support for the two open standards (IFC and CityGML) and the conversions between them, in order to formulate recommendations for further development of the standards and the software that implements them. In addition, we tested two known major technical issues related to GeoBIM integration and which are known to be solved only partially in practice: the ability of tools and methods to georeference IFC and the conversion procedures between IFC and CityGML (Noardo, Harrie, et al., 2020).

The relevant outcomes regarding the support of software for buildingSMART IFC standard are the subject of this article.

2.1 | Industry Foundation Classes and knotty points

The buildingSMART IFC standard (https://technical.buildingsmart.org/standards/ifc/) is an open standard data model for building information modeling to be shared and exchanged through software applications, domains and use cases, within the AEC and facility management (FM) fields. It includes classes for describing both physical and abstract concepts (e.g., cost, schedule) mainly concerning AEC-FM for buildings (recent versions are extending it to include infrastructure and other kinds of construction, https://technical.buildingsmart.org/standards/ifc/ifc-schema-specifications/). It has also been adapted as the ISO 16739 international standard (ISO, 2013).

The standard includes relevant constructs for a wide variety of disciplines, use cases and processes associated with the construction domain, most prominently the semantic description and geometric representation of typical construction elements and their relationships.

The IFC is structured in a deeply hierarchical data model, and organized in several and deep and complex meronymic (part-of) trees. Spatial composition (site/building/storey/space/zone) is one more kind of aggregation, different from element (meronymic) composition (e.g., a stair and the assembled elements in it). Moreover, nesting is a slightly different kind of element composition, representing the products which are specifically designed as complementary by manufacturers. Finally, subtraction relationships are also part of the IFC model, representing openings by means of the voiding mechanism. A great number of further relationships are added to this complexity (e.g., to associate materials, geometric representation or other property information).

An additional complexity to the semantic model is given by the possibility to store the same kind of object by means of several entities. For example, the layers within a compound wall object can be represented by means of an associated `IfcMaterialLayerSet`, but also as a more generic decomposition where every wall layer is modeled as a distinct `IfcBuildingElementPart`.

A great number of attributes can be associated with entities (and inherited by the parent ones), both directly or through property sets.

All this semantic complexity is intended to faithfully represent the buildings as functional to the standard designed scope. However, the implementation and use of such a theoretically precise model are difficult and can result in inaccuracies or underuse of it, besides hindering interoperability for leaving too much freedom in filling the information in and in choosing the kind of representation to be used.

Additional terms which can be used in IFC are defined within the buildingSMART Data Dictionaries (bsDD) and are modeled according to the International Framework for Dictionaries (IFD) (http://bsdd.buildingsmart.org). It is based on the standard ISO 12006–3.

The IFC current versions are IFC2 × 3, which was released in 2005 (with the latest corrigendum in 2007) and IFC4.1 from 2018. At the time of writing, the most frequently implemented and used version is still (by far) IFC2 × 3. For this reason, both versions were considered in this study.

A third part of the standards is the Information Delivery Manual (IDM), which defines the workflow and the information exchange specifications and requirements in the processes involved in the construction life cycle.

In an IDM, a set of model view definitions (MVD) can be defined for identifying the portion of the IFC model which is needed for a particular information exchange in the IDM to be fulfilled. This can define a use-case-oriented part of the wide IFC model, to be implemented in software.

It is interesting to note that this mechanism is in a way opposite to CityGML Application Domain Extensions (ADEs, Biljecki, Kumar, & Nagel, 2018). In an IFC MVD a subset of definitions is selected from the monolithic schema, instead of an addition. On the other hand, for property set definitions, the usage of MVD and ADE is analogous, as both nominate additional attributes.

IFC derives many aspects from ISO (2014), informally known as STEP. The majority of geometry definitions are derived from ISO 10303–42 and the typical exchange formats are based on STEP Physical File (SPF, ISO 10303–21) and an XML encoding (ISO 10303–28).

Parametric modeling is usually employed in BIMs and IFC, which makes it possible to encode many kinds of geometries. This includes Boolean operations and complex sweeps, for example the sweep of an arbitrary profile along a curve while constraining the normal vector. Also explicit geometries are supported in the form of boundary representations and (added in IFC4) efficient support for triangulated meshes. The implementation of the former type of geometry is that supporting the full stack of geometric procedures in IFC is a major implementation effort and due to implementation choices can sometimes lead to different results in importing applications. The complexity can, therefore, have consequences on interoperability and the way different pieces of software read and re-export the same geometry.

2.1.1 | IFC geometries

The particular issues described in this section guided the methodological choice of providing a specific set of IFC geometries among the benchmark data sets, in order to point out specific related behavior (Figure 1 is an example of uncontrolled geometry transformations within some of the tested software).

IFC is a complex standard and the fidelity of both importers and exporters is still evolving. Implementations are based on varying paradigms and geometric modeling principles: boundary representation, polyhedra or triangular meshes. Some implementations use so-called toolkits that offer complete support for STEP and EXPRESS (the modeling language in which the IFC schema is specified; with an embedded complex Turing-complete constraint language to specify WHERE rules and *FUNCTIONS*) Other implementations use custom in-house developments and, for that reason, typically have less support for the automatic validation of IFC models based on the WHERE rules. The IFC schema is evolving as well,[1] mostly aiming at the improvement of reliability, reuse of industry-standard best practices and the reduction of implementation effort.
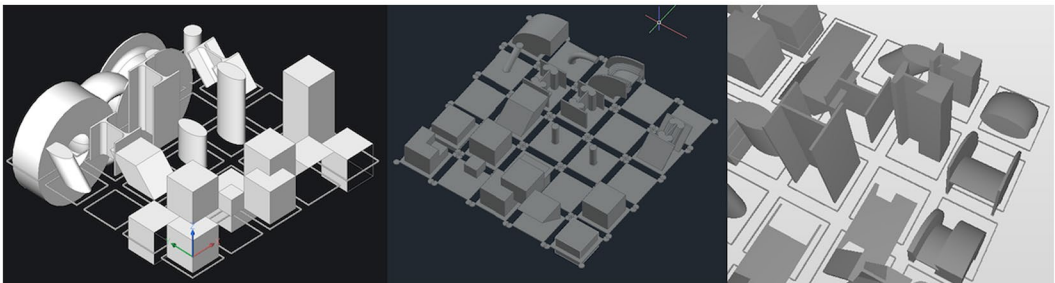
FIGURE 1 Example of unusual behavior of the geometries in some of the tested software

Given these considerations, the permutations of IFC definitions (some of which assessed by this set of analytical geometries) span a three-dimensional matrix along the axes (Figure 2) that specify:

1. whether the situation is known to be exported by authoring tools;
2. whether the situation is handled successfully by importing applications; and
3. whether that situation is valid according to the IFC schema.

As such we can summarize this state in a tuple of three elements. For example, for a situation that is exported by authoring tools, successfully imported, but not valid according to the schema, the tuple would be ⟨Y, Y, N⟩.

From these states we want to highlight three that are meaningful:

1. ⟨N, Y/N, Y/N⟩ is a situation that is never exported and can hence be removed from the schema to reduce the implementation effort of a fully compliant application.
2. ⟨Y, Y, N⟩ are situations that are successfully imported, but not valid according to the schema; schema constraints may be loosened to actually allow such situations.
3. ⟨Y, N, Y/N⟩ is the biggest problem for practitioners—geometries that are exported, but not successfully imported (whether valid or not).

An additional concern is that while validity according to the schema is fairly constant and universal, import success depends on the application used, and with many different tools used by practitioners, the diverging results of importers may cause misconceptions and disputes. Also note that validity is not as universal as it might seem at first glance. The IFC schema defines a point equality tolerance (*IfcGeometricRepresentationContext.Precision*) that defines "the tolerance under which two given points are still assumed to be identical". How this is handled in geometry kernels is implementation dependent. Additionally there are WHERE rules that specify that an extrusion length needs to be positive (*IfcExtrudedAreaSolid.Depth* is of type *IfcPositiveLengthMeasure* which has a rule `WR1 : SELF > 0`.) but at the same time EXPRESS does not specify restrictions on the floating-point number type, so an infinitesimal value can be supplied that is greater than zero, but when mapped to the number type in the importing application cannot be distinguished from it. Or an extrusion depth can be supplied that is below the model tolerance and therefore may fail in applications that adhere to this tolerance.
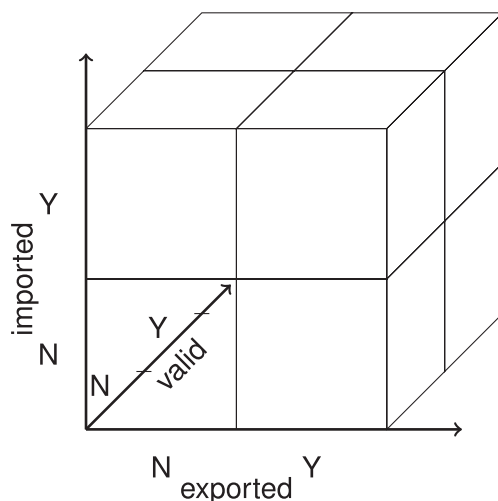
FIGURE 2 Three-dimensional matrix representing the IFC geometry conditions: can be exported/can be imported/is valid according to the IFC standard

A significant set of geometries with a variation of such three features was added as part of the tested IFC data sets, as described in Section 3.2.1.

2.1.2 | IFC georeferencing

Proper georeferencing of an IFC file facilitates a link between the local Cartesian coordinate system used by IFC, with their corresponding real-world coordinate reference system (CRS), and thus to place the model of a single building or construction within its context and environment. There are several options to store georeferencing information in IFC, with varying level of detail. These options range from basic address information over the specification of the geolocation of a reference point to the definition of an offset between the project coordinate system and the global origin of a coordinate reference system and the corresponding rotation of the *XY*-plane. Some of the georeferencing options are described and classified into Levels of Georeferencing (LoGeoRefxx) by Clemen and Gorne (2019) (Table 1). This classification scheme is not officially defined in the IFC standard but can help practitioners to assess which georeferencing information is available in the IFC file. It is common for the options presented that coordinates for a reference point (which is usually the origin of the local Cartesian system where the model is designed) are stored and sometimes complemented with a direction of the axis in the local system.

The georeferencing of BIM has not been a priority for designers and software developers. Therefore georeferencing information is not regularly stored and used, and not always read and exported in a completely consistent way. Moreover, the available tools to georeference BIMs are not perfectly optimized to acquire and store the correct information in the foreseen attributes and entities in the IFC files. For this reason, the IFC georeferencing,

TABLE 1 List of georeferencing options in IFC classified as *LoGeoRefs* (Clemen & Gorne, 2019)

LoGeoRef	Supported CRS	Storing entities
LoGeoRef10	No CRS, approximate location by means of the address	*IfcPostalAddress* referenced by either *IfcSite* or *IfcBuilding*
LoGeoRef20	WGS84 EPSG:4326	Attributes *RefLatitude, RefLongitude, RefElevation* within *IfcSite*
LoGeoRef30	Any Cartesian CRS, including projected coordinates (CRS not specified in the file)	*IfcCartesianPoint* referenced within *IfcSite* (defining the projected coordinates of the model reference point);*IfcDirection* attribute of *IfcSite* (stores rotations regarding project or global north)[a]
LoGeoRef40	Any Cartesian CRS, including projected coordinates (CRS not specified in the file)	Attribute *WorldCoordinateSystem* storing the coordinates of the reference point in any Cartesian CRS (including the projected ones) and direction *TrueNorth*. Both are stored within *IfcGeometricRepresentationContext*[b]
LoGeoRef50	Specific projected CRS, specified by means of the EPSG code	IFC v.4 only. Coordinates of the reference point stored in *IfcMapConversion* using the attributes *Eastings, Northings* and *OrthogonalHeight* for global elevation. Rotation for the *XY*-plane stored using the attributes *XAxisAbscissa* and *XAxisOrdinate*. The CRS used is specified by *IfcProjectedCRS* in the attribute *Name* by means of the proper EPSG code

Note: EPSG is a coding list for CRS (https://epsg.io/).

[a]Ad-hoc solution used by several tools.

[b]Most official IFC2 × 3-way to store the reference system.

which is fundamental for GeoBIM integration, was investigated within the benchmark, from two main points of view.

On the one hand, the ability of software to correctly interpret the georeferencing information provided within the data sets was checked, with the focus on georeferencing levels LoGeoRef20 and LoGeoRef30. Since LoGeoRef30 is more accurate, allowing the storage of Cartesian projected coordinates, we asked participants to check if this one could be read and interpreted by software correctly, within Task 1 (see Section 3.1).

On the other hand, it is quite unexplored if and how the tools can edit the georeferencing information of IFC models: normally there is little or no control of how georeferencing information is stored in the final model. For this reason, we tested this issue separately, within benchmark Task 2 (see Section 3.1; see also Noardo, Harrie, et al., 2020).

3 | METHODOLOGY

3.1 | The GeoBIM benchmark general set-up

The benchmark was intended as a way to combine the expertise of many people with different skills, coming from several fields and interests, in order to describe the present ability of current software tools to use (i.e., read, visualize, import, manage, analyze, export) CityGML and IFC models and to understand their performance while doing so, in terms of both information management functionalities, and possible information loss. Moreover, since the large size of such standardized data sets often generates difficulties in their computational management, the ability to handle large data sets was a further part of the tests.

In particular, the four topics investigated in the benchmark are:

Task 1. What support is there for IFC within BIM (and other) software?[2]
Task 2. What options for georeferencing BIM data are available?
Task 3. What support is there for CityGML within GIS (and other) tools?
Task 4. What options for conversion (software and procedural) (both IFC to CityGML and CityGML to IFC) are available?

For this purpose, a set of representative IFC and CityGML data sets were provided (Noardo, Arroyo Ohori, et al., 2019) and used by external, voluntary participants in the software they would like to test in order to check the support in it for the open standard considered (Noardo, Biljecki, et al., 2019).

Full details about the software tested and a full list of participants can be found in the respective pages of the benchmark website.[3] The significant number of participants, balance in skills, fields of work, levels of confidence about the software tested (asked them to be declared) offered the possibility to limit the bias in the results.

The participants described the behavior of the tools tested following detailed instructions and delivered the results in a common template with specific questions, provided as online forms. In the end, they delivered both their observations and the models as re-exported back to the original standardized format (CityGML or IFC).

In order to cover the widest part of the list of software potentially supporting the investigated standards, we completed the testing ourselves, by searching the online documentation of both the standards and the potential software.

In the final phase of the project, the team coordinating the study analyzed the participants' observations, descriptions and delivered further documentation (screenshots, log files, related documents and web pages). From this review, an assessment of the performances and functionalities of the tools tested was derived. Moreover, the models delivered were validated and analyzed using available tools, when possible, and/or through manual

inspection (Section 3.3). This approach enabled the inquiry about the level of interoperability given by the standard and its software implementation, by comparing the results of the export with the imported model features.

It is important to notice that the test results are not intended to substitute the official documentation of each software program. Moreover, there were no expertise nor skill requirements to participate in the benchmark tests. Therefore, some information could be wrong or inaccurate, due to little experience with the software tested or the topics managed. The declaration of the level of expertise was intended to lower this possible bias. Moreover, the benchmark team and the authors tried to double-check the responses (at least the most unexpected ones) as much as possible, but the responses reported in the data were generally not changed from the original ones. Any discrepancies between the best potential software performances and what was tested could anyway be showing a low level of user-friendliness of tools (and thus a degree of difficulty in achieving the correct result).

3.2 | The IFC data sets provided

We have selected IFC data sets allowing the testing of the main issues regarding the support of IFC (Table 2): one data set was the architectural IFC model of a small two-floor building, intended to test the support for IFC features in the most reliable way (*Myran.ifc*). The second one was a large and complex architectural IFC model of a big tower in Rotterdam (*UpTown.ifc*), intended to test the software's performance; and the last one was the architectural IFC model of a medium-size building in IFC4 (*Savigliano.ifc*), which was chosen to test also the support for IFC4 files and any discrepancies with the management of the other data sets, which were provided in IFC version 2 × 3 (which our previous experience suggests is better supported). All the IFC files where originally modeled in *Autodesk Revit* and thereafter exported as IFC files.

Moreover, since performing tests of IFC geometries could be tricky in complex files combining many geometry types in an uncontrolled manner, we chose to monitor this aspect better by means of synthetically generated geometries (using *IfcOpenShell*, http://ifcopenshell.org) to test the support of software for specific kinds of geometry modeling (see Section 3.2.1 for more details). Those geometries are provided in both IFC2 × 3 and IFC4.

TABLE 2 Summary of descriptions, features and aims of the IFC data sets provided

Name	Description	IFC version	Size (MB)	Source	Aim
Myran.ifc	Model of a small 2-floor building in Sweden, by Swedish architects. Georeferenced	2 × 3	27.14	Mondo arkitekter, (SE)	Test of the main functionalities of software and common procedures
UpTown.ifc	Model of a big complex tower in Rotterdam, by Dutch architects.	2 × 3	241.04	Municipality of Rotterdam (NL)	Test of the software's performance
Savigliano.ifc	Model of a building in Italy, by an Italian architect within a research environment	4	21.55	Arch. Lorenzo Polia (IT)	Test of the support for IFC v.4 and to enable the tests of procedures and tools working with IFC v.4
Specific IFC geometries	Set of geometries modeled using a range of modeling alternatives allowed in IFC (see Section 3.2.1)	2 × 3; 4	0.31 (v.2 × 3); 0.27 (v.4)	Synthetically generated with *IfcOpenShell*	Test of the support and behavior of software with respect to these specific geometries

3.2.1 | IFC geometry sets

The geometries used in the BIMs can have a huge numbers of variations and a complete validation of them, as used in models, is still an unsolved task. Moreover, IFC admits a number of geometry types that can be useful to modelers, but are sometimes not supported or can be interpreted in different ways by software. On the other hand, IFC puts validity constraints on certain geometries, but some software implements workarounds aiming at reading those invalid geometries too, which are often undocumented, and there is therefore often little possibility to keep track of these solutions. For these reasons and in order to explore the issues detailed in Section 2.1.1, a specific set of geometries was provided among the benchmark data sets (see Figure 3 and Table 3).

The structure of the results template guiding the test about such data sets is as follows, for every geometry in the set. First there is a general question whether an object is displayed in that slot after import. Then there are questions about the positioning relative to the $Z = 0$ plane (mostly to distinguish how the negative extrusion depth is handled), the rendering of curved surfaces and what shape is displayed (to identify some specific situations such as the non-normalized extrusion direction).

3.3 | Analysis of responses about the support for IFC

The methodologies for analyzing the results about the support of software for IFC (Task 1) and CityGML (Task 3) are very similar, since they were also conceived to test similar issues concerning interoperability and the ability of software to keep files consistent with themselves after import–export phases.

The initial part of results analysis (Section 4.2) is qualitative, providing the description of software support and functionality based on the responses delivered.

The complete responses and documents delivered in the online templates (https://doi.org/10.5281/zenodo.3964445, Noardo et al., 2020b) were double-checked for correctness and consistency with respect to
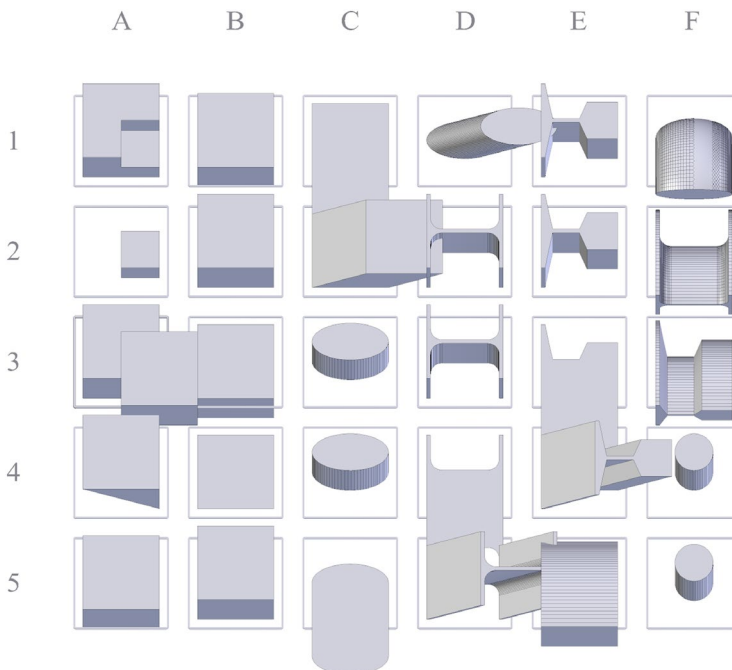
FIGURE 3 The data set with synthetically generated IFC geometries

TABLE 3 Description of each object included in the IFC geometries set, in both IFC2 × 3 and IFC4 (the leftmost column refers to the row and column in Figure 3)

	IFC definition	Description
A1	IfcBooleanResult_1	Result of boolean subtraction with two cube operands with partial overlap
A2	IfcBooleanResult_2	Result of boolean intersection with two cube operands with partial overlap
A3	IfcBooleanResult_3	Result of boolean union with two cube operands with partial overlap
A4	IfcBooleanClippingResult_1	Result of boolean clipping operation with a cube and a halfspace operand
A5	IfcShellBasedSurfaceModel_1	A shell-based surface model, an explicit collection of faces
B1	IfcFacetedBrep_1	A faceted boundary representation, an explicit collection of faces
B2	IfcExtrudedAreaSolid_1	Extrusion of a rectangular profile
B3[a]	IfcExtrudedAreaSolid_2	Extrusion of a rectangular profile, negative extrusion depth
B4[a]	IfcExtrudedAreaSolid_3	Extrusion of a rectangular profile, zero extrusion depth
B5	IfcExtrudedAreaSolid_4	Extrusion of a rectangular profile, non-normalized direction vector
C1[b]	IfcExtrudedAreaSolid_7	Extrusion of a rectangular profile, direction vector parallel to profile
C2	IfcExtrudedAreaSolid_10	Extrusion of a rectangular profile, slanted direction vector
C3	IfcExtrudedAreaSolid_13	Extrusion of an elliptical profile
C4	IfcExtrudedAreaSolid_16	Extrusion of an elliptical profile, non-normalized direction vector
C5[b]	IfcExtrudedAreaSolid_19	Extrusion of an elliptical profile, direction vector parallel to profile
D1	IfcExtrudedAreaSolid_22	Extrusion of an elliptical profile, slanted direction vector
D2[f]	IfcExtrudedAreaSolid_25	Extrusion of an I-shaped profile
D3[f]	IfcExtrudedAreaSolid_28	Extrusion of an I-shaped profile, non-normalized direction vector
D4[fb]	IfcExtrudedAreaSolid_31	Extrusion of an I-shaped profile, direction vector parallel to profile
D5[f]	IfcExtrudedAreaSolid_34	Extrusion of an I-shaped profile, slanted direction vector
E1	IfcExtrudedAreaSolid_37	Extrusion of a crane rail (A-shaped) profile
E2[d,e]	IfcExtrudedAreaSolid_40	Extrusion of a crane rail (A-shaped) profile, non-normalized direction vector
E3[b,d,e]	IfcExtrudedAreaSolid_43	Extrusion of a crane rail (A-shaped) profile, direction vector parallel to profile
E4[d,e]	IfcExtrudedAreaSolid_46	Extrusion of a crane rail (A-shaped) profile, slanted direction vector
E5	IfcRevolvedAreaSolid_1	Revolution of a rectangular profile
F1	IfcRevolvedAreaSolid_2	Revolution of an elliptical profile
F2[f]	IfcRevolvedAreaSolid_3	Revolution of an I-shaped profile
F3[d,e]	IfcRevolvedAreaSolid_4	Revolution of a crane rail (A-shaped) profile
F4	IfcSweptDiskSolid_1	Swept disk
F5[c]	IfcSweptDiskSolid_2	Swept disk with parameter range outside of curve definition

[a]Not valid due to violation of the positive length measure requirement on IFcEXTRUDEDAREASOLID.DEPTH.

[b]Not valid due to violation of the where rule VALIDEXTRUSIONDIRECTION.

[c]Not valid as the parameter range for the sweep is outside the parametric range of the curve.

[d]Not included in the IFC4 dataset due to removed entities in the schema.

[e]Not part of the Coordination View 2.0 Model View Definition in IFC 2 × 3.

[f]This shape should have fillet radii, but not all viewing software displays those.

the questions asked. However, due to the nature of the initiative, we trusted the delivered information about the software, double-checking it with new tests only in cases of inconsistent responses in different tests about the same software, or possibly unexpected responses. In these cases, we also considered the level of expertise of the participant to assess if further checks were actually needed.

The responses delivered in the templates were critically assessed and cross-checked with the different tests about the same software and the attached screenshots. A score for each aspect considered for the assessment of general support and software functionalities was assigned (1, full support; 0.5, partial support; 0, no support). Those were synthesized in a table (Table 5) in order to more easily deduce possible patterns across many issues for a single software package or across many software packages for a single issue.

The definition of software groups is growing increasingly fuzzy, since their functionalities are continuously being extended and now tend to overlap with each other. However, in the tables, and more generally in the analysis, in order to help the detection of possible patterns, the software tested was classified according to the criteria that usually guide the choices made by users, based on their different needs for specific tasks:

- *GIS* are expected to combine different kinds of geodata and layers and carry out analysis on them, structured in a database, in a holistic system.
- *"Extended" 3D viewers* are likely software that was originally developed for visualizing 3D semantic models, including georeferencing, and query of them. They were (sometimes later) extended with new functions for applying symbology or making simple analysis.
- *Extract transform and load (ETL)* software, and conversion software, is expected to apply some defined transformations or computations to data;
- *3D modeling tools* have good support for geometry editing, but are not intended to manage georeferenced data or semantics.
- *Analysis software* is intended specifically for few kinds of very specific analysis (e.g., energy analysis),
- *BIM* software is intended to design buildings or infrastructure according to Building Information Modelling methods.

The issues investigated, reflected in the different sections of the templates provided, regarded mainly the support of the software for the two standards (how the software reads and visualizes the data sets) and the functionalities allowed by the software with standardized data sets (what is it possible to do with such data). In particular, the test for the support was intended to check: how the georeferencing information in the files is read and managed (Section 4.2.1); how the semantics are read, interpreted and kept after importing (Section 4.2.2); and the state of the geometry after importing (Section 4.2.3).

The last questions were related to the kind of functionalities (Section 4.3) that are offered by the software for the management of IFC data:

- What kind of visualization is enabled (3D, 2D, with textures, with specific themes).
- What kind of editing is possible (attributes, geometry, georeferencing).
- What kind of query (query the single object to read the attributes, selection by conditions on attributes, spatial query, computation of new attributes).
- What analyses are allowed. This topic is more complex, since very different analyses are possible. Therefore we summarized it by defining two analysis types: "Type 1" is any kind of analysis regarding the model itself (e.g., geometric or semantic validation), and "Type 2" is the simulations and analyses of the performance of the object represented (e.g., a building) with respect to external factors, in the city or environment (e.g., shadow, noise, energy).
- Finally, is it possible to export back to IFC?

Moreover, the support for each of the data sets delivered was noted, given the specific features: IFC2 × 3 building model (*Myran.ifc*), IFC4 building model (*Savigliano.ifc*), very heavy model (*UpTown.ifc*).

This first part provides a reference about the tools themselves for people intending to use standardized information. In addition, the most challenging tasks and most frequent issues for the management of standards were supposed to be pointed out.

A second, more quantitative, part of the analysis considers the delivered models exported back to IFC from the software tested (Sections 4.4 and 4.5). The exported models of *UpTown.ifc* were visually inspected but could not be completely analyzed since the tools used (such as the *NIST IFC File Analyzer*) crashed without producing any useful result, probably given the large size of these data sets. For this reason the results for *UpTown.ifc* are not described in the article. The models can, however, be downloaded together with the other data from Noardo et al. (2020a, https://doi.org/10.5281/zenodo.3964368).

The numbers and types of features of such files were calculated and compared to the same features in the initial data sets that were provided for the test. The *NIST IFC File Analyzer* (https://www.nist.gov/services-resou rces/software/ifc-file-analyzer), calculating the number of IFC entities of each type in the data set, was used for this check. Moreover, manual inspection of the files within 3D viewers (*BIM Vision*, *RDF IfcViewer*) was used to check the apparent problems (e.g. missing elements) and further changes (e.g. grouping in storeys, transformation of building elements in *IfcBuildingElementProxy*, change to some kind of element to others). The inspection of the text format was also helpful in checking some elements (e.g. the more formal ones, documenting organization, applications, IFC version).

This allowed us to assess the level of interoperability that the connected standards-tools can actually achieve in the different cases: that is, can the data be imported and re-exported without any change?

A further assessment (Section 4.7) was intended to evaluate the software- and hardware-connected performance. The times declared by the testers were compared for each data sets to see how much their computational weight could affect their management within software.

Given the complexity of measuring software performance to the nearest second, this was not requested of the users. Instead, they were asked to provide an approximate timing value for each test, according to a classification that was proposed following the way they could affect the perception or the work of a user, as explained in the following list:

- It is almost immediate (good!).
- Less than a minute (ok, I will wait).
- 1–5 min (I can wait, if it is not urgent).
- 5–20 min (in the meantime I can do other things).
- 20 min–1 hr (I cannot rely on it for frequent tasks),
- More than 1 hr (I launch my process and go home, definitely ineffective for regular work).

Other options included reporting whether the software crashed or whether the task was not possible with the software provided, and participants were also asked to provide information about the specification of the machine, as this may impact overall performance of the software.

4 | RESULTS: SUPPORT OF SOFTWARE FOR INDUSTRY FOUNDATION CLASSES

4.1 | Software tested

For the benchmark we tested 31 different software packages (Table 4), to try to cover all possible IFC management solutions, although a huge number of software packages and other tools exist, considering that the BIM-related

TABLE 4 Software tools tested for support for IFC within the benchmark Task 1

	Open source	Proprietary	Freeware
GIS software		Bentley Map Enterprise [L1] ESRI ArcGIS Pro [L3]	
		ESRI ArcGIS Pro [L3]	
'Extended' 3D viewers	Datacomp Sp. z.o.o. BIM Vision RDF IFC Viewer [L1]	CSTB eveBIM [L4] CSTB eveBIM Viewer [L1] TeamSystem STR Vision IFC Viewer [L1] Solibri Anywhere [L1]	FZKViewer [L1+L2]
	RDF IFC Viewer [L1]	CSTB eveBIM Viewer [L1]	
		TeamSystem STR Vision IFC Viewer [L1] Solibri Anywhere [L1]	
		Solibri Anywhere [L1]	
ETL and conversion software		FME Desktop [3L3]	
3D modeling software (CAD)	FreeCAD [L1+L2]	Bentley MicroStation+TerraSolid [L3]	
	Bricsys Blender [L1]	Trimble SketchUp [L1]	
		Cadwork Lexocad [L1]	
		Bricsys BricsCAD Ultimate [L1]	
Analysis software		ACCA PriMus-IFC [L1]	
		Dlubal RFEM [L1]	
BIM software	BIMServer [L2]	Autodesk Civil 3D [L1]	
		Solibri Office [L1]	
		Tekla Structures [L1]	
		ACCA usBIM.viewer+ [L1]	
		CadLine Ltd ARCHline.XP [L1]	
		Simplebim [L1]	
		Allplan [L1]	
		AutoCAD Architecture [L1]	
		ACCA Edificius [L1]	
		Autodesk Revit [3L3+3L1+1L2]	
		Vectorworks Designer [2L1]	
		Graphisoft ArchiCAD [2L1+L2]	
		DDS-CAD [L1]	
		Autodesk Infraworks [L1]	

Note: Level 1 of participant expertise is abbreviated L1, etc.

industry is quite developed and still growing. In the table, they are organized based on the kind of software and divided into open source, proprietary and freeware (but not open source) software. Moreover, the levels of expertise of participants making the tests (from Level 1, the least expert, to Level 4, the most expert) are also reported.

Some of the software packages were tested several times, with different levels of expertise: 2 tests with *eveBIM* (Level 1, beginner, and Level 4, developer); 2 tests with *FZKViewer* (Levels 1 and 2); 3 tests with *Safe Software FME* (all experts); 2 tests with *FreeCAD* (Levels 1 and 2); 3 tests with *ArchiCAD* (Levels 1 and 2); and many tests with *Revit*, as expected, this being very popular software (Levels 1, 2 and 3).

In addition, more (unsuccessful) tools were considered, for example within the *Tekla* suite, *Tekla Structure* is supposed to be the only tool supporting IFC (and finding the trial versions of others was tricky anyway); *Autodesk Fusion*, which, however, is not supposed to support IFC; *bimspot* (https://bimspot.io) crashed when trying to import any of the data sets.

Other software, found on the internet or initially proposed by the participants, that was found to be unable to support IFC included *ACCA Solarius* and *Solarius PV*, *iTown*, *BimView* and *bimvie*. There were also issues with finding the trial versions of other software, such as *BuildingReconstruction*, *OpenDesignAlliance IFC SDK*, only allowing some of the tools to be tried, or not reading IFC correctly (e.g., other *Bentley* software, *ACCA EdiLus*). However, the most popular tools were covered and tested thoroughly.

4.2 | Software support for IFC

In Table 5, the more qualitative analysis of the participants' reports (https://doi.org/10.5281/zenodo.3964445, Noardo et al., 2020b) is synthesized. The results are discussed below, focusing on the themes of georeferencing, semantics and geometry.

4.2.1 | Load of IFC data: Georeferencing information

It is possible to notice that in most cases, georeferencing information, as well as elevation, is not read correctly and the models are moved to a local system having the origin at (0, 0, 0), using different measurement units (mainly metres and millimetres). In some cases it is possible to change this. For example in *eveBIM*, by default, it would load the model with a local coordinate system; it is, however, possible to change the settings to have it use the model's previously defined local coordinate system.

Some tools work only partially; for example, *FME* uses *RefLatitude RefLongitude* in *IfcSite*, although more accurate coordinates are assigned through an *IfcLocalPlacement* (LoGeoRef30) or the project's representation context (LoGeoRef40).

In other cases, the CRS is misinterpreted; for example *Bentley Microstation+Terrasolid* moved all the models to (0, 0, 0) in the CRS EPSG:3152. In other software, some georeferencing information is reported in very great detail, including datum, ellipsoid and so on. However, this information is not correct.

With some other software, like *Revit* or *ArchiCAD*, different tests give different responses, therefore it is possible that it depends on import settings.

To summarize, only 30% of the tools correctly use the georeferencing information with proper global coordinates (as provided by LoGeoRef30 or LoGeoRef40). Very few changes are reported for the tests with the IFC4 data set, which does not use the extended IFC4 georeferencing features anyway. Orientation of the model is instead kept correctly in most cases (90%), although it was difficult to assess if the added true north information was read.

4.2.2 | Load of IFC data: Focus on semantics

The semantics are read more consistently than the georeferencing information, with generally good support, although some inaccuracies are found, mostly consisting of: partial recognition of entities[4]; loss (or partial loss) of relationships;[5] and partial inconsistencies in the reading and interpretation or loss of attributes.[6]

Approximately, 80% of the tools manage semantics in a satisfactory way; the 3D viewers group especially performed well. However, many inaccuracies need to be solved to achieve good interoperability. In particular,

it is noticeable how *Revit* reports inconsistencies with respect to the original imported data, although being the software where all the data sets were modeled and from where they were exported.

4.2.3 | Load of IFC data: Focus on geometry

The software tools in this study are deemed to support the geometry if there are no apparent errors; no detailed study of geometry was conducted on the building models, since this is a very complex issue. However, reported errors generally regard the loss of some elements or their change so that intersections, deformations or modifications (e.g., in normals) occurred.[7]

In addition, Table 5 reports whether the software is able to load and work with the different data sets. Issues include: failures most likely connected to the computational requirements in the case of *Uptown.ifc*; to the IFC version in case of *Savigliano.ifc*; and perhaps to geometry issues in the case of *Myran.ifc*, which is not read correctly in a few cases.

For example, version 0.18 of *FreeCAD* is not able to import the *Myran.ifc* model, and one of the tests with *FZKViewer* fails with *Myran.ifc* as well. Other import errors are reported in Table 6.

4.2.4 | IFC geometry interpretation

The data set is synthetically generated for both the IFC2 × 3 schema and IFC4 schemas, with the IFC4 geometries being a subset of the ones supplied for IFC2 × 3. For that reason one would expect similar import success ratios for IFC2 × 3 and IFC4. This is not the case, as shown in Figures 4 and 5. A possible explanation is that applications supporting IFC4 are newer or better and therefore have higher success rates.

TABLE 5 Synthesis table of the delivered tests of support for IFC (benchmark Task 1)

Note: The color scale from green to red is assigned according to the scores from 0 (no support) to 1 (full support).

TABLE 6 Errors in the imports for some of the software tools

Software	Import error
FZKViewer	with *Savigliano.ifc*: The software crashes when trying to add an element
	with *Uptown.ifc*: Many errors reported
	with *Myran.ifc*: Error 776: boolean operation (clipping plane generation for CdgisModel geometry failed); geometry face (invalid outer loop); geometry polyline (polyline contains collinear points, points removed); geometry polylines (double point removed)
eveBIM	with *Savigliano.ifc*: Some geometries cannot be generated. Error message is: "Warn—IFCProduct (#565376): error IfcRepresentationItem (#564617) Error—this representation Item IfcAdvancedBRep (#564922) is not implemented" (this will be a further development of the software)
RFEM	Many elements were not imported because said to be "not relevant for structural analysis"
Tekla structure	Conversion necessary to Tekla's own structure first. It could not convert quite many objects
Allplan	It ignored 14 objects for *Myran.ifc* and 440 for *Savigliano,ifc*, but does not state which ones
AutoCAD architecture	Despite there not being any errors during the import, not all elements shown on the data page are present
ACCA edificius	It reported the following kinds of errors: incorrect element referencing, unexpected error while calculation entity 2409 cross beam, a critical error has occurred in the structural model calculation
Revit	IFC opening elements not imported and needed to be discarded
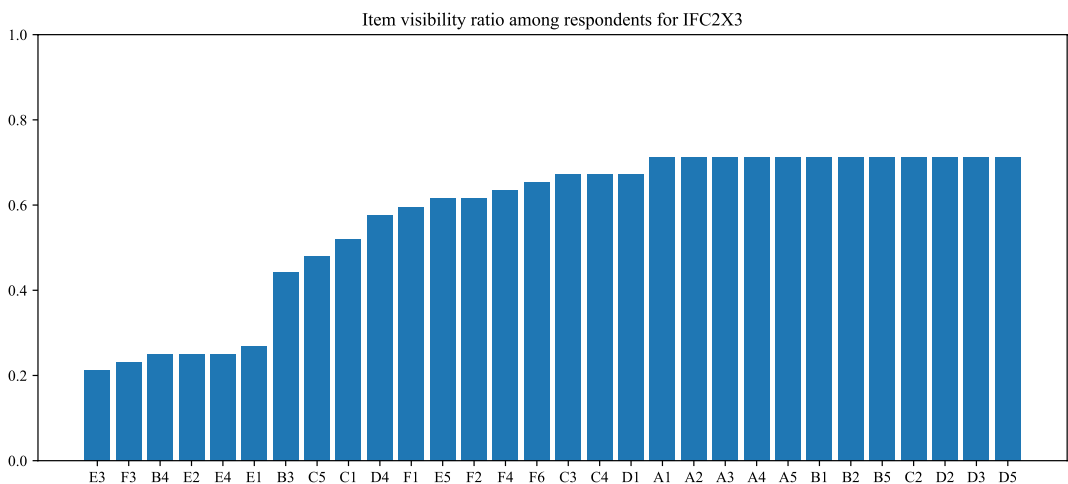
FIGURE 4 Item visibility ratio for IFC2 × 3. The labels on the horizontal axes refer to the column and row of the geometry in Figure 3

As mentioned in the discussion in earlier paragraphs, one of the main concerns of the authors regarding the state of IFC support in importing tools is the divergence in how situations are handled. On the one hand, this can be explained as evolving support for more complicated geometry types or types that are not considered native to the domain of the importing tool. On the other hand, what can be observed in practice is that some tools import

Item visibility ratio among respondents for IFC4
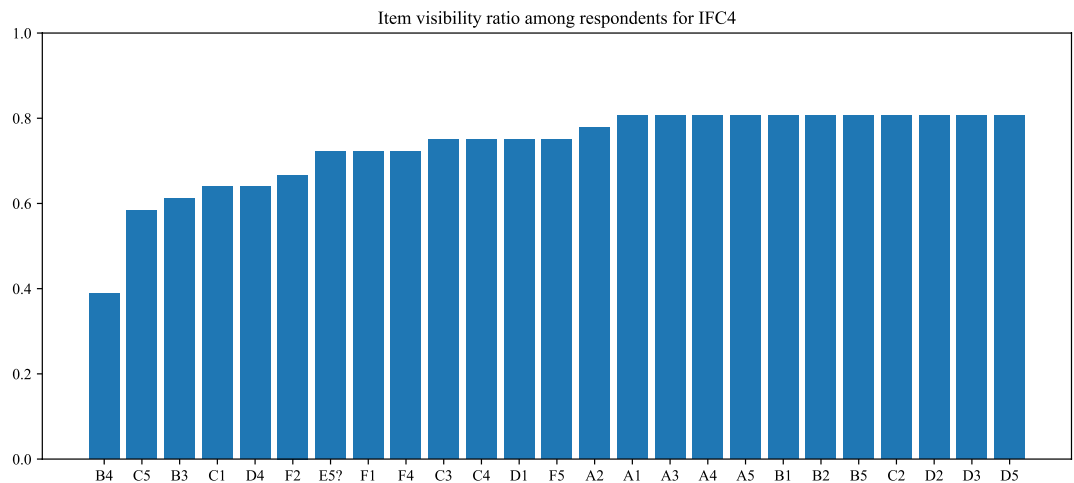
FIGURE 5 Item visibility ratio for IFC4. The labels on the horizontal axes refer to the column and row of the geometry in Figure 3

Consistency among respondents for IFC4
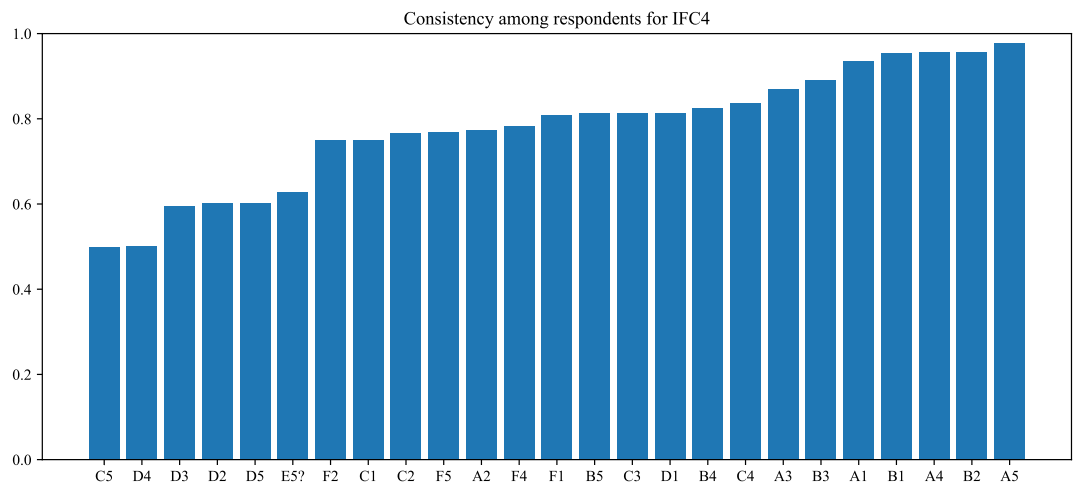
FIGURE 6 Consistency among respondents, defined as the pairwise equality of all combinations of responses (1.0 if all respondents respond the same, 0.0 if all different, although that is generally due to the nature of the questions). Consistency is averaged over the three follow-up questions (regarding position, surface shading and shape type) and excludes responses that indicated the shape is not displayed

invalid situations without notifying the user. In some cases this is due to tools not performing checks on the entity instance attribute values or on the resulting geometries. For example, in the case of the negative extrusion depth, when the IFC parser does not validate WHERE rules automatically, detecting the negative extrusion depth requires an explicit statement in the program code that may have been omitted accidentally. In other cases there is a deliberate effort to "heal" certain invalid situations. For example the *IfcOpenShell* software library provides functionality to detect self-intersections in face boundary definitions and discard all but the largest cycle of edges (https://github.com/IfcOpenShell/IfcOpenShell/blob/df81490/src/ifcgeom/IfcGeomFunctions.cpp#L3656). While this is implemented with good intentions, this behavior does deviate from strict standard compliance and results in different flavors or dialects of IFC being supported among applications, which in the long run will hurt interoperability. The divergence of importing applications is displayed in Figure 6.

The least consistency among the IFC2 × 3 results is seen in D4, C1, C5 (various profiles with invalid extrusion direction). One would expect consistent results here, given that this is a violation of a schema constraint, but this shows the state of implementation in the industry that support for these constraints in parser libraries cannot be assumed, especially not the complicated ones as in this case (the `IfcDotProduct` function). D3, D5 and D2 are also inconsistent, which stems from the question on whether fillet radii are incorporated and whether the surfaces are shaded smooth or faceted. It is interesting to highlight that there is quite some consistency in F4 and F5 as virtually no application took the *StartParam* and *EndParam* of *IfcSweptDiskSolid* into account.

4.3 | Using IFC data

Building Information Modelling tools are usually very complex and making a very detailed report about them is difficult. However, the general functionalities regarding visualization, editing, query, analysis and export possibilities are tested (Table 5). All the software tools offer visualization functions, mainly only 3D, but usually also 2D within the BIM and 3D modeling software groups. Those same two groups are the ones mainly allowing editing of geometry or attributes. Usually, query functionalities are partial within most software.[8]

Looking at analysis, very few software packages allow it: only 20% in the case of analysis concerning the model itself (Type 1), and 30% in the case of designed building performance (Type 2), which is very little for something which is supposed to be an operational tool. Since many different kinds of analysis can fall within the two categories (Types 1 and 2), and the tools tested offered various features, only the general support for the two types is reported in Table 5. In some cases, the support is partial; for example, some Type 1 analyses are sometimes foreseen but not working, as in the case of *Solibri Anywhere*, or only in specific cases, as for *Vectorworks*, working only with NURBS. With Type 2 analyses the pattern is similar, with some tools giving rise to issues; for example, in *Allplan*, *Revit* and *ArchiCAD*, the Type 2 analyses sometimes do not work with the data sets, or with IFC files in general, although available. It is quite difficult to give an exhaustive overview of this, however.

Of the software tools tested, 80% can export to IFC and only 30% allow some customization about the IFC version and model view definition to be used. Additional notes and warnings were reported to the export operations, regarding the possibility of customizing the mapping of entities, in some cases, and a few errors about the way geometry and other features could result from the export.[9]

No sharp connection between the IFC certification and the software support and functionalities is verifiable.

4.4 | Writing of IFC2 × 3 files: Analysis of the exported *Myran.ifc* models

The models which were exported from the tested software (Noardo et al., 2020a, https://doi.org/10.5281/zenodo.3964368) were analyzed using the *NIST IFC File Analyzer*, in order to formally check them. The tool counts the number of entities, relationships and properties and summarizes it in reports. From comparing the summaries, counting the features in the exported files, with the ones regarding the original data sets, we obtain general information about the way in which the software modified the models and we can point out which are the entities that tools have limited support for.

Some of the *Myran.ifc* models delivered were re-exported by the tools without errors, but apparently the export failed, since the building is not represented anymore: this is the case for one of the *ArchiCAD* tests, where only the signboard of the building is stored; the test with *Tekla structure*, where only one beam is there; and one of the models exported by *Revit*, which is empty. In addition, the models exported by *FZKViewer* could not be read by *BIM Vision* or *NIST IFC File Analyzer* to be analyzed and inspected, and *RDF IfcViewer* only shows the *IfcSite* geometry, as terrain, and another two random (wrong) objects in one of the *FZKViewer*-exported cases, and a completely incorrect building in the other one (Figure 7).
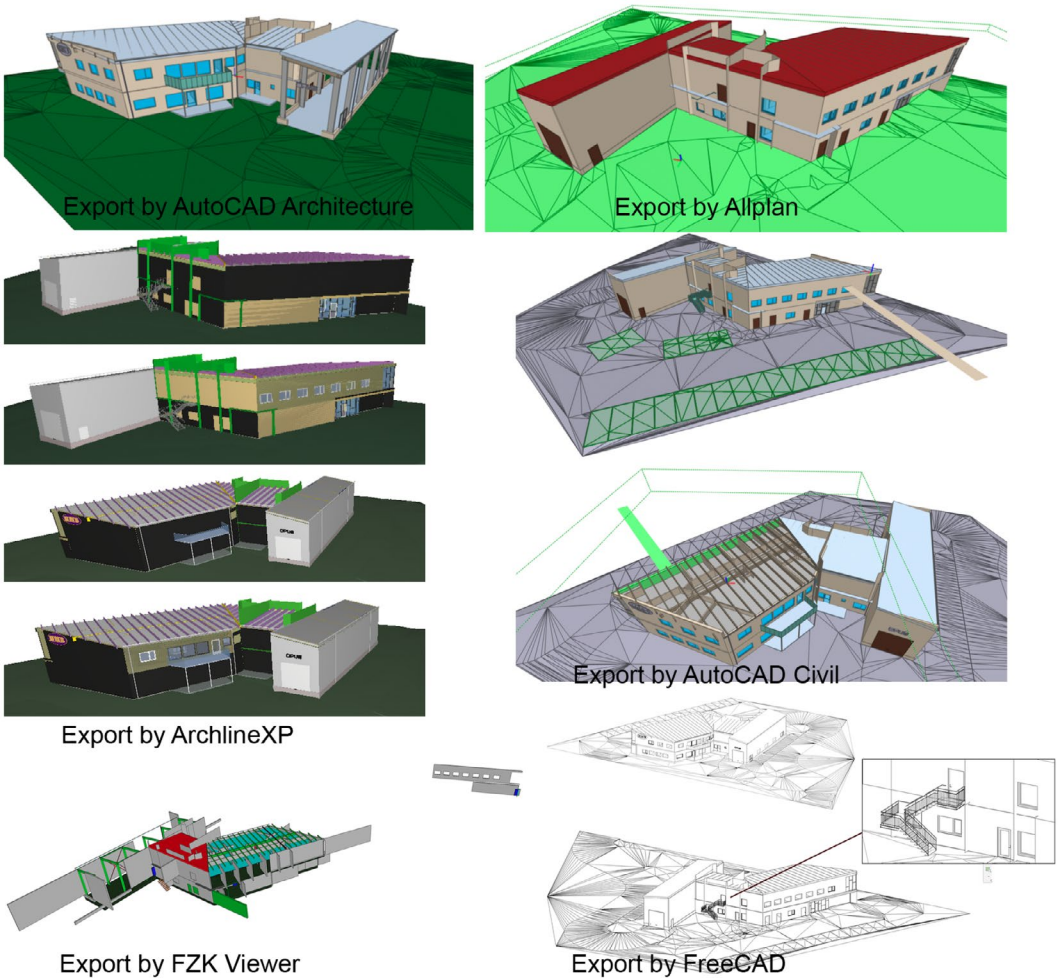
FIGURE 7 Exported Myran models

In the models shown in Figure 7, issues in the exported models mainly regard a change in the model properties, loss of elements, change in the grouping of entities and some occasional apparent change in geometry.[10]

The two *Revit* ones, the two *ArchiCAD* ones, the *ACCA PriMus* one, the *ACCA usBIM.viewer* one (only showing something weird, similar to a duplication in the wireframe representing the site), and those in *eveBIM*, *Simplebim*, *FME* and *Bricscad* look similar to the original data set, and one could assume they are consistent. However, a closer look into the kind and number of entities included in the models, by means of the *NIST IFC File Analyzer* analysis (Table 7), shows they are actually not.

The only two tools leaving the model completely unchanged (same number of each feature) are *ACCA Primus*, which is software supporting calculation of materials for costs assessment, and *ACCA usBIM.viewer*. The only IFC entities which are left unchanged across all the models are *IfcUnitAssignment* and *IfcBuilding* (which is 1 for all the models).

The section regarding some general information, from *IfcApplication* to *IfcSite* (blue-green), is among the most consistent ones. It includes subcategories of *IfcSpatialStructures* information, together with some metadata-level information. Apparently, it is quite easy for the software to keep track of such entities, interpreting and writing them back correctly: the number of *IfcBuildings* remains the same (one). Moreover, for example, *eveBIM* and

TABLE 7 Differences in the NIST analysis results between the Myran models exported by the tools and the one provided

Simplebim add one *IfcApplication* to the file, probably intended to document that the file passed also through such tools. Therefore it is not a mistake in the writing of the file, but a wise update of the file metadata. The same applies for *IfcOwnerHistory*, *IfcOrganization*, *IfcPerson*, *IfcPersonAndOrganization*. However, the syntax used to add

the information in these two files is not the same. Although both are not wrong, this flexibility in the writing of the new file would certainly be a difficult condition for the interpretation of the file by third applications.

In the section representing entities concerning quantities (dark yellow), which are not present in the original file, almost all the models remain the same, except for two of the *ArchiCAD* tests (which report different numbers, however). Also the units representation (orange) is kept quite consistent, and the inconsistencies could be due to the different software settings for export (both for the original file and for the ones tested), not necessarily meaning inconsistency in the model. *IfcDerviedUnit* and *IfcDerivedUnitElement* entities are the ones least kept by the re-exported models, which lose them completely in 7 out of 16 cases.

The most interesting part of this analysis, however, concerns the IFC entities contained in the 'shared building elements' and 'product extension' parts of the IFC structure (white).

Models exported by the same software by different participants (e.g., the two coming from *Revit*, the two coming from *ArchiCAD*) present very similar results in the number of entities, although not exactly the same. The same happens with *AutoCAD Architecture* and *AutoCAD Civil*, which we can suppose use similar algorithms.

Moreover, it is not possible to find any balance in the lost/gained entities considering either the groups of entities being subclasses and superclasses, or that can be considered as alternative representations of the same object (e.g., *IfcWall*, *IfcWallStandardCase*, *IfcWallType* or *IfcStair*, *IfcStairFlight*, *IfcStairFlightType*), or the specific entities and the *IfcBuildingElementProxy* and *IfcBuildingElementProxyType*.

It is also curious how some additional elements appear even if similar ones are kept consistently. For example, in the case of 'stairs' entities, in both *ArchiCAD* tests and in *Bricscad*, although the *IfcStairFlight* entities are still of the same number, another four *IfcStairFlightType* objects (the same number of occurrences of *IfcStairFlights*) are added (only three in one such case, for some reason). In the models exported by *Revit*, 19 more *IfcStair* entities are added.

Similar reasoning is possible when considering *IfcWall*, *IfcWallStandardCase* and *IfcWallType*, although in this case in (only) one of the *ArchiCAD* models and in the *Allplan* case the balance is kept, at least, with *IfcWallStandardCases* becoming *IfcWalls* in the *ArchiCAD* case and *IfcWallType* becoming *IfcWalls* in *Allplan*. Such change in the choice of an entity whose meaning is similar, although losing the optimization advantages for which different entities were proposed in the data model, would keep the model semantically consistent. However, it is not possible to verify a consistent behavior with respect to such choice in software.

The models exported by *FreeCAD* lose parts of objects (as we could see from visual inspection, for example, part of the external stairs is no longer represented), as well as *Allplan* (e.g., again the external stairs). Moreover, *Revit*-exported files, *FreeCAD*, *Allplan*, *AutoCAD Architecture* and *AutoCAD Civil* lose all the *IfcMemberType* entities.

But looking at the final balance in the number of entities, we can see that their number rises. Therefore, assuming that no objects or parts of the model were not exported at all, we can just report on a major number of entities describing the building elements, which means that probably some of the objects were split into several. The increase in the number of entities is also confirmed by the number of geometric objects, whose numbers vary substantially, losing many instances of several geometric entities, while some of them are more than doubled (especially the simplest ones, such as *IfcCartesianPoint*, *IfcFace* and *IfcPolyLoop*). Again, a pattern is difficult to find. Similarly, for the parts concerning the other groups of entities, it is difficult to find similarities across the software or the same entities.

The increase in the number of entities is also reflected in the file sizes, which change from the original 27 MB to a maximum 94 MB for both the *Revit* models.

In addition, the size decreases to 24 MB in the models exported by *eveBIM*, *SimpleBIM*, *Bricscad*. This could mean that either those are better optimized than the original data set or that something was lost in the conversions. In both cases it does not represent the best situation for interoperability, even if it maybe not too bad *per se*.

4.5 | Writing of IFC4 files: Analysis of the exported *Savigliano.ifc* models

There are similar considerations for the exported *Savigliano.ifc* models as for the *Myran.ifc* model.

The models exported by *BIMserver*, *eveBIM* and *SimpleBIM* look good. Again, the models exported by *FZKViewer* cannot be opened in *BIM Vision*, nor it is shown in the *RDF IfcViewer* or analyzed by *NIST IFC File Analyzer*.

In the other models, the loss of elements and properties was the main issue, besides a few changes in some geometries[11] (Figure 8).

When looking at the *NIST* analysis, it is again possible to notice several more inconsistencies, except for the model exported by *BIMServer*, which leaves the model exactly unchanged. *Simplebim* comes second, considering that all the others perform very poorly. In this case there are no differences in the number of building elements, and very little difference among the geometric entities, where many *IfcDirections*, 25% of *IfcCartesianPoints* and some other ones are lost. In this case the main differences are found in properties, materials and relationships.

As in the case of *Myran.ifc*, it is very difficult to find patterns in the loss/gain of entities. In the model exported by *Bricscad* it is possible to see 718 *IfcWalls* lost and 718 *IfcWallStandardCases* gained, probably resulting from the conversion of these ones. The numbers do not correspond perfectly, but similar behavior can be seen also in
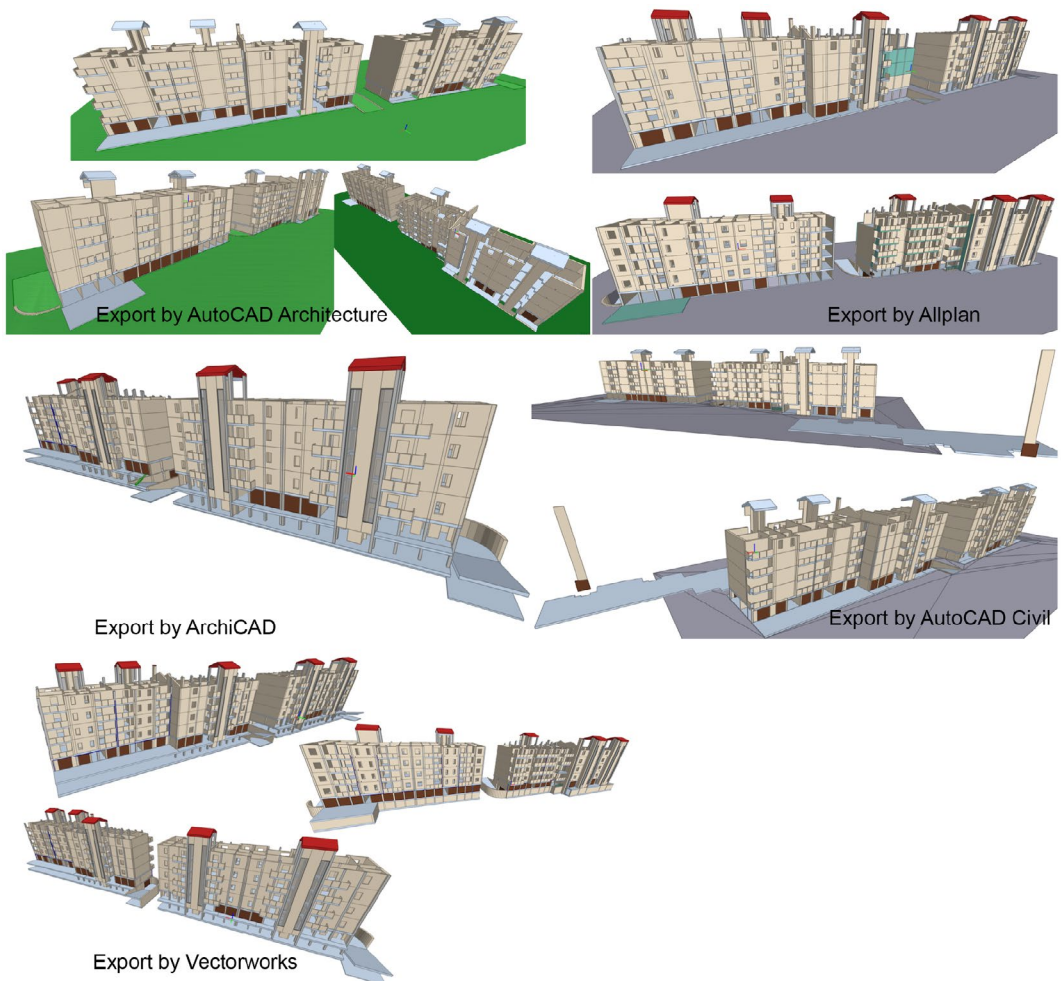
FIGURE 8 The *Savigliano.ifc* models exported

ArchiCAD and *Allplan*, while *AutoCAD Architecture* loses 35 *IfcWalls* and gains 23 *IfcWallTypes*. The most general entity being *IfcWall*, this is a weird result, since the opposite direction of the conversion would be expected to be the safest one for the sake of consistency.

Very few other patterns can be found. In this case many entities are lost, contrary to the *Myran.ifc* case, perhaps due to the scant support for some IFC4 new entities. We could also see from visual inspection that many objects were lost and this is even clearer from counting the entities, where we can see that many are lost completely. The tendencies are generally not straightforward, even if it is possible to notice that the added entities are mainly *IfcPropertySingleValue*, *IfcFace* and *IfcPolyLoop*, but the behavior varies from case to case, and it is necessary to look at Table 8 to check the details.

4.6 | Writing of IFC geometries: Analysis of the *IFCgeometries.ifc* and *IFC4geometries.ifc* models exported

The differences in the NIST analysis results between the *IFCgeometries.ifc* models exported by the tools and the one provided are shown in Table 9 and those for *IFC4geometries.ifc* in Table 10. The information in the two tables indicates that there is no relevant difference between the test with the data set in IFC2 × 3 and the one in IFC4. In the case of these data sets, the original files consisted only of geometries, with only minimal semantic information. Therefore, when it is present in the exported files (e.g., materials, styles and property sets), it means that the software which exported them had attached such information based on its default settings.

Few tools are able to re-export the data leaving them completely consistent with the imported data, namely: *ACCA Primus*, *ACCA usBIM.viewer* and *ArchiCAD*. *eveBIM* can also be included in the list of the good ones, since some elements are added regarding *IfcPropertySets* and metadata, but no elements are lost. *Simplebim* follows, with only few entities missing.

Many geometries resulting from parametric modeling are completely lost, as it is possible to see in correspondence of the lines in light green in Tables 9 and 10 (e.g., *IfcRevolvedAreaSolid*, *IfcCircleProfileDef*, *IfcIShapeProfileDef*, *IfcSweptDiskSolid*) as well as some of the entities which are in the other green rows in the tables, such as *IfcBooleanResults*. Instead, it is possible to observe that the number of other more generic entities increases a lot (e.g., *IfcCartesianPoint*, *IfcFace*, *IfcFaceOuterBound*, *IfcPolyLoop*). The parametric geometries apparently tend to be simplified when exported, by most of the software, in favor of a more explicit representation. This is also evident in the analysis of the other exported models, but in these data sets it appears very clearly.

4.7 | Software performance with IFC

A total of 43 different reports were returned, for 33 different software packages. In particular, multiple results were submitted for *FreeCAD* (2 sets), *FZKViewer* (2 sets), *Autodesk Revit* (6 sets, multiple versions), *Vectorworks Designer* (2 sets), *Archicad* (3 sets). These offer the opportunity for timing comparisons to investigate the impact of hardware on software performance. The graph in Figure 9 gives a summary of the success rates returned for the tests on the three data sets. The graph in Figure 10 gives the count of the different timing values for the successful tests.

Note that in some cases users reported results for some of the tests but not for all of them ("No result reported"). Additionally, some users typed in comments such as "no error" instead of giving specific timing. These are included in the "No result reported" count.

For the *Myran.ifc* data set, none of the tasks took more than 5 min to execute, and the large majority of the software packages tested could zoom and pan the data immediately. Data loading times of less than a minute were

TABLE 8 Differences in the NIST analysis results between the *Savigliano.ifc* models exported by the tools and the one provided

TABLE 9 Differences in the NIST analysis results between the *IFCgeometries.ifc* models exported by the tools and the one provided

also dominant (65% in total). Of the 43 participants, 28 reported successful IFC export, with 86% of these exports being executed in less than a minute.

The impact of the more complex data set (adopting IFC4) on the tests is visible in the results, with only 58% of the software packages able to carry out the basic timed test that involved opening the data set in the software, and 35% reporting that the software crashed. For the *Savigliano.ifc* data set, in contrast to the *Myran.ifc* data set, nine of the successful attempts to open the data (36%) took 5 or more minutes to execute, although once opened zooming and panning the data was immediate, and the large majority of the software packages tested could zoom and pan the data immediately. Data loading times of less than a minute were also dominant (65% in total). Sixteen of the 43 participants reported successful IFC export, with all of these taking less than 5 min to execute.

The results with the *UpTown.ifc* data demonstrate clearly the impact of the larger data set on the tests carried out, with only 47% of the software packages able to carry out the basic timed test that involved opening the data set in the software, and 41% reporting that the software crashed. In addition, in contrast to the *Myran.ifc* data set, nine of the 20 successful attempts to open the data (55%) took 5 or more minutes to execute, with two of the participants reporting test times of between 20 min and 1 hr, and three of over 1 hr. Interestingly, once open, the vast majority of the testers reported sub-minute execution times for zooming and panning. Only 15 of the 43 participants reported successful IFC export, with more than 50% of them taking over 20 min.
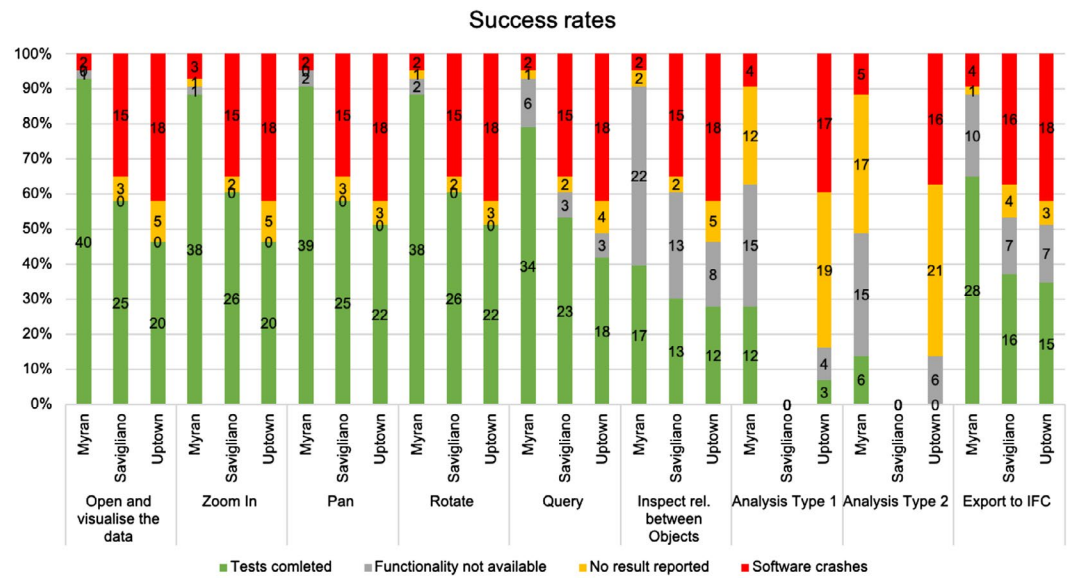
4.7.1 | Multiple tests on same software packages

As noted above, the crowdsourced approach taken in this project resulted in multiple participants testing the same software, providing the opportunity for comparison. For both *FreeCAD* and *FZKViewer*, one of the two respondents reported being unable to open the *Myran.ifc* data set. *Vectorworks* results are similarly consistent, with

TABLE 10 Differences in the NIST analysis results between the *IFC4geometries.ifc* models exported by the tools and the one provided

Entity	Differences absolute											Difference percentage									
	'Ext' 3Dviewer	3Dmodelling sw	Analysis sw	BIM								'Ext' 3Dviewer	3Dmodelling sw	Analysis sw	BIM						
	F2X Viewer	BricsCAD	ACCA PriMus	ACCA usBIM.viewer	Allplan	AutoCAD Architecture	AutoCAD Civil 3D	Simplebim	ArchiCAD	Revit		F2X Viewer	BricsCAD	ACCA PriMus	ACCA usBIM.viewer	Allplan	AutoCAD Architecture	AutoCAD Civil 3D	Simplebim	ArchiCAD	Revit
IfcBuildingElementProxy	3825	9155	0	0	9674	12548	10548	-40	0	-470		686,7145	1643,627	0	0	9133,571	2252,783	1893,716	-7,18133	0	-84,3806
IfcBuildingElementProxyType	0	-4	0	0	-4	-6	-6	0	0	-50		0	-8	0	0	-8	-12	-12	0	0	-100
IfcArbitraryClosedProfileDef	15	3	0	0	0	50	50	0	0	0											
IfcArbitraryProfileDefWithVoids	25	0	0	0	0	0	0	0	0	0											
IfcCircleProfileDef	0	2	0	0	0	0	0	0	0	0											
IfcEllipseProfileDef	-2	1	0	0	-2	-2	-2	0	0	-2		-100	50	0	0	-100	-100	-100	0	0	-100
IfcExtrudedAreaSolid	-2	-26	0	0	-42	-42	-42	0	0	-42		-4,7619	-61,9048	0	0	-100	-100	-100	0	0	-100
IfcIShapeProfileDef	-2	-2	0	0	-2	-2	-2	0	0	-2		-100	-100	0	0	-100	-100	-100	0	0	-100
IfcRectangleHollowProfileDef	-25	-25	0	0	-25	-25	-25	0	0	-25		-100	-100	0	0	-100	-100	-100	0	0	-100
IfcRectangleProfileDef	-5	2	0	0	-5	-5	-5	0	0	-5		-100	40	0	0	-100	-100	-100	0	0	-100
IfcRevolvedAreaSolid	-3	-3	0	0	-3	-3	-3	0	0	-3		-100	-100	0	0	-100	-100	-100	0	0	-100
IfcSweptDiskSolid	-2	-2	0	0	-2	-2	-2	0	0	-2		-100	-100	0	0	-100	-100	-100	0	0	-100
IfcMaterial	0	0	0	0	46	0	0	0	0	0											
IfcRelAssociatesMaterial	0	0	0	0	46	0	0	0	0	0											
IfcPropertySet	0	3	0	0	46	1	1	0	0	1											
IfcPropertySingleValue	0	18	0	0	184	2	2	0	0	2											
IfcRelDefinesByProperties	0	3	0	0	46	1	1	0	0	1											
IfcGeometricRepresentationContext	0	0	0	0	0	0	0	1	0	0		0	0	0	0	0	0	0	100	0	0
IfcGeometricRepresentationSubContext	0	1	0	0	1	1	1	6	0	4											
IfcMappedItem	0	0	0	0	0	44	44	0	0	0											
IfcMaterialDefinitionRepresentation	0	0	0	0	46	0	0	0	0	0											
IfcProductDefinitionShape	0	-4	0	0	-4	-8	-6	0	0	-50		0	-8	0	0	-8	-12	-12	0	0	-100
IfcRepresentationMap	0	0	0	0	0	44	44	0	0	0											
IfcShapeRepresentation	0	-4	0	0	-4	38	38	0	0	-50		0	-8	0	0	-8	76	76	0	0	-100
IfcStyledRepresentation	0	0	0	0	46	0	0	0	0	0											
IfcRelAggregates	0	0	0	0	0	0	0	0	0	-1		0	0	0	0	0	0	0	0	0	-33,3333
IfcRelContainedInSpatialStructure	0	0	0	0	0	0	0	0	0	-1		0	0	0	0	0	0	0	0	0	-100
IfcRelDefinesByType	0	0	0	0	0	44	44	0	0	0											
IfcColourRgb	45	1	0	0	1	45	45	0	0	0											
	0	0	0	0	92	1	1	0	0	0											
	0	0	0	0	1	0	0	0	0	0											
IfcPresentationLayerAssignment	0	0	0	0	1	0	0	0	0	0											
IfcPresentationStyleAssignment	0	1	0	0	0	1	1	0	0	0											
IfcStyledItem	45	0	0	0	44	44	0	0	0	0											
IfcSurfaceStyle	45	1	0	0	1	44	44	0	0	0											
IfcSurfaceStyleRendering	0	0	0	0	1	0	0	0	0	0											
	45	1	0	0	0	44	44	0	0	0											
IfcApplication	0	0	0	0	0	0	0	1	0	0		0	0	0	0	0	0	0	100	0	0
IfcBuilding	0	0	0	0	0	0	0	0	0	0		0	0	0	0	0	0	0	0	0	0
IfcBuildingStorey	0	0	0	0	0	0	0	0	0	-1		0	0	0	0	0	0	0	0	0	-100
IfcOrganization	0	0	0	0	0	0	0	2	0	1		0	0	0	0	0	0	0	200	0	100
IfcOwnerHistory	0	0	0	0	0	0	0	1	0	0		0	0	0	0	0	0	0	100	0	0
IfcPerson	0	0	0	0	0	0	0	1	0	0		0	0	0	0	0	0	0	100	0	0
IfcPersonAndOrganization	0	0	0	0	0	0	0	1	0	0		0	0	0	0	0	0	0	100	0	0
IfcPostalAddress	0	0	0	0	0	0	0	0	0	2											
IfcProject	0	0	0	0	0	0	0	0	0	0		0	0	0	0	0	0	0	0	0	0
IfcSite	0	0	0	0	0	0	0	0	0	0											
IfcAdvancedBrep	0	28	0	0	0	0	0	0	0	0											
IfcAdvancedFace	0	485	0	0	0	0	0	0	0	0											
IfcAxis1Placement	-1	-1	0	0	-1	-1	-1	0	0	-1		-100	-100	0	0	-100	-100	-100	0	0	-100
IfcAxis2Placement2D	-1	6	0	0	-1	-1	-1	0	0	-1		-100	600	0	0	-100	-100	-100	0	0	-100
IfcAxis2Placement3D	39	38	0	0	-6	36	36	4	0	-51		69,64286	67,85714	0	0	-10,7143	64,28571	64,28571	7,142857	0	-91,0714
IfcBooleanClippingResult	0	-1	0	0	-1	-1	-1	0	0	-1		-100	-100	0	0	-100	-100	-100	0	0	-100
IfcBooleanResult	-3	-3	0	0	-3	-3	-3	0	0	-3		-100	-100	0	0	-100	-100	-100	0	0	-100
IfcCartesianPoint	2136	1688	0	0	17638	4178	3378	-46	0	-105		1977,778	1562,963	0	0	16516,67	3868,519	3127,778	-42,5926	0	-97,2222
IfcCartesianTransformationOperator3D	0	0	0	0	0	44	44	0	0	0											
IfcCircle	0	9	0	0	0	0	0	0	0	0											
IfcClosedShell	2	28	0	0	44	47	47	0	0	-2		100	1400	0	0	2200	2350	2350	0	0	-100
IfcCompositeCurve	0	2	0	0	0	0	0	0	0	0											
IfcCompositeCurveSegment	0	32	0	0	0	0	0	0	0	0											
IfcCylindricalSurface	0	206	0	0	0	0	0	0	0	0											
IfcDirection	191	-26	0	0	-38	96	96	-33	0	-27		502,6316	-68,4211	0	0	-100	252,6316	252,6316	-86,8421	0	-71,0526
IfcEdgeCurve	0	1296	0	0	0	0	0	0	0	0											
IfcEdgeLoop	0	541	0	0	0	0	0	0	0	0											
IfcEllipse	0	1	0	0	0	0	0	0	0	0											
IfcFace	390	9	0	0	10782	2586	2186	0	0	-12		3250	75	0	0	89850	21550	18216,67	0	0	-100
IfcFaceBound	0	56	0	0	520	0	0	0	0	0											
IfcFaceOuterBound	390	494	0	0	10782	2586	2186	0	0	-12		3250	4116,667	0	0	89850	21550	18216,67	0	0	-100
IfcFacetedBrep	3	1	0	0	45	42	42	0	0	-1		300	100	0	0	4500	4200	4200	0	0	-100
IfcHalfSpaceSolid	0	-1	0	0	-1	-1	-1	0	0	-1		0	-7,54717	0	0	-100	-100	-100	0	0	-100
IfcLocalPlacement	0	-4	0	0	-4	-6	-6	0	0	-51		0	-7,54717	0	0	-7,54717	-11,3208	-11,3208	0	0	-96,2264
IfcOrientedEdge	1	0	0	0	0	0	0	0	0	0											
IfcPlane	0	2592	0	0	0	0	0	0	0	0											
IfcPolyLoop	0	269	0	0	-1	-1	-1	0	0	-1		0	26900	0	0	-100	-100	-100	0	0	-100
IfcPolyline	390	9	0	0	10832	2586	2186	0	0	-12		3250	75	0	0	90266,67	21550	18216,67	0	0	-100
	64	870	0	0	-1	-1	-1	0	0	-1		6400	87000	0	0	-100	-100	-100	0	0	-100
	0	448	0	0	0	0	0	0	0	0											
	0	5	0	0	0	0	0	0	0	0											
IfcShellBasedSurfaceModel	0	-1	0	0	-1	5	5	0	0	-1		0	-100	0	0	-100	500	500	0	0	-100
IfcTrimmedCurve	0	8	0	0	0	0	0	0	0	0											
IfcVertexPoint	0	101	0	0	0	0	0	0	0	0											
IfcConversionBasedUnit	0	0	0	0	0	0	0	0	0	0		0	0	0	0	0	0	0	0	0	0
IfcDerivedUnit	0	0	0	0	0	0	0	4	0	6											
IfcDerivedUnitElement	0	0	0	0	0	0	0	8	0	16											
IfcDimensionalExponents	0	0	0	0	0	0	0	0	0	0		0	0	0	0	0	0	0	0	0	0
IfcMeasureWithUnit	0	0	0	0	0	0	0	0	0	0		0	0	0	0	0	0	0	0	0	0
IfcSIUnit	0	2	0	0	2	2	2	10	0	14		0	50	0	0	50	50	50	250	0	350
IfcUnitAssignment	0	0	0	0	0	0	0	0	0	0		0	0	0	0	0	0	0	0	0	0

the only minor difference being that one participant reported that the *Myran.ifc* data opened immediately and another that it took less than 1 min (both on machines with 16 GB RAM).

Identical timing results are reported for the zoom/pan/rotate/query tasks for the remaining two packages (*Autodesk*, *ArchiCAD*), and as noted above, the large majority of software packages managed these functions in near immediate time. As different analytical tasks were selected by participants, these are not comparable.

Differences do, however, arise when execution time for the remaining functionality is considered. Table 11 summarizes the remaining timing results for these packages for the *Myran.ifc* data set, which was chosen as this proved the least problematic of the three data sets in terms of successful execution of tasks.

For *Revit*, there is an apparent performance improvement when opening *Myran.ifc* data in the 2020 version of the software, and an improvement of export time from 1–5 min in the 2018 version, to "almost immediate"

FIGURE 9 Graph reporting the success rate of the tested tools in the indicated tasks (in the bars, the number of tools for each case)
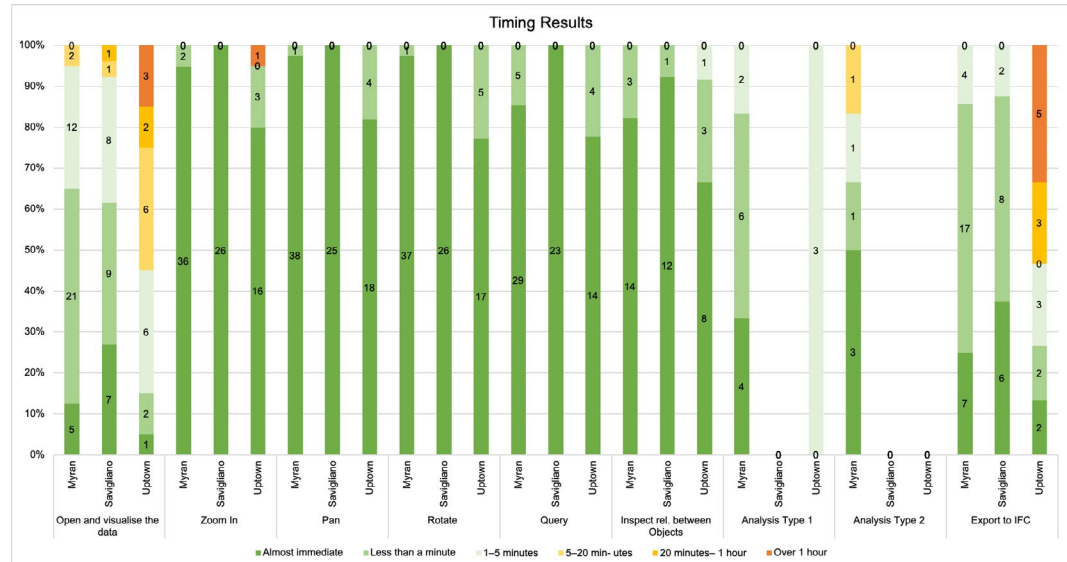
FIGURE 10 Graph reporting the timing results of the tested tools in the indicated tasks (in the bars, the number of tools for each case)

in the 2019 and 2020 versions of the software. In terms of hardware, all tests were carried out on Windows 10 machines. However, the machine used for *Revit 2020* had 64 GB of RAM, with result 1 obtained with 16 GB, and results 2, 4 and 5 with 8 GB. Contradictory results in the linked object query make these difficult to compare. For *ArchiCAD*, all users report an IFC export time for *Myran.ifc* of less than a minute, but two report that it was not possible to query linked information. Examining the hardware used, result 3 for *ArchiCAD* (file opening time of 1–5 min) was obtained on a machine with 8 GB RAM, whereas the others were tested on machines with 16 GB RAM (all Windows 10 machines, all with dedicated graphics cards).

TABLE 11 Inconsistent timing results submitted for the *Myran.ifc* data set

	Import	Inspect Linked Objects	Export to IFC
Autodesk Revit 2019—Result 1	1–5 min	Almost immediate	Almost immediate
Autodesk Revit 2018—Result 2	1–5 min	Almost immediate	1–5 min
Autodesk Revit 2018—Result 3	1–5 min	The software does not allow this	1–5 min
Autodesk Revit 2018—Result 4	1–5 min	The software does not allow this	Crashes
Autodesk Revit 2019.2—Result 5	1–5 min	The software does not allow this	Almost immediate
Autodesk Revit 2020—Result 6	Less than a minute	Less than a minute	Almost immediate
ArchiCAD—Result 1	Less than a minute	The software does not allow this	Less than a minute
ArchiCAD—Result 2	Less than a minute	The software does not allow this	Less than a minute
ArchiCAD—Result 3	1–5 min	Almost immediate	Less than a minute

5 | DISCUSSION

What is apparent from analyzing the results of the exported files is that it is very difficult to find a pattern in or a sensible explanation for what happened in the export phase, since there are very few cases in which the IFC entities are completely kept/lost/gained in all the exported models. The most frequent scheme is: some models keep all, some models lose all, some models keep part, some entities are added. The very high complexity of both the semantic data model and the geometry management is certainly related to the issue.

Similarly, the investigation into loading IFC files suggests a general difficulty in reading semantics correctly, very little support for georeferencing and quite good support for general geometry, as visualized, which is not unambiguously read though, when looking at the IFC geometry data sets results. But no clear pattern is found that could make it possible to better understand the specific issues of IFC interoperability. A clear result is just that little interoperability is actually achieved, since there are very few tools able to read the standardized data sets correctly and even fewer that are able to export them consistently. The ability to uniquely interpret the models and to leave them consistent through the import–export phases is absolutely essential for interoperability and what it enables (data exchange, data reuse, etc.). At this stage it is not possible to fully trust standardized models, even for simple file exchange.

Results were reported for 33 software packages, highlighting the wide range of tools available to users. Different versions of *Revit* also demonstrated that *Autodesk* is making improvements to the interoperability of the software: *Revit 2019* gave results for all data, but *Revit 2018* could not handle two of the three data sets (*Uptown.ifc* and *Savigliano.ifc*). *Myran.ifc* could be opened by 40 participants, *Savigliano.ifc* by 25 and *Uptown.ifc* by 20. The 12 participants who reported that they were not able to open *Uptown.ifc* also reported the same issue with *Savigliano.ifc*. Ten participants reported problems with *Uptown.ifc* but none with *Savigliano.ifc*, and four with *Savigliano.ifc* but not with *Uptown.ifc*, indicating that the size of the data set is, overall, more problematic than the IFC version.

Regarding performance, the impact of the size of the data set on overall performance is marked, with 55% of the software packages that managed to open the *Uptown.ifc* data set reporting a time of 5 min or more. While it is not possible to say which software package is the fastest (the approach to timing used general timing categories rather than requiring the user to undertake the onerous task of time measurement) and performance will also depend on the hardware being used, we can report that none of the software packages managed to carry out all the visualization tasks in under a minute for the *Uptown.ifc* data set, although nine packages achieved this with the *Myran.ifc* data set and nine with the *Savigliano.ifc* data set. For IFC export, 24 software packages managed to export the data in 1 min or less for *Myran.ifc*, 14 for *Savigliano.ifc*, but only two for *Uptown.ifc*.

Moreover, the set of available functionalities of software to work with IFC models is quite limited. The analyses concerning the model itself (Type 1) are essential to make sure a model is valid and suitable to support further

analysis. On the other hand, tools are necessary to use IFC models for the operational intentions it was conceived for. However, not many tools are now able to work with IFC effectively.

The fact that the inquiry is based on voluntary and completely open contributions can be considered both a strength and a limitation of this work. On the one hand, it is essential to cover the investigated object in the most thorough way: as many software packages as possible, with many experts involved, but also with the inclusion of less expert users to test user-friendliness. However, one outcome of this approach is some incompleteness in the tools that are assessed.[12] This one was limited by the *ex post* integration of tests for software not originally considered. Moreover, the tests reporting suspicious results according to the experiences of the promoting team (for both too good or too bad performances) were double-checked with new tests or by asking for clarifications. A further issue could be the inexperience of some testers, reporting inaccurately on tools' behavior. In mitigation, it was checked that all responses were given with sufficient care, whatever the level of expertise. Once verified, any conflicting responses with respect to the tools' actual potential could indicate a deficiency in the suitability of the tool for an inexpert user. The involvement of a large part of the community (twice the number of participants delivering the results initially registered to the initiative) is also important to perceive the relevance of the topic, although dealing with an issue considered not as a priority among the high-level standardization and academic communities.

5.1 | Lessons learned

The insight and data gained from this study can form the basis for reasoning about improvements to the IFC model, the common behavior of the models and the most urgent issues to tackle with software, for example, the suitable implementation of georeferencing.

Control in export and import would be critical to make the choice explicit about possible alternatives for storing information in IFC. Although possible ambiguities and equivalent alternatives should be avoided as much as possible, in some cases applications could require information to be stored according to specific structures, considering the slight differences between them (as an example, see the difference between LoGeoRef30 and LoGeoRef40, or the generalization level of semantics required by applications).

Such control for georeferencing would consist in the choice about the LoGeoRef to be read, when alternative ones are stored (including LoGeoRef50 enabled by IFC4). It could be both a choice, to be made during import, and an instruction, as a requirement for the model to be input. An argument could be made about the opportunity of allowing the reading of planar coordinates North and East, heights and rotation separately, according to different LoGeoRefs.

The georeferencing issue is even more articulated. First, regardless of the country and its level of advancement in implementing the BIM concept in ongoing projects, the little use by designers and architects of BIMs with the right georeference is similar. Almost all designers expect a model in their local coordinate system, despite the fact that the point cloud obtained for this purpose has the correct coordinates, consistent with the national system. To make matters worse, in most cases, "long" coordinates are truncated due to software problems with them. Therefore, in addition to developing effective methods for storing georeferencing in IFC files, it seems necessary to provide education to designers and architects and clear guidelines on how to deal with it.

With respect to semantics, probably a smaller set of entities, to be used according to stricter rules, would help in implementing a suitable interpretation in software. At least Building Information Modelling software, supposed to have the most general scope, should try as much as possible to implement internal structures of entities/families in the native format in agreement with the IFC data model. This would be not supposed to be a top-down directive from buildingSMART, but probably an agreement coming from the collaboration of software developers with standardization institution, starting from the needs of practitioners and users.

If such a stricter collaboration were achieved, the IFC standard developers could have a more solid indication for selecting the most important core part to be maintained, improved and likely simplified or made less ambiguous.

Control and transparency would be the next requirements. They would represent added value in the case of export. How will the software export the data? For example, a software-specific but explicit MVD could describe exactly how the internal model of the software is mapped on which part of the IFC model. And they would help the import phase as well: how will the IFC classes be interpreted? Which ones are supposed to be interpreted exactly, since part of the internal data model followed by the software? Which ones could instead be misinterpreted and how? Perhaps a choice could be provided to map the unrecognized classes, or they could be represented as their parent class, or as a generic class. This should be a straightforward, guided and clear process instead of an advanced option of the software used mainly by experts.

Since it is not possible to detect any pattern in the behavior of software, the issue is more identifiable in such indefiniteness than in one specific aspect of the standard or of the software.

One unique way to write IFC files should also be imposed, to avoid redundancy or ambiguity in the storage of information. Semantics would have advantages from simpler and clearer rules about how to structure entities, priorities and limits. For implementations, and therefore practice, it is certainly preferable to rely on a simpler model than a very complex and expressive but inconsistent one.

Finally, reducing the set of geometries in the schema, together with clear constraints on how and when to use these geometries, would enable software developers to implement them more robustly, avoiding loss of information, intersections and so on. As is already the aim in future versions of IFC, this should also be made more use-case dependent. When an export is intended to function as a static reference for visualization, coordination or inference checking, procedural and parametric representation should be largely avoided. When transfer of design intent and parametric behavior is required the more procedural geometry definitions can be used. In addition, the non-compliances possibly implemented by some software should at least give an error message and make the user explicitly choose if using a non-compliant IFC file anyway, to the detriment of interoperability. This should be part of the IFC certification process too. One potential avenue to detect flaws in the geometric interpretation is to serialize a set of descriptive properties (such as center of mass, surface area, volume, surface genus) along with the geometry during export. In this way, the importing application can calculate the same parameters from its reinterpreted geometry representation and compare them. These are the so-called geometric validation properties from Pratt, Anderson, and Ranger (2005) in the case of STEP.

Again, implementing some control on the export of geometries by modeling software would help in making the (likely few) possible choices explicit, fostering reusability and interoperability. The storage of such choices in metadata would also be useful.

As a general rule, the use of default settings while importing or exporting files to overcome the presence of undefined information should always be avoided, since it is source of many misleading inconsistencies (e.g., inferring CRS).

In some cases, it could be possible to object that the IFC aim is different from some of the aspects tested here. For example, someone could say that editing is not a priority or not even a planned possibility for IFC files, which should only be exchanged as reference for modelling further parts by different professionals designing the same building. For this task, semantics are barely needed. Another use of the IFC models is analysis (asset and facility management, project management, project analysis, computations and so on), where semantics and quality do matter. If it is the case, the purpose of IFC itself should be clearly stated and perhaps reduced.

6 | CONCLUSIONS

The study described in this article, part of the GeoBIM benchmark project, was intended to point out and provide evidence about the support for and issues of available software for standardized information in IFC. Interoperability is essential for a number of use cases, and even for simply exchanging and reusing data. Standards

are supposed to be enabling such interoperability, and standardization is the essential premise to the development of any integration, including GeoBIM.

The Industry Foundation Classes, by buildingSMART, are the official reference open standard for BIM management and storage. However, we have personally stumbled across a number of issues while using IFC, and many more have been often informally reported to us by practitioners. These prevent the effective use of IFC in practice, without the issues being clear to users, who usually rely on proprietary formats (such as the *Revit* native format), using them as often more reliable *de facto* standards. Moreover, although we were aware that some data or functionality loss was prevalent when using IFC, no systematic proof was available in the literature. This project aimed to create such proof, which could be used as the basis for future improvements in implementations, in data modeling and in the standard itself.

The results show how difficult it is for software tools to read, manage and export the IFC data sets consistently, in a manner that would take full disadvantage of interoperability, and that would enable the effective usability of data sets in use cases too.

Besides the internal interoperability of IFC in BIM software, the potential integration with 3D geoinformation and respective standards is additionally hindered. It is clear that if even IFC software cannot consistently read and write IFC, multi-format software will do even worse, especially when dealing with the areas where both domains diverge (e.g., georeferencing and semantics). In fact, it is almost impossible to plan, design and implement effective solutions for mapping, conversions and object transformation to other formats without relying on a stable and trustworthy initial format.

We are aware that there is a possible bias in this study's results, which could be produced by the lack of expertise of some participants, or by the initial inaccuracies in the data sets provided. However, such a bias would also reflect the way in which users use software and product data in practice. Nevertheless, in order to minimize potential bias, the data sets were validated and improved as much as possible for the purpose of the benchmark, in order to limit their influence on the quality and reliability of results. Therefore, if there are still problems in the data sets, it would likely reflect additional drawbacks of the standard itself, for the lack of clarity about its use for the modeling of actual data sets and the difficulty in implementation, which could produce unintuitive tools.

This study shows the drawbacks of the current IFC standard and implementation, and the related difficulties, also due to the challenges to which it is intended to respond (e.g., representation of the information regarding a vast and complex field, flexibility to multiple needs). A particularly important point is that the outcomes of the study are officially documented here, in a manner that they can be the basis for future research in the field and development of concrete solutions, such as the addition of constraints and specific guidelines, simpler ways to store geometry, better selection of useful semantics, and so on.

Giving clear guidance towards changes in the IFC schema is outside the scope of the present article. However, it is clear that from the point of view of software support, more consistent implementations of the existing schema are needed. There are many possibilities that could help in this regard, such as the creation of high-quality open source libraries to read/write IFC, the creation of reference implementations, and the writing of more extensive documentation that shows how the IFC data model is mapped to software's internal data models. All of these would enable the reuse of source code and the definition of best practices.

Considering the results of this study as evidence, future work should be aimed at filling in gaps in the results (e.g., through the study of additional software or new reference data sets), updating the results with respect to new software, and the solution of the outlined issues.

For example, a clear area of opportunity is the definition of clear guidelines about the use of complex semantics, fixing priorities and criteria for their use. Moreover, specific kinds of geometries should be selected for specific cases, adding constraints to guarantee that software can import, read, use and re-export them without any change. In addition, when considering the performance related to computational requirements, we can easily understand that the reduction of data size is urgent. buildingSMART is working towards solving such issues, striving for less complex models offering more straightforward choices, easier to implement (https://github.com/

buildingSMART/NextGen-IFC/wiki/Ten-principles-for-a-future-IFC), and with a view towards IFC v.5. Thanks to this study, it has been possible to gain a greater awareness of the specific issues to be tackled in future research in order to foster an improved adoption of the IFC standard.

ACKNOWLEDGMENTS

ORCID

Francesca Noardo https://orcid.org/0000-0003-2269-5336
Ken Arroyo Ohori https://orcid.org/0000-0002-9863-0152
Lars Harrie https://orcid.org/0000-0003-3252-1495
Helga Tauscher https://orcid.org/0000-0003-0411-8770
Cristina Leoni https://orcid.org/0000-0003-4906-5692
Artur Warchol https://orcid.org/0000-0002-2248-1357

NOTES

[1] As seen in repositories such as https://github.com/buildingSMART/NextGen-IFC.

[2] This task is the subject of this article.

[3] https://3d.bk.tudelft.nl/projects/geobim-benchmark/software.html for the software tested and https://3d.bk.tudelft.nl/projects/geobim-benchmark/participants.html for the list of participants.

[4] As examples, in *Bentley Microstation+TerraSolid*, for the *Myran.ifc* model, only *IfcWall* and *IfcWallStandardCase* were translated; in *ACCA Edificius* "opening, door, and curtain wall are detected by the software; IFCBeam and IFCCovering are loaded as an IFC Proxy Object; It also has vertical/horizontal envelopes which are actually not IFC classifications"; for *Revit*, "some classes that are different in IFC are the same in Revit, following the standard settings; it is possible to set own Revit Categories (therefore set them to the correct IFC Class name)"; in *ArchiCAD*, "Some classes/layers are missing, most notably are the roofs and windows"; *Tekla structures* does not import all the entities, although the present ones are consistent.

[5] In *FME*, hierarchy and relationships are kept by means of parent IDs; for *Revit*, the loss of many relationships is reported, probably due to the need to separate elements while importing; likewise for *BIM Vision*; in *FreeCAD*, "spatial aggregation and containment is properly retained (although in debug mode, aggregations with more than 10 objects are skipped), Type objects (IfcRelDefinedByType) seem not to be considered, IfcRelConnectsPathElements (connection between walls) is ignored"; In *Autodesk civil 3D*, only storey aggregation is kept; in *ACCA PriMus-IFC*, it is only possible to find many "IfcRelConnectsPathElements" relations.

[6] In the *FZKViewer* sometimes attributes are different from the reference ones, although generally well managed and interpreted; *Bentley Microstation+TerraSolid* can read consistent attributes for the *Uptown.ifc* data set but not for *Myran.ifc*; *DDS-CAD* is supposed to read attributes correctly; however, the function did not work, therefore it was not possible to check; in *Revit*, some of the attributes were consistent with the IFC ones, but most of them were missing; in *ArchiCAD*, some objects are not recognized correctly in terms of attributes and semantics.

[7] Specific reported features include the following. *FZKViewer* generally works well, except with the *Myran.ifc* model, whose geometrical representation looks a bit distorted in one of the tests, while the geometry is not visualized at all in the other one; moreover, with the *Savigliano.ifc* (IFC4) file, some elements (roof pitches) are not visualized. *eveBIM* detected incorrect normals, although we do not know if they were incorrect at the beginning or were changed during import, and with respect to IFC4 files, *IfcAdvancedBrep* is not implemented so the geometry is not shown in this case. For *Bentley MicroStation +TerraSolid*, some slight changes in dimensions are reported. *BIMServer* seems to read geometry

correctly, except that the roof at the top of the tower is missing, and possibly windows. For **Revit**, some issues are reported: some walls intersecting with the floors and beams that do not join correctly, subtracting the volumes; mistaken display of subtraction solids inside the families of doors and windows, which are used to pierce the walls; often the stratigraphy of walls is not correct in the corners and intersections. In **Vectorwork** a scale can be applied to the data, thus, it is necessary to check that it is correctly set.

8   They are foreseen within GIS software. 3D viewers usually have partial query abilities; for example **BIM Vision** provides general pre-defined ways to view the information, listed by structure, by types, by groups (system, zones or other) and by layers. For each, properties, location, classification and relations can be shown at the object level. Also **FZKViewer** offers mainly pre-defined queries. **Lexocad** can only query based on object type (e.g., wall, slab, wall standard, …). In **Autodesk civil 3D**, as well as in **Allplan** and **AutoCAD Architecture**, it is only possible to select properties which are in the native software format. In **Vectorworks**, query tools are present, but query is possible if the project is connected to a database; object selection and information retrieval is possible Finally, **ArchiCAD** presents a "find-and-select" tool, but issues are reported.

9   In **FME**, there were a number of warnings related to inappropriate geometry types. In **FreeCAD**, the default settings yielded an error regarding colors (to be checked in later versions and reported for fixing if necessary). By changing the settings to use the IfcOpenShell serializer, the error did not appear. For **ACCA PriMus**, it is possible to add additional properties. In **Revit** it is possible to customize the export by means of manual entity mapping. Similarly, in **Vectorworks**, a layer mapping pre-process was needed to export the file. Each layer had to be selected separately and assigned to the correct "vector story name."

10 The model exported by **FreeCAD** gives some problems when trying to load it into viewers: **RDF IfcViewer** can show it only very far and small (probably some change in the coordinate reference system happened), and it can be hardly handled in such software; **BIM Vision** is not able to visualize it; **Revit** gives a warning ("IFC:only 2 points in polyloop #702627, expected ≥3"), but then shows it quite correctly. There, it is possible to see that some elements are missing (e.g., the beams in the exterior stairs). In **Allplan**, the difference that could be noticed with respect to the original is the lack of the external stair and, in the grouping, the **IfcSite** is stored separately from the three other storeys, which is actually a better storage, but different from the original benchmark-provided file. In **AutoCAD Architecture,** the three storeys are collapsed into a single one, big parts of objects belonging to the side building are missing (e.g., walls, doors), the roofs' color changed for some reason, probably due to the loss of some properties, and some of the walls that were cut at the roof level are not, so that they protrude above the roof. In **AutoCAD Civil**, the three storeys are collapsed into only one, the terrain model constituting the **IfcSite** is split in its different components (parking places, street, different parts of the terrain), which are recognizable by a human eye, but were not labeled differently in the original data set. All of these are represented by means of **IfcBuildingElementProxy** entities. Also in this case, the roofs and the walls are changed in the same way as in the **AutoCAD Architecture** model. Moreover, one of the walls has a part protruding towards the exterior in a clear error. In the model exported by **Archline XP**, the windows and doors seem to disappear, even if they rest below the most exterior layers; probably the reason was some issues in the use of **IfcOpenings** to subtract the volumes corresponding to doors and windows in all the walls and coverings concerned. The roof slab is also missing.

11 In the model exported by **Allplan**, the upper part representing the roofs is missing, and all the windows as well. Moreover, some of the external walls and some slabs are colored in a different way (blue), but it is not very clear what is changed. The model exported by **AutoCAD Architecture** looks very similar to this last one: the top part of the building is almost completely missing, even more than in the **Allplan** model. Moreover, the windows are missing and, in this case, the openings in the walls, where the windows and doors would be supposed to be inserted, are missing too, presenting plain filled walls. Even the ramp leading to the underground parking is filled and appears as the continuation of the site. Moreover, other elements are missing, such as the windows in the stair towers, closing elements. In addition, the color of some elements, especially slabs, is missing, so that probably some property was lost. The **AutoCAD Civil** one is very similar to the **AutoCAD Architecture** one, but in addition, a slab and a door are completely displaced and the site color changes. However, the ramp to access the underground parking is well represented. One possible explanation is that, since the software is intended to represent civil works, those ramps can be better interpreted with respect to the same in **AutoCAD Architecture**, although probably the two tools are based on similar algorithms. The model exported by **ArchiCAD** is also missing the top part and windows. Moreover, the site around the building is missing. An apparent error is a sort of beam (albeit classified as wall) that goes from one of the ground slabs into the ramp going to the parking (in green in Figure 8). Another model, exported by **Vectorworks**, is very similar, although without the beam invading the ramp.

12 For example, the open source **xBIM** (https://docs.xbim.net/index.html), the very recent **BlenderBIM** (https://blenderbim.org/) and some niche engineering tools were not tested.

REFERENCES

Aksamija, A. (2018). *Methods for integrating parametric design with building performance analysis*. Retrieved from https://www.arcc-repository.org/index.php/repository/article/view/459

Aleksandrov, M., Diakite, A., Yan, J., Li, W., & Zlatanova, S. (2019). Systems architecture for management of BIM, 3D GIS and sensors data. *ISPRS Annals of Photogrammetry, Remote Sensing & Spatial Information Sciences*, 4(4/W9), 3–10.

Arroyo Ohori, K., Diakite, 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.

Azhar, S. (2011). Building information modeling (BIM): Trends, benefits, risks, and challenges for the AEC industry. *Leadership and Management in Engineering*, 11(3), 241–252.

Baik, A., Yaagoubi, R., & Boehm, J. (2015). Integration of Jeddah historical BIM and 3D GIS for documentation and restoration of historical monument. *International Archives of the Photogrammetry, Remote Sensing & Spatial Information Sciences*, 40(5/W7), 29–34.

Barbini, A., Malacarne, G., Massari, G., Monizza, G. P., & Matt, D. T. (2019). BIM objects library for information exchange in public works: The use of proprietary and open formats. *WIT Transactions on the Built Environment*, 192, 269–280.

Bartie, P., Reitsma, F., Kingham, S., & Mills, S. (2010). Advancing visibility modelling algorithms for urban environments. *Computers, Environment & Urban Systems*, 34(6), 518–531.

Biljecki, F., Kumar, K., & Nagel, C. (2018). CityGML Application Domain Extension (ADE): Overview of developments. *Open Geospatial Data, Software & Standards*, 3, 13.

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

Clemen, C., & Gorne, H. (2019). Level of Georeferencing (LoGeoRef) using IFC for BIM. *Journal of Geodesy, Cartography & Cadastre*, 10, 15–20.

Egusquiza, A., Izkara, J. L., & Gandini, A. (2018). Energy efficiency improvement in historic urban environments: From decision support systems to co-creation strategies. In *Proceedings of the Third International Conference on Energy Efficiency in Historic Buildings*, Visby, Sweden (pp. 576–584). Uppsala, Sweden: Uppsala University.

Farid Mohajer, M., & Aksamija, A. (2019). Integration of building energy modeling (BEM) and building information modeling (BIM): Workflows and case study. *Building Technology Educator's Society*, 2019(1), 37.

Fosu, R., Suprabhas, K., Rathore, Z., & Cory, C. (2015). Integration of building information modeling (BIM) and geographic information systems (GIS): A literature review and future needs. In *Proceedings of the 32nd CIB W78 Conference*, Eindhoven, the Netherlands (pp. 27–29).

Haddock, A. (2018). *Building information modelling for asset and facilities management* (Unpublished MS thesis). Wellington, New Zealand: Victoria University of Wellington.

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

ISO. (2014). *Industrial automation systems and integration: Product data representation and exchange*. Geneva, Switzerland: International Organization for Standardization.

Jakubiec, J. A., & Reinhart, C. F. (2013). A method for predicting city-wide electricity gains from photovoltaic panels based on LiDAR and GIS data combined with hourly Daysim simulations. *Solar Energy*, 93, 127–143.

Kamari, A., Laustsen, C., Petersen, S., & Kirkegaard, P. H. (2018). A BIM-based decision support system for the evaluation of holistic renovation scenarios. *Journal of Information Technology in Construction*, 23, 354–380.

Kang, T. W., & Hong, C. H. (2015). A study on software architecture for effective BIM/GIS-based facility management data integration. *Automation in Construction*, 54, 25–38.

Kensek, K. M. (2014). Integration of environmental sensors with BIM: Case studies using Arduino, Dynamo, and the Revit API. *Informes de la Construcción*, 66, e044.

Kumar, K., Labetski, A., Arroyo Ohori, K., Ledoux, H., & Stoter, J. (2019). The LandInfra standard and its role in solving the BIM-GIS quagmire. *Open Geospatial Data, Software & Standards*, 4(1), 1–16.

Kumar, K., Ledoux, H., Commandeur, T. J. F., & Stoter, J. E. (2017). Modelling urban noise in CityGML ADE: Case of the Netherlands. *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences*, 4(4/W5), 73–81.

Lamartina, L., & Sa, J. (2019). Structural pre-analysis through implementing Revit structures and robot to develop and compare engineered timber structural families. *WIT Transactions on the Built Environment*, 192, 229–243.

Liang, J., Gong, J., Li, W., & Ibrahim, A. N. (2014). A visualization-oriented 3D method for efficient computation of urban solar radiation based on 3D–2D surface mapping. *International Journal of Geographical Information Science*, 28(4), 780–798.

Lim, J., Tauscher, H., & Biljecki, F. (2019). Graph transformation rules for IFC-to-CityGML attribute conversion. *ISPRS Annals of Photogrammetry, Remote Sensing & Spatial Information Sciences*, 4(4/W8), 83–90.

Liu, X., Wang, X., Wright, G., Cheng, J. C.-P., 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.

Lv, S. (2018). Study on BIM modeling method of bridge railway integration based on Revit and Civil3D. In *Proceedings of the Seventh International Conference on Energy and Environmental Protection*, Shenzhen, China (pp. 214–220). Amsterdam, the Netherlands: Atlantis Press.

Nguyen, H. T., & Pearce, J. M. (2012). Incorporating shading losses in solar photovoltaic potential assessment at the municipal scale. *Solar Energy*, *86*(5), 1245–1260.

Niu, S., Yang, Y., & Pan, W. (2019). Logistics planning and visualization of modular integrated construction projects based on BIM-GIS integration and vehicle routing algorithm. In *Proceedings of the 2019 Modular and Offsite Construction Summit*, Banff, Canada (pp. 579–586). Edmonton, Canada: University of Alberta.

Noardo, F., Arroyo Ohori, K., Biljecki, F., Ellul, C., Harrie, L., Krijnen, T., & Stoter, J. (2019). GeoBIM benchmark 2019: Design and initial results. *ISPRS International Archives of Photogrammetry, Remote Sensing & Spatial Information Sciences*, *42*(2/W13), 1339–1346.

Noardo, F., Arroyo Ohori, K., Biljecki, F., Ellul, C., Harrie, L., Krijnen, T., & Stoter, J. (2020a). *IFC files exported within the GeoBIM benchmark 2019—Task 1*. Retrieved from http://dx.doi.org/info:doi/10.5281/zenodo.3964368

Noardo, F., Arroyo Ohori, K., Biljecki, F., Ellul, C., Harrie, L., Krijnen, T., & Stoter, J. (2020b). *Reports about the software behaviour delivered within the GeoBIM benchmark 2019—Task 1*. Retrieved from http://dx.doi.org/info:doi/10.5281/zenodo.3964445

Noardo, F., Arroyo Ohori, K., Biljecki, F., Ellul, C., Harrie, L., Krijnen, T., ... Stoter, J. (2020c). *Reference study of CityGML software support: The GeoBIM benchmark 2019 - Part II*. Preprint, arXiv:2007.10938.

Noardo, F., Biljecki, F., Agugiaro, G., Arroyo Ohori, K., Ellul, C., Harrie, L., & Stoter, J. (2019). GeoBIM benchmark 2019: Intermediate results. *ISPRS International Archives of Photogrammetry, Remote Sensing & Spatial Information Sciences*, *42*(4), 47–52.

Noardo, F., Ellul, C. C., Harrie, L., Overland, I., Shariat, M., Arroyo Ohori, K., & Stoter, J. (2020). Opportunities and challenges for GeoBIM in Europe: Developing a building permits use-case to raise awareness and examine technical interoperability challenges. *Journal of Spatial Science*, *65*(2), 209–233.

Noardo, F., Harrie, L., Arroyo Ohori, K., Biljecki, F., Ellul, C., Krijnen, T., Stoter, J. (2020). Tools for BIM-GIS integration (IFC georeferencing and conversions): Results from the GeoBIM Benchmark 2019. *ISPRS International Journal of Geo-Information*, *9*(9), 502.

Papadopoulos, N. A., Sotelino, E. D., Martha, L. F., Nascimento, D. L. M., & Faria, P. S. (2017). Evaluation of integration between a BIM platform and a tool for structural analysis. *Sistemas & Gestão*, *12*, 108–116.

Pazlar, T., & Turk, Z. (2008). Interoperability in practice: Geometric data exchange using the IFC standard. *Journal of Information Technology in Construction*, *13*(24), 362–380.

Peters, R., Ledoux, H., & Biljecki, F. (2015). Visibility analysis in a point cloud based on the Medial Axis Transform. In *Proceedings of the Eurographics Workshop on Urban Data Modelling and Visualisation*, Delft, the Netherlands (pp. 7–12). Geneva, Switzerland: Eurographics.

Petersen, S., Bøving, A., Nielsen, C., & Purup, P. B. (2018). Integration of thermal simulations in Revit for decision support and national building code compliance. In *Proceedings of the Fourth Building Simulation and Optimization Conference*, Cambridge, UK (pp. 407–413). Plymouth, UK: International Building Performance Simulation Association.

Petri, I., Kubicki, S., Rezgui, Y., Guerriero, A., & Li, H. (2017). Optimizing energy efficiency in operating built environment assets through building information modeling: A case study. *Energies*, *10*(8), 1167.

Pratt, M. J., Anderson, W. D., & Ranger, T. (2005). Towards the standardized exchange of parameterized feature-based CAD models. *Computer-Aided Design*, *37*(12), 1251–1265.

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.

Sun, J., Olsson, P., Eriksson, H., & Harrie, L. (2020). Evaluating the geometric aspects of integrating BIM data into city models. *Journal of Spatial Science*, *65*(2), 235–255.