Student thesis series INES nr 495

# Investigating result quality indicators in industry foundation classes to city geography markup language conversion

**Ville-Pekka Soini**

2019
Department of
Physical Geography and Ecosystem Science
Lund University
Sölvegatan 12
S-223 62 Lund
Sweden

Ville-Pekka Soini (2019).
***Investigating result quality indicators in industry foundation classes to city geography markup language conversion***
Master degree thesis, 30 credits in *Geomatics*
Department of Physical Geography and Ecosystem Science, Lund University

Level: Master of Science (MSc)

Course duration: *January* 2019 until *June* 2019

Disclaimer

This document describes work undertaken as part of a program of study at the University of Lund. All views and opinions expressed herein remain the sole responsibility of the author, and do not necessarily represent those of the institute.

# Investigating result quality indicators in industry foundation classes to city geography markup language conversion

Ville-Pekka Soini

Master thesis, 30 credits, in *Geomatics*

Supervisor 1 Lars Harrie
Lund University

Supervisor 2 Per-Ola Olsson
Lund University

Exam committee:
Petter Pilesjö, Lund University
Augustus Aturinde, Lund University

# Acknowledgements

I would like to take this opportunity to thank the people at the Department of Physical Geography and Ecosystem Science in Lund, including my supervisors Lars Harrie and Per-Ola Olsson.

Additional thanks goes out to the partners at 'The GeoBIM benchmark 2019' for providing the data for this master's thesis.

I would also like to express my gratitude towards my fellow students for encouragements during the long hours in the computer lab.

# Abstract

Industry Foundation Classes (IFC) models from the Architecture, Engineering and Construction (AEC) -industries can contain information on building life-cycle entities. Urban planning and related domains are interested in data conversions between IFC and City Geography Markup Language (CityGML) because of this. However, the maintenance and versioning of CityGML life-cycle building data poses a serious challenge. Semantics play a big role in CityGML data contents in addition to converted objects. General methods for testing data conversion result for quality are not strongly present in the prevailing research literature. The evaluation of CityGML file contents and conversion quality are therefore challenging.

CityGML is a Three-dimensional (3D) data format created for storing 3D city data into databases. The exchange of life-cycle data from BIM environments enables more spatial analyses on urban and environmental related data.

This master thesis explores methods that are in use for evaluating conversions and data accuracies within the realms of Building Information Modelling (BIM) and geodata by researching application requirements and measures for quality in a case study.

The INSPIRE directive implementation guides distribution of public domain geodata in the EU and gives instructions on implementing CityGML. Different applications of CityGML and their requirements are leading to the creation of national guidelines. A literature study/review conducted revealed four different main sources for CityGML data requirements.

An explorative case study compares four different IFC models from the GeoBIM benchmark testbench data. The conversion tools FME 2017 and ArcGIS PRO Data Interoperability extension are used to convert the GeoBIM benchmark IFC data to CityGML 2.0 Level-of-Detail (LOD) 3 and LOD4 data. A total of ten test methods are performed to assess the quality of IFC to CityGML conversion data.

The evaluation results for the quality indicators created in the case study reveal that most metrics used for indicating quality of IFC to CityGML 2.0 data conversions are applicable for single LOD4 features but are more difficult to interpret for LOD3.

The results from the study reveal that the conversion methodologies for IFC data should be verified before tackling performance and optimisation issues. Metrics for deriving positional accuracies within the data conversion geometries and those taking advantage of the FME data inspector features are easier to apply. Detailed findings from the case study data conversions revealed more interesting facts about the data evaluation methods and conversion workflows.

There is a severe lack of automated eXtensible Markup Language (XML) formatters for writing CityGML. More study is also required on the documentation of data conversion methods.

**Keywords:** Physical Geography, CityGML 2.0, IFC, BIM, Geodata, Conversion, FME, Quality, Evaluation, GeoBIM

# List of abbreviations

| | |
|---|---|
| 3D | Three-dimensional |
| ADE | Application Domain Extensions |
| AEC | Architecture, Engineering and Construction |
| AIA | The American Institute of Architects |
| BIM | Building Information Modelling |
| B-rep | Boundary representation |
| CityGML | City Geography Markup Language |
| CG | Computer Graphics |
| CRS | Coordinate Reference System |
| EMF | Eclipse Modelling Framework |
| EXPRESS | EXPRESS data modelling language |
| GML | Geography Markup Language |
| GIS | Geographical Information System |
| IAI | International Alliance for Interoperability |
| IDM | Information Delivery Manual |
| IFC | Industry Foundation Classes |
| IFD | International Framework for Dictionaries |
| IoT | Internet-of-Things |
| ISO | International Standardization Organisation |
| LOD | Level-of-Detail |
| MVD | Model View Definition |
| OFF | Object File Format |
| OGC | Open Geospatial Consortium |
| RF | Random Forest |
| SIG 3D | Special Interest Group 3D |
| TIC | Terrain Intersection Curve |
| UBM | Unified Building Model |
| XSD | XML Schema Definition |
| XML | eXtensible Markup Language |

# Table of contents

# 1 Introduction

Three-dimensional (3D) city modelling has grown from the idea of smart cities. Smart city is a concept in which information is distributed and exchanged between devices through a ‘central’ network hub (3D city model) within urban areas. This can for example include ‘smart’ network connected devices and services.

3D city models are more habitually constructed due to their recognised value in urban planning and Architecture, Engineering and Construction (AEC) industries. Although some information in 3D city models is only meant to be visualized, 3D city models can contain and process information for the smart city networks (OGC 2015). This makes the 3D city model, if managed correctly, a powerful tool for multiple (AEC) related domains like urban environment planning and facility management (FM) (Mohanty et al. 2016; Laakso and Kiviniemi 2012).

Applications that use 3D city models can have demanding data requirements. For example, closed geometries or multiple floor heights together with other geometry related attributes such as material composition and density (Mohanty et al. 2016; Laakso and Kiviniemi 2012). There are many ways to collect geodata for 3D city models stored in City Geography Markup Language (CityGML) (Biljecki et al. 2015). Most methods use remote sensing techniques and require significant storage space and investment. The maintenance of 3D model information created in this way is challenging because even small changes in the modelled environment make the 3D model outdated (Prieto et al. 2017). Therefore, it is of major interest to find other reliable and cost effective ways to collect geodata from outside sources like Building Information Modelling (BIM) models. For this purpose, BIM model contents have to be converted into a compatible format. BIM is a name in use for many type of data modelled to 3D for workflow and design management within different domains. Because of this, the BIM data content varies a lot between different applications. The format Industry Foundation Classes (IFC) was created by the International Alliance for Interoperability (IAI) to allow different types of information contained within BIM files to be transferred and viewed by different participants involved in a project. Often IFC files contain what is called a combined view that usually consists of AEC and FM data. Combined views contain entities that are imported to the IFC file from their own design disciplines. The entities contained within IFC files can be used to enrich and create 3D city models (Billen et al. 2014).

When it comes to 3D data storage requirements in municipalities and the urban planning domain, the open standard CityGML format is designed for 3D city model data storage in mind. CityGML is an Open Geospatial Consortium (OGC) standard that uses eXtensible Markup Language (XML) and a Geography Markup Language (GML) application schema to facilitate 3D model interoperability. While cumbersome in nature, it allows for the storage of semantics and topological relationships together with 3D geodata (OGC 2012). Here interoperability means that software developed by different companies works together and data like geographical information (geodata) can be stored to and read from database(s).

Numerous efforts have been initiated for BIM and geodata integration aiming to achieve IFC and CityGML interoperability. Challenges faced when converting IFC data into CityGML consist of different aspects but the two main obstructions are the interpretation of georeferencing data and the different geometry representations between IFC and CityGML. These challenges mean that there is no standard way to convert IFC files to CityGML format, although the software in use may support the format conversion (Arroyo et al. 2017). Furthermore, the inspection of translation results is left to the user and often when dealing with large amounts of data it is hard to say how much time and effort is required to fix issues between the original IFC model and the translated CityGML files.

This master thesis explores methods that are in use for evaluating conversions and data accuracies within the realms of BIM and geodata. Common terminology is also borrowed from ICT (Information and Communication Technology). The purpose of this master thesis is to find common ground on how IFC to CityGML conversion result quality is to be measured.

The thesis work is linked to the GeoBIM benchmark project which is a European Spatial Data Research (EuroSDR) project studying the integration of CityGML and IFC standards by conducting a study on software support for open standards of city and building models. GeoBIM benchmark aims to provide insight into problems within this integration process by benchmarking support from existing software tools for IFC and CityGML conversion. The project is coordinated from the Netherlands by Delft University of Technology with Lund University as one of its partners. The GeoBIM benchmark scientific initiative is funded by the International Society for Photogrammetry and Remote Sensing (ISPRS) and EuroSDR. Results from the practical part of this master's thesis are to be submitted to the GeoBIM benchmark project (GeoBIM benchmark 2019).

The data that is converted to the CityGML format from an IFC file is referred to as converted data for the rest of this thesis.

## 1.1 Problem statement

The use of 3D city models has increased in recent years due to applications for visualizing and analysing data within a 3D city model and linked databases (Biljecki et al. 2015). This means that maintaining and updating 3D city models is required. Parts of the maintenance and update processes can be achieved automatically using remote sensing data, although some manual 'fine tuning' is often required for the datasets afterwards (Maas et al. 1999; Suveg et al. 2004).

A substantial challenge remaining for urban planning and related domains is the integration of BIM model information. This is why the IFC standard is being looked at as an intermediate between the BIM and Geographical Information System (GIS) domains with the intention to find a reliable and easy data source to enrich application in both domains through data stored in the IFC standard. However, the quality of conversion results from IFC to CityGML is difficult to assess. This is partly due to the requirements changing over time for 3D city model applications and data storage as well as the multitude of participants in large projects. For this reason it is vital to find uniform methods to evaluate data-format conversion results. With standard evaluation methods for IFC to CityGML conversions it will be easier to say if geodata in a server is compatible with intended applications.

## 1.2 Aim of the study

The aim of this study is to find metrics for IFC to CityGML conversion quality evaluation. The research questions are as follows:

1) What requirements are set for CityGML data by AEC-industries and different authorities in the European Union?
2) What key quality metrics can be identified for the converted data?
3) What methods/tests can be performed to assess the quality of the IFC to CityGML data converted in the GeoBIM benchmark 2019 project?

## 1.3 The method of the study

The method of this study consists of five different phases. Phases one and two are conducted with the help of a literature review. In phase one the requirements for detailed city model data and applications are studied. Phase two is a literature review of methods for evaluating data conversions and the key metrics to assess converted IFC data quality. In phase three the requirements from phase one are linked to phase two results. Phase four creates, devises and refines techniques for assessing the converted data quality by relying on the phase three results. Phase five implements, tests and evaluates the techniques for assessing conversion quality by trying them out on resulting CityGML files. The flow of different phases in the method is illustrated in Figure 1.1.

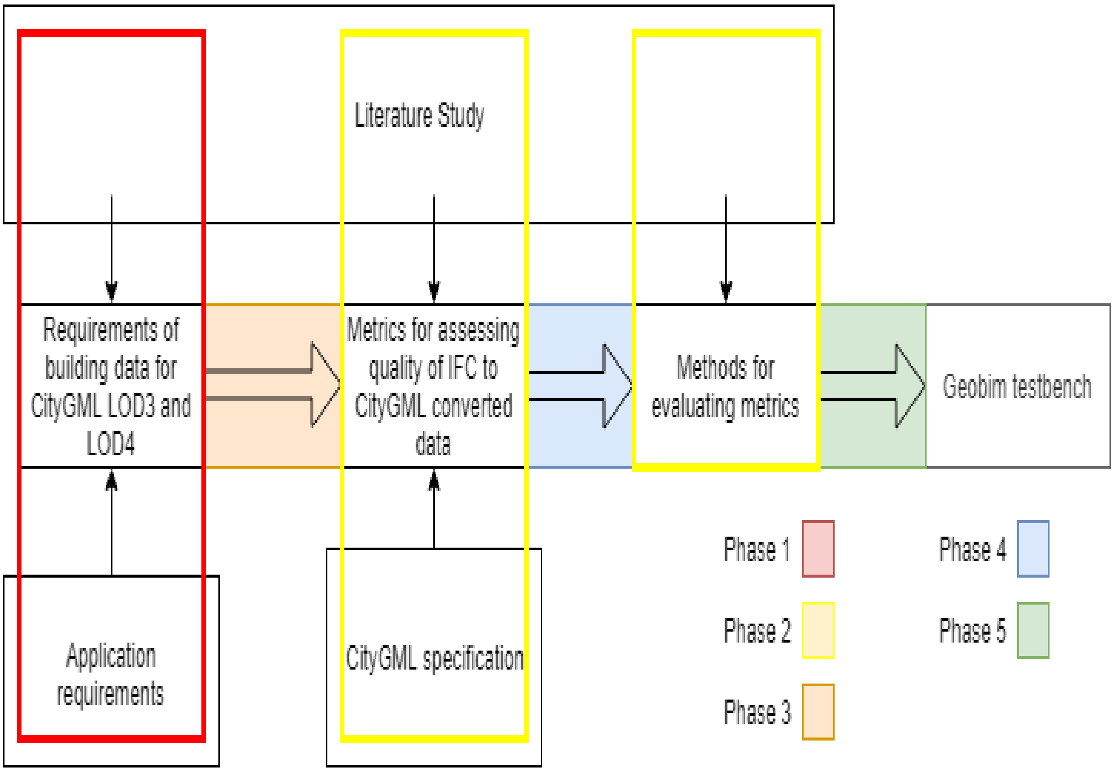


Figure 1.1: Methods of the study.

The first research question of the study is targeted by deriving common requirements from related CityGML applications in literature. A template for CityGML data contents is created with the help of the literature review.

The research question number two is targeted by exploring the IFC and CityGML file contents and surveying existing quality metrics from literature. A literature review is conducted on IFC to CityGML conversion results and the evaluation of these results.

The third and final research question is targeted by devising techniques for deriving key quality metrics from IFC to CityGML conversions and evaluating them in a case study.

## 1.4 Disposition

This master thesis is divided into six main chapters. The first chapter introduces the subject of the study. The chapter two describes theory and related work in regards to IFC and CityGML conversions. In the chapter three previous findings from the literature study are combined to devise methods for assessing IFC to CityGML converted data quality. The chapter four introduces the case study and individual workflows for the conversion tools and the provided data. The results produced by the conversion tool workflows are presented. The quality metrics relevant to the workflows are explained for the GeoBIM test bench data. In the chapter five the results are reviewed and their significance in relation to the master thesis aim discussed and shortcomings of the study are addressed. In the chapter six the thesis is concluded by reflecting on the findings and their implications for future studies.

## 1.5 Limitations

Limitations in this master's thesis:

1) The software selected for this study is limited to solutions that work without extra involvement and do not need any specific expertise to start them up. This means that conversion methods/approaches, that require significant time investment to get them working, are not viable for the thesis.
2) Systematically generated or synthetic datasets are not included in the evaluations because they rarely contain IFC geometry.
3) Georeferencing is only handled in the context of IFC files provided by the GeoBIM test bench.
4) Testing of devised IFC to CityGML quality metrics is limited to files that have not received automatic geometry correction treatment.
5) Since geometry and topology are in practice included in the same feature in the module structure of CityGML, the topological aspect of conversions is not being evaluated. This topological model where the topological aspects and geometry are joined in CityGML leads to a dilemma where some features can only be modelled using incorrect topological definitions. These topological structures could be corrected by implementing a separate topological structure via code list into CityGML features. The validation methods that are in use for CityGML do not care about the topological structure and only require that modules deployed contain valid geometry types.
6) The CityGML data created in the conversions is of LOD3 or LOD4 depending on selected data conversion approach
7) No Application Domain Extension (ADE) is constructed.

# 2 Literature review

## 2.1 City Models

### 2.1.1 General

Three-dimensional (3D) city models are digital representations of urban environments. The concept of a 3D city model is backed up by Smart City thinking where information models are interconnected together with Internet-of-Things (IoT) expanding the 3D city model capabilities. The applications of 3D city models are numerous (Biljecki et al. 2015; Chowdhury et al. 2016; Mohanty 2016):

- urban planning
- risk assessments
- navigation
- advertisement
- visualization
- other analyses and simulations
- provision of data for IoT

A 3D city model is though to be a key part of a framework to share digital information in the context of Smart Cities. With the numerous application expectations imposed on the concept of 3D city models, a solid plan for updating and maintaining data in a 3D city model is required. However, the maintenance of a 3D city model is not a small task and needs a proper plan of action. An example of data layers in a 3D city database is depicted in Figure 2.1. In addition to collecting well documented data assets to a 3D city model connected database, the terms on which this data is to be used, need to be agreed upon (Prieto et al. 2017).

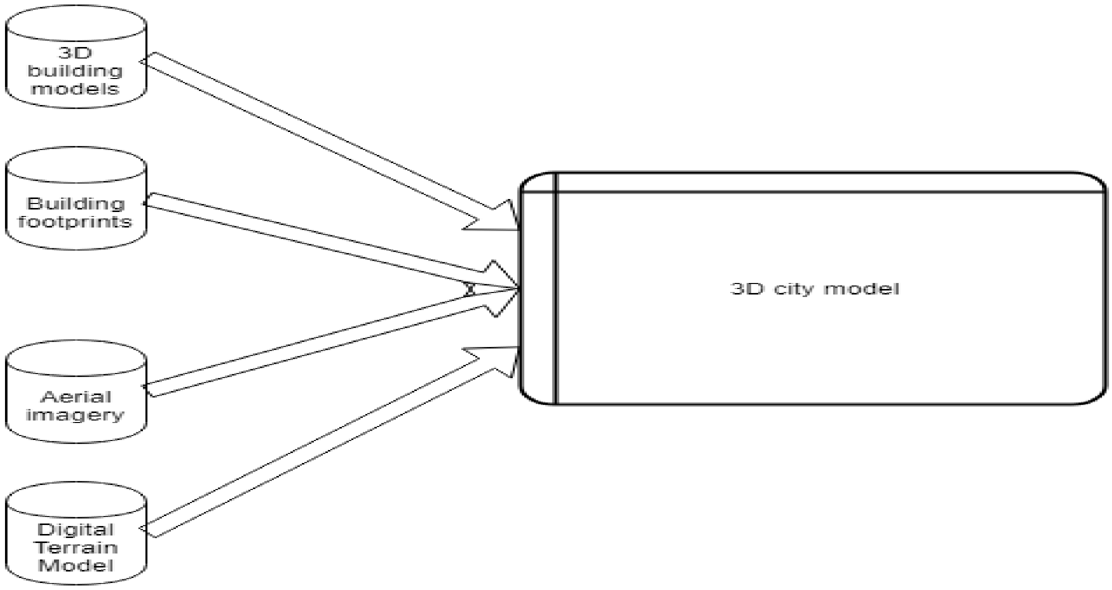



Figure 2.1: Example of 3D city model database contents that need maintaining in a 3D city model (created after Steinhage et al. 2010).

A 3D city model borrows a concept named Level-of-Detail (LOD) from computer graphics (CG) rendering that reduces the time it takes to render objects further away from the camera or point of view. In 3D city models this is also used to indicate the general geometry and attribute content in the LOD classification. The difference between LOD techniques in CG and 3D city modelling lies in the generalisation objective. While the original method simplifies LODs in CGs and is used to only satisfy visual appearance, in 3D city modelling it also maintains object structures (Figure 2.2). For example in CityGML this means that surfaces within the 3D city model can be semantically related to one or more solids if their Boundary representation (B-Rep) facets allow this. B-Rep is a way to represent 3D objects by defining them as a boundary (facet) presentation of bordering surfaces.

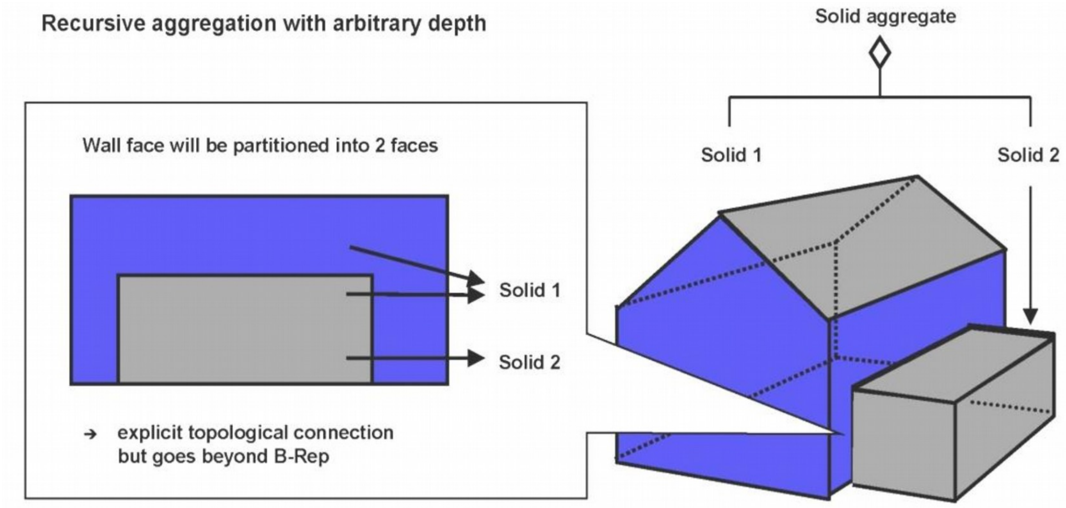



Figure 2.2: The Level-of-Detail (LOD) used in 3D city models also preserves object structures while simplifying geometry (OGC 2012).

LOD plays an important role when planning data acquisition for updating and maintaining a 3D city model because the amount of data needed for each dataset collected depends on the LOD requirements within the 3D city model. The 3D city model construction also aims to reduce the costs deriving from multiple simultaneous data collection efforts. The 3D city model is used as a tool/platform for multiple applications. Thus, the data linked to 3D city models needs to be checked or controlled for quality so that it fulfils the requirements for data interoperability (Biljecki 2017).

3D city models can be used in analysing and visualising data in different ways. One of the major user groups to benefit from 3D city model data contents are urban planners. In fact, nearly all use case descriptions in Biljecki et al. (2015) are urban planning related. For such a large application base it is important that information linked to 3D city models is constantly up to date. Urban planners can gain insight to previously unseen processes with the help of big data by combining it with 3D city models. However, when it comes to large scale usage of 3D city models in urban planning, there are challenges that have to be solved beforehand. Most use cases of 3D city models in use do not require complex geometries or high levels of LOD. From a city planning perspective area and volume are important factors when it comes to detail planning. The main concern for datasets, that can be used in city planning applications, is their applicable extend. Often continuous data and/or data with higher levels of detail is not feasible to obtain without starting a time consuming modelling effort. For most city planning applications in use, this means, that heavily generalised data must be used. By extracting more detailed modelling resources from Industry Foundation Classes (IFC) entities it is possible to design continuous 3D data analysis processes that allow for results to be derived from multiple levels of detail on objects with small and large scale.

3D city models in municipalities are often created as part of a pilot project and can therefore be short-lived. For a 3D city model to offer continuous support for applications a degree of maintenance is required. Often 3D city model status is dependent on government mapping agency data acquisition plans. In practice this means that applications requiring 3D city model information either in the form of generalised geometries or linked attribute datasets are limited to this data collection schedule. Additional data collection is often required to maintain the datasets of a 3D city model updated for the duration of the pilot. After the pilot project has run its course a lot of datasets used in 3D city model pilots never see the light of day due to licensing reasons.

### 2.1.2 City Geography Markup Language

The City Geography Markup Language (CityGML) standard which is based on International Standardization Organisation (ISO) 19107 and ISO 19109 was created to facilitate the implementation of sustainable maintenance goals in 3D city models. The current version of CityGML is 2.0. This is an implementation of the Geography Markup Language (GML) TC211 (OGC 2012). However, there are some slight changes on how geometries work in CityGML. These changes affect position listings, surfaces and basic geometry types. All geometry elements in CityGML must have a Spatial Reference System (SRS) definition that is either inherited from parents or defined in the local geometry (OGC 2012).

Each Level-of-Detail in CityGML can contain multiple simultaneous entities with different entity versions (Figure 2.3). However, if no alternative modelled data with less generalised visuals exists within a level-of-detail, there are well-defined rules for LOD that divide them into 5 different categories based on general structural complexities. These categories range from LOD0 to LOD4 in CityGML 2.0. CityGML can represent different aspects of entities geometric, semantics, topology and appearances. Appearances in this context can be though for example as textures or analytical result layers (OGC 2012; Biljecki et al. 2016b).

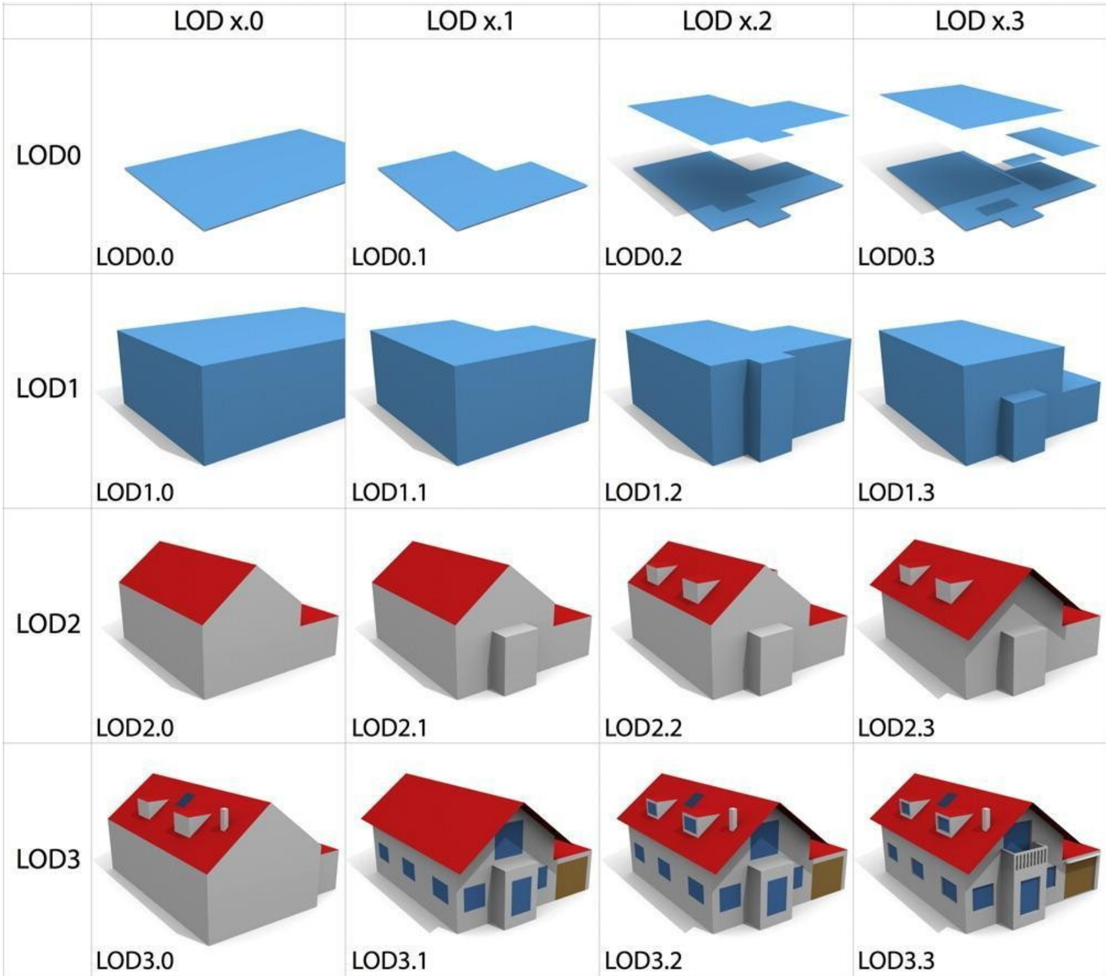


Figure 2.3: Suggestion for LOD versioning contents (Biljecki et al. 2016b).

CityGML is modular and can be extended to accommodate more features. Each module is associated with a Namespace identifier in XML. Geometry in CityGML is represented by different structured feature groups that are associated with the *gml::_Geometry* namespace. The default or normative XML schemas defined in CityGML 2.0 are (OGC 2012):

- Core
- Appearance
- Bridge
- Building
- CityFurniture
- CityObjectGroup
- Generics
- LandUse
- Relief
- Transportation
- Tunnel
- Vegetation
- WaterBody
- TexturedSurface.

The data model for CityGML composes of core and additional thematic extensions. The thematic extensions can be extended even further by applying Application Domain Extensions (ADE). ADEs are a way of adding custom application schemas and can be used to add new properties to existing ones and/or to create new modules. Other possibilities in CityGML include *ClosureSurface*, Terrain Intersection Curve (TIC) and external referencing. *ClosureSurface* is a feature type that makes calculation of volumes possible even when modelled solids are not closed. Defining a TIC corrects 3D objects by having them stick to the terrain. Code lists that store custom enumerative attributes can be used to define external references to linked databases (OGC 2012).

The OGC CityGML Standards Working Group (SWG) and SIG 3D are working to bring out a new version of CityGML called CityGML 3.0. This new version is envisioned to make the CityGML standard more approachable by including extra encodings. CityGML is getting a new Core model that includes features for representing point clouds (Space and *SpaceBoundary*) and is using a new LOD definition model for inside and outside surface representations (3 LODs). This Core model has also been enhanced with a new class '*AbstractToplevelCityObject*' that makes constraining '*CityModel*' members possible. Some features that allow CityGML 3.0 to better accommodate data structures present in other standards have been added. This means changes to modelling restrictions by allowing for divisions of space within the modules. New modules presented in CityGML 3.0 include for example: Construction, Versioning and Dynamizer. The intended use for 'Dynamizer' is time series data. 'Versioning' deals with different data states and INSPIRE while 'Construction' is used to transfer (inherit) surface properties (in line with INSPIRE) to other modules. The third iteration of CityGML has better support for IFC with the introduction of the new classes and modules (Kutzner and Kolbe 2018). The development of CityGML 3.0 is on its final stretch and will extend on the definitions provided by the CityGML standard. The conceptual model of the standard is based on GML, Relational DB and JSON encodings. This allows for existing datasets to be integrated into CityGML 3.0. CityGML 3.0 is going to have only 3 LODs but supports indoor and outdoor data for defined LODs in the 3D city model (CityGML 3.0 Development).

## 2.2 Building Information Modelling

### 2.2.1 General

Building Information Modelling (BIM) is a term commonly used in Architecture, Engineering and Construction (AEC) and Facility Management (FM) 3D models to incorporate attributes and process into 3D modelling within their own disciplines. The goal of BIM is to advance resource management and make workflows more efficient by utilizing different aspects of BIM toolkits. A BIM tool is a program that allows the management and construction of elements in a BIM software. The toolkit applications within a BIM software are measured based on Dimensions of BIM. Currently there are seven dimensions while more are being added as the capabilities of BIM tools continue to grow. Each of the dimensions stands for a problem that BIM tools can address (Figure 2.4).

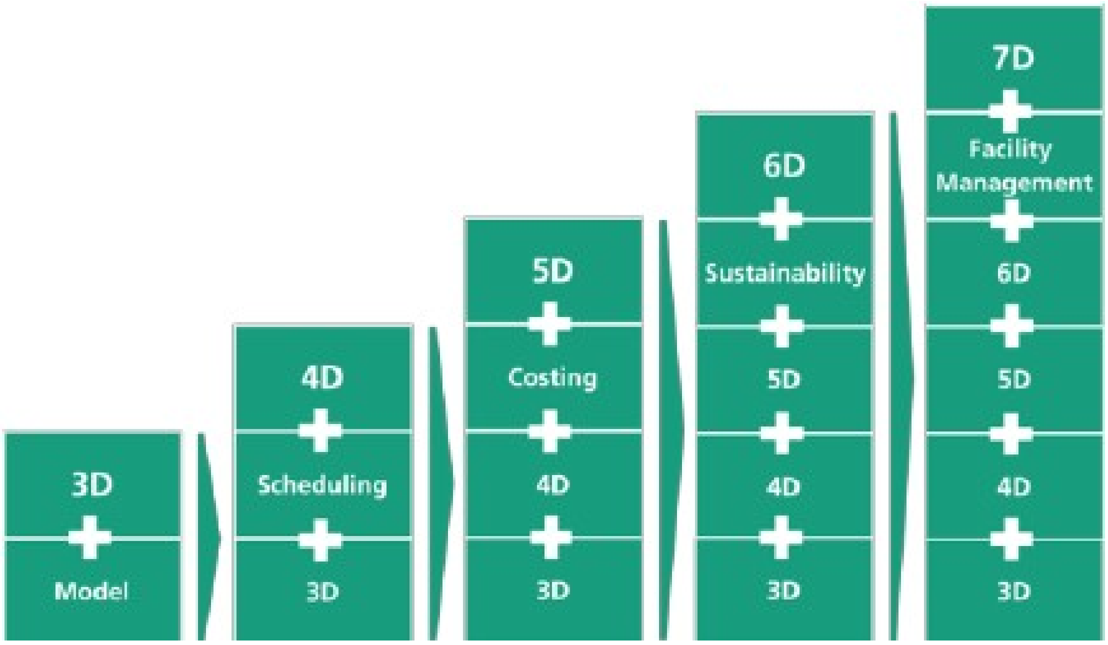



Figure 2.4: A depiction of different BIM dimensions. Features from previous dimensions stack into the next dimension (Dallasega et al. 2015).

BIM software comes in many forms because of the multitude of application areas within ACE and FM. Some of the major BIM vendors are:

- Autodesk Inc.
- GRAPHISOFT (NEMETSCHEK INC)
- Bentley Systems Inc.

Software suits have their own proprietary file formats. Proprietary file formats cause challenges for interoperability between BIM and 3D city model actors.

BIM software consists of tools that enable the use of 3D models and their related attributes in AEC and FM industries. In general, if not otherwise stated, these 3D models drive a certain purpose in their designated discipline and workflows. How much information a BIM based model contains is based on its Level of Development (LoD). LoD is used to describe object development stages in BIM.

LoD is a specification developed by The American Institute of Architects (AIA) that is used to communicate characteristics and elements of the different existing Building Information models (BIMs). This specification is currently yearly updated and is constantly evolving. The basic concept of Levels of Definition has definition levels from 100 to 500. The definition of these levels of development is, however, quite abstract and left to the BIM practitioner. An interpretation can be written out as (AIA LOD Specification 2018):

- Level of definition 100: Model cost and placement is known.
- Level of definition 200: Model presented as a wireframe with approximate placement.
- Level of definition 300: Model is presented as  a 3D model, linked content and coordinates.
- Level of definition 350 (specified by BIMForum): Model is a unique presentation, linked content and coordinates.
- Level of definition 400: Model contains related parts in detail, linked content and coordinates.
- Level of definition 500: Model contain user field specific additions to level 400.

To clarify, the levels of definition are used as tools to measure how far from completion a specific feature (object) in the BIM model is during its creation.

Level of maturity is a United Kingdom (UK) originating conceptual way to measure BIM benefits. The BIM maturity levels are commonly depicted using a triangle similar to that in the top part of Figure 2.5 (Laakso and Kiviniemi 2012).

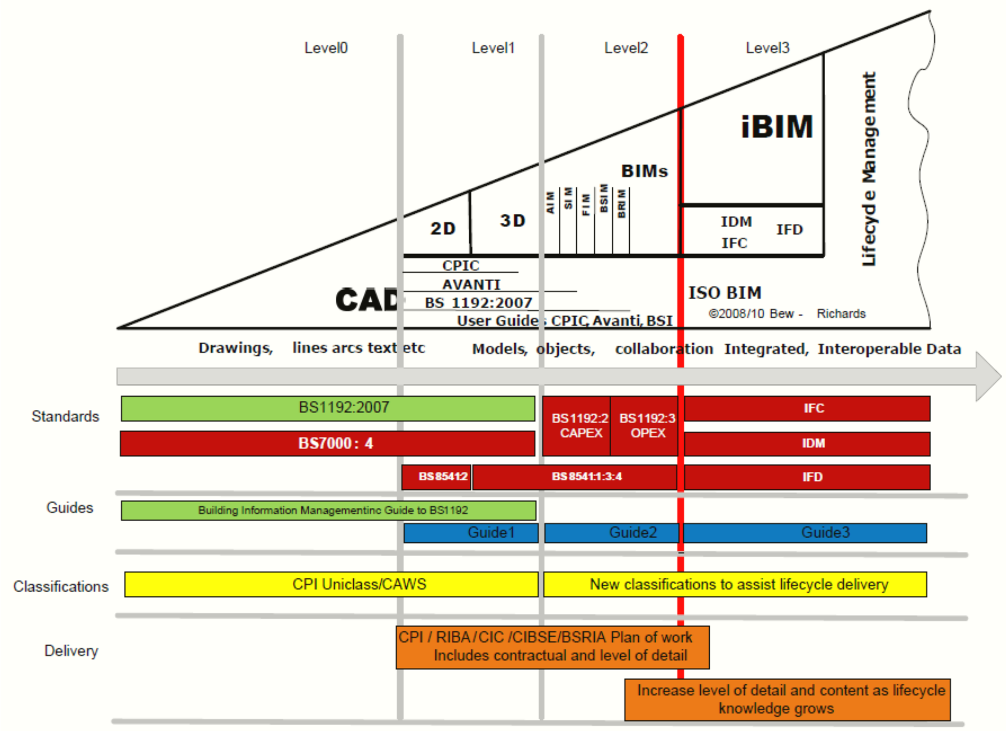


Figure 2.5: BIM advancement and standard development from UK's perspective (BIMTalk).

It is observed that 3D models constructed using BIM are fragmented and divided in data content based on the design and user requirements in a sector of AEC and/or FM discipline that has BIM incorporated into their workflow processes. Furthermore, there was a divide in terminologies and fragmentation of process requirements that has led to effort for universally defined exchange formats that could bridge information created with BIM. The efforts have led to two ISO standards known as International Framework for Dictionaries (IFD) ISO 12006-3 and Information Delivery Manual (IDM) ISO 29481 that are used as a basis for the open standard ISO 16739 (IFC) managed by BuildingSMART (Laakso and Kiviniemi 2012). As an example of this development process Figure 2.5 depicts the situation of existing standards, guides, classifications and delivery requirements in relation to level of BIM maturity in the UK.

### 2.2.2 Industry Foundation Classes

There are many proprietary file formats used in BIM today for different disciplines. These file formats hold data on construction and design field projects and workflows to help the managing of information. Depending on the 'maturity level' of a BIM the format can hold 2D and 3D data related to disciplines. BIM that is of level three maturity contains sequenced workflows, cost estimates and life-cycle management information for all the disciplines involved. In many cases, especially when bigger projects are undertaken, subcontractors are required. This has created a need for non-proprietary file formats. One such format is the (IFC). These file formats make information exchange possible between different BIMs. IFC is defined in the EXPRESS data modelling language but the data can be described also in XSD (XML schema definition) file that contains the EXPRESS schemas and data in IFC4 (Laakso and Kiviniemi 2012)

The BuildingSMART International Ltd. is working together with the International Standardization Organisation (ISO). In this context BuildingSMART uses a triangle pattern to describe the interconnectivity of BIM standards. The Processes modelled into IFC come from a concept known as Information Delivery Manual (IDM). There are five different types of geometry in the current IFC version. These are (IFC4 Documentation):

- Tessellated surface models
- Constructive solid geometries
- Surface models
- Swept solids
- (Body) Boundary representation (B-Rep) geometry

IFC is currently a loosely defined data transfer file format. What this means is that although the format is standardised there is no consensus on how much data each defined schema structure in IFC should contain. For this reason, there are conversions that are required by software developers for their own needs. While this kind of approach to standardisation allows more freedom for designs, it makes the interpretation of extended data within IFC a challenge. Thus, BIM software often has to rely on plug-ins to handle IFC export and import between BIM software. The interpretation of the IFC files depends on the configured properties and types for entities within BIM data. For these data transfer processes IFC has introduced the concept of Model View Definitions (MVDs). In IFC4 these MVDs are named Reference View and Design Transfer View after their conceptual contained properties and intended use (Laakso and Kiviniemi 2012; IFC4 Documentation).

An MVD needs to be defined before exporting data to it. Data between the BIM model in use and the defined MVD has to be mapped for information to be exported (and read) correctly from the IFC file format. Some objects or entities in BIM can have many variations that need exporting to IFC. This makes the mapping of objects challenging. A software vendor can obtain a certificate for MVDs based on tests conducted for support. This means that commercial software can be compatible out-of-the-box with commonly used MVDs (Laakso and Kiviniemi 2012).

Part of the motivation to use data stored within IFC files is their ability to support life-cycle data. Normally this can involve details like entities needed for building maintenance or additional attributes on material properties and ageing. Life-cycle data consists of attribute and entity instances that provide a continuous 'snapshot' from the building design process to its tear down (Figure 2.6). The full reuse of such data assets is yet to be discovered but for this to happen the data needs to be first stored into a flexible data storage format.

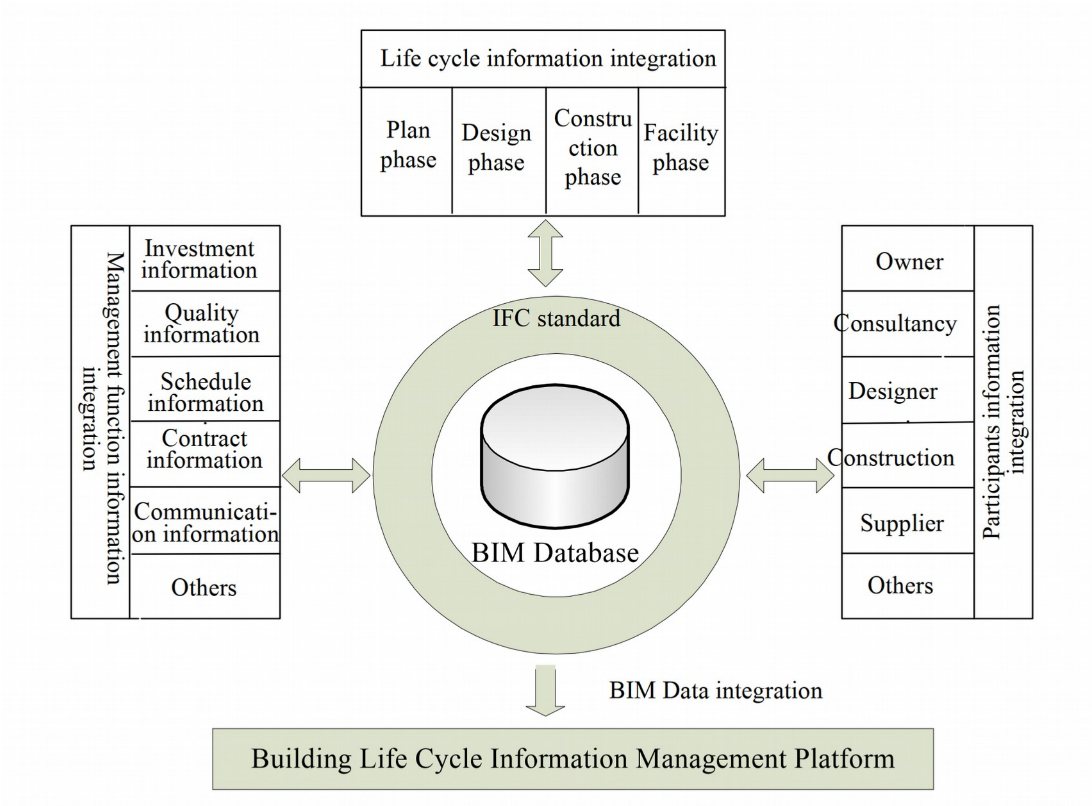



Figure 2.6: Examples of life cycle data contained in an IFC model (Qing et al. 2014).

## 2.3 Conversions between building information models and city models

When 3D data is converted between formats, there is usually a phenomenon called ‘data loss’. This means that when data is converted between two complex systems, a portion of it is left unused and/or discarded; the data is lost in the conversion. When a conversion is lossy, it usually means that a big portion of the information contained within the starting format is not carried over and can lead to redundant data collection and/or difficulties with maintenance. Conversion can be either unidirectional or bidirectional (Deng et al. 2016). Floros et al. (2017) is an example of research that investigates the integration of IFC and CityGML for 3D city models. The IFC also registered as ISO 16739 is practically the only open standard used to exchange BIM data between different software in the BIM sectors. Therefore, it is logical that the focus for translating BIM data into geodata is on this standard. The current version of IFC is IFC4 Addendum 2. An older version named 2x3 TC1 is commonly used as well.

Converting BIM data formatted in IFC to geodata has a multitude of applications that are tied together with the concept of 3D city model data storage and Smart Cities. The majority of case studies is focused on one application at a time. While many conclusions reach favourable results in this regard, the true goal of interoperable geodata is often not reached because of insufficient data validation. The consequence is that the data saved into 3D city models cannot be used without friction with other software/applications.

The reason why conversions from IFC to CityGML are challenging, is because of the different modelling approaches (CSG for IFC and *gml_Geometry* features for CityGML) used when creating the models. For this reason uniform models for IFC to CityGML conversions are practically non existent. The B-Rep geometry type present in both formats is an exception to this. It allows to write entities modelled in B-Rep straight to CityGML as long as they follow the right presentation structure (*gml_surface*). Often translated datasets contain gaps and/or there are missing elements that have not been modelled correctly in a IFC geometry for the converter to produce acceptable results. Other common errors are non-planar entities, overlap between objects and georeferencing, or just plain orientation errors in models (Bilejecki et al. 2016a).

The OGC Quality Interoperability Experiment was launched to investigate the issues related to challenges faced when performing dataset conversions from IFC to CityGML. The aim of the project was to provide guidelines for successful implementation of CityGML conversions. In the experiment the prerequisites for successful conversions are considered and different CityGML data validation methods used by software are compared. The intent was to form a uniform model/framework for validating CityGML geometries and topology (OGC 2016). The guidelines provided by the experiment suggest setting tolerances for geometric validation objects. In total three different suggestions are proposed for tolerance requirements. The implementation of said tolerance parameters aims to create valid geometries. In this context the recommendations to include only roof overhangs as *Multisurface* elements and to model *BoundarySurfaces* as volumes is recommended. No definitive conclusions are reached regarding semantic validation methods, although it is noted that there should be similar constraints and tolerances implemented for semantically relevant objects. The practical implementations of the experiment suggestions are to be considered individually for each use case because data schemas and requirements for applications change over time (OGC 2016).

Since CityGML cannot support all information in IFC models (combination views), a natural way of extending the details that can be stored into CityGML format is using ADE. However, there are some drawbacks from using this solution. Using an ADE to extend the default CityGML schema support means that validation of the core data within CityGML with XSD is not enough any more. The new ADE contents needs a separate validation process and added support in software (Stouffs et al. 2018).

### 2.3.1 Conversion frameworks and geometric processing

Isikdag et al. (2009) defined a framework that is used to transform information from IFC to CityGML. The steps in the framework are as follows. Map objects from IFC to CityGML LODs. Build algorithms to implement rules for geometric simplification or define a new MVD to facilitate this. Define remaining semantics to be reconstructed. This is the general framework still in use today when mapping IFC to the CityGML schemas. Other frameworks, such as the Unified Building Model (UBM) by El-Mekawy et al. (2012) and a JDK 7 based instance comparison framework by Deng et al. (2016), support bidirectional data-flows.

General conversion methodologies to enrich 3D city models with semantic information are divided into addition, aggregation and generalisation. In IFC the information that is moved between software environments is implemented as MVD's. The generalisation of 3D model data is important when deciding what LODs should be assigned to semantics and geometries inside the 3D city model construct. Aggregation on the other hand is in many cases used to automatically fix disjoin geometries between building complexes. Addition of information in the case of CityGML 3D city models is important because it allows the addition of new domain information and semantics via ADE. Another important aspect of 3D city models is the quality and size of facet textures. Depending on the advancements in processing power and 3D rendering technologies; the 3D city models and LODs can be created in many ways. In many cases where high LOD data can be collected fast it is most beneficial to use this data as a starting point and proceed by generalising data in the 3D city model for lower LODs. However, if a 3D city model is already in use, can importing specific information be more convenient and end up consuming less resources (Billen et al. 2014).

Applications that convert IFC information to CityGML usually need to have access to both data schemas or alternatively to a predefined 'rule set' applying for all data structured in the models. This structure/ontology is used to read the information within the datasets and then converted to the other. Some ontologies are unidirectional and only support CityGML to IFC or IFC to CityGML conversions. There are several ways to reach the goal of mapping features for conversions. Mapping can be a simple one-to-many process where features from the IFC schema are associated to existing or extended datasets in CityGML. While this does not usually provide uniform conversion frameworks, it gets the job done for specific domain related data in the IFC MVD. The mapping of features from one format to another can also be done using what is called a reference ontology. This process involves the same mapping principles of one-to-many but uses a unified ontology reference that contains associations for both format features, acting as a middle reference for the format conversion. Lastly it is possible to build a feature map for unique IFC entities based on their instancing. This is a time consuming method since the entity structure in IFC can turn quite complex. However, if enough restrictions are applied to a schema it can provide one-to-one mapping rules for targeted information (Deng et al. 2016).

There are several approaches for schema matching to make this process faster. Ranh and Bernstein (2001) describe most common concepts and approaches used in schema matching and building. Currently a common problem for matching IFC and CityGML ontologies is how to define sufficient constraint based ontology rules. Considering IFC is capable of storing multiple instances of one entity and supports life-cycle management activities, it would be beneficial to employ an approach that can distinguish between differently versioned data in the IFC format (digital twins). Approaches to schema matching can also follow custom schema matching rules, that are based on BIM application workflows or entity processing order, to make data conversions more seamless between practitioners. The goal is to have the conversions from IFC to produce satisfactory results for each intended application user group (Ranh and Bernstein 2001).

While some geometry and topology can be read into ADE and the CityGML schema, others require a mapping approach and conversion framework. IFC model semantics mappings pose a challenge because they are not always related to neighbouring features. This is why conversions are often carried out by separating the data into multiple parts: geometry, topology and semantics. An example that uses a common conversion workflow with this approach between IFC and CityGML is shown in Figure 2.7.

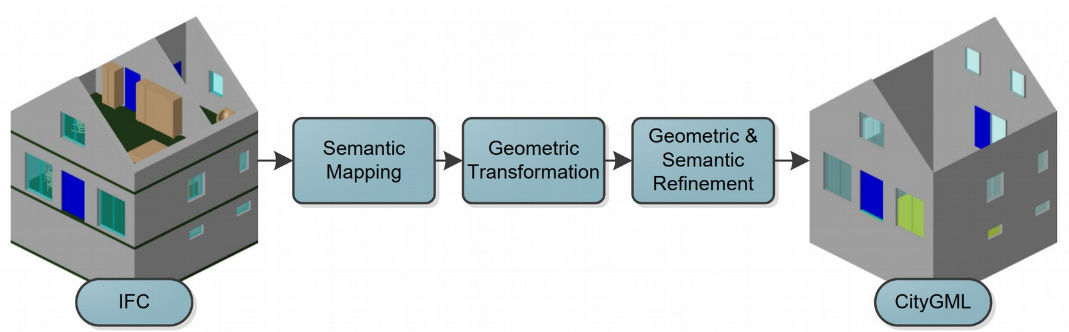



Figure 2.7: A common workflow for IFC to CityGML conversions (Donkers et al. 2016).

The features  converted in a conversion framework or application schema can be presented using a graph grammar. Graph grammars like Stouffs et al. (2018) are a great tool in the construction of conversion workflows because correspondences between different depicted feature types are visible. This can help enormously when reconstructing topological geometries. A good graph also depicts the necessary components in a complete conversion workflow in the form of ordered node structures.

### 2.3.2 Semantic aspects of the conversion

The Special Interest Group 3D (SIG 3D) has published CityGML validation and modelling guides that instruct on building and city object modelling as well as on what should be considered as a valid geometry in the CityGML format (SIG 3D / Quality Working Group. 2017). There is also an example that uses the simple dictionary structure from GML 3.1.1 for code lists available at www.sig3D.org/code lists .

El-Mekawy et. al (2012b) evaluates recent unidirectional IFC to CityGML conversions by taking concepts deployed in IFC file structures and comparering them to an equal/similar structure in CityGML. The result highlighted by the work focuses on IFC hierarchical structure and building space definitions. A conclusion is reached where IFC to CityGML mapping is difficult because of semantic differences between the two formats. Therefore, it is suggested that an extended content mapping UBM is used to describe data and act as a medium when converting IFC and CityGML data.

### 2.3.3 Conversion tools

Some of the found tools only convert IFC data to a specified LOD. If the tool has more functionality, it can usually convert data between different LODs and perform XML schema validation for them. More advanced features include geometry validation and additional analysis tools.

Berlo Laat (2011) has developed an ADE for the open-source BIMserver that acts as a data storage for IFC models. To achieve the conversion from IFC to CityGML six different steps are described (van Berlo et al. 2011):

- Fetch IFC data from BIMserver.
- Run data with IFC Engine DLL to simple geometry (triangles).
- Read data in to the EMF interface with BIMserver.
- Read in IFC properties to Eclipse Modelling Framework (EMF, a common tool for generating JAVA code ) core.
- Get next object from BIMserver.
- Use CityGML4j to convert data into CityGML from EMF core.

The Feature Manipulation Engine (FME) is a commercial software package by Safe Software. FME allows for data translations using set workflows. It can read in data formatted in IFC and separate it into features and attributes that are written into the CityGML format. The translations can be done with custom user defined properties.

Predefined workspaces for FME that can be downloaded from the FME website (FME website). FME also works together with major software developers like AUTODESK, ESRI and Bentley for format translations. The translation results from IFC to CityGML LODs depend on how well the mappings between both schema objects are defined in the FME workspace. FME has tools for geometry and schema validation, but ultimately it is up to the user to double check the resulting CityGML translation (Bengtsson and Grönkvist 2017).

Olsson (2018) shows how volumetric LOD2-LOD3 buildings are generated from IFC files for a somewhat simplified Swedish profile of CityGML denoted "Svensk Geoprocess Byggnad Version 3.0". He applies ray tracing methods to identify correct IFC elements (walls) to extract by using the building centroid as a focus point. An identified point on the centroid aligned ray that matches the building outer wall extend is used as a starting point for the method. The upper and lower surface coordinates are used in determining attachment points for triangulated roof surfaces by interpolation after identifying the exterior walls.

The open source software library, toolkit and geometry engine IfcOpenShell is a collection of tools for working with IFC data.

A conversion solution by Donkers (2013) employs a software *Ifc2CityGML* and is an open source automatic conversion tool developed for converting IFC2x3 models into CityGML LOD3. This software is broken up into two parts according to the readme in its GitHub repository (Ifc2CityGML); the *ifc2off* and the *off2CityGML*. The geometries are stored inside a separate Object File Format (OFF) file. The program relies on its dependencies to do this. The OFF file is converted into a CityGML presentation (Donkers 2013).

The GeoBIM (not to be confused with the GeoBIM 2019 test bench) project started in 2017 has introduced an *IFCLocator* tool that is meant to be an open source alternative for georeferencing IFC models. The *IfcLocator* implementation relies on the *Cesium* javascript library. Additionally the previously separate *CGAL* library is included in the *IfcOpenShell_CGAL* GitHub repository (IfcOpenShell_CGAL). The goal of this project was to research the data extraction from BIM and GIS models and their integration by developing an interface for converting IFC to CityGML. The research group also gives recommendations for preparing IFC files for automated processing (GeoBIM 2017).

A project report by Deng et al. (2016) describes a research project for bi-directional mapping ontologies between IFC and CityGML. Using JDK 7, a beta version tool was developed for mapping different schema instances between IFC and CityGML. The approach uses a meditated reference ontology as a medium between the two standardized schemas. The reference ontology acting as a medium has a mutual data schema devised from IFC and CityGML attached. Special attention was paid to inverse relationships when constructing the ontology for conversions (Deng et al. 2016).

*IfcExplorer* is a software developed by *Research Center Karlsruhe, Institute for Applied Computer Science*. This software has the ability to perform IFC to CityGML conversions for LOD0,LOD1 and LOD2. It seems that the software has an interface similar to *FZKViewer* available at the KIT website (KIT FZKViewer). A description of CityGML Export is found at *ifcwiki* (Ifcwiki 2007) which also mentions *IfcExplorer* as an internal version of FZK.

### 2.3.4 Related work

Synthetic models are 3D models created without actual data conversions. This means that a 3D model is created from ground up using available resources and methods. The quality of a synthetic 3D model depends on outside factors and is not tied to the structure of an IFC file. This is different from models created through data conversions. Biljecki et al. (2016c) creates synthetic city models for test purposes by using a procedural modelling engine *Random3Dcity* (a modelling engine that creates synthetic models procedurally) to address underlying issues with LOD definitions and specifications. They point out that data acquisition workflows dictate the formation of LOD data contents in CityGML and that multiple geometric reference styles are not supported in the current CityGML LOD structure because of cardinality in design. The procedural modelling approach allows the extraction of geometrically referenced modular features in the CityGML format. To find the best geometrical reference styles for a LOD, it is recommended to use test data with similar acquisition methods and modelling requirements. It is concluded that metadata on dataset level is not enough and more accurate levels of information are required (Biljecki 2017).

Biljecki et al. (2017) has conducted an experiment that is focused on investigating generation of LOD1 city models without available elevation data. In the paper a Random Forest (RF) machine learning algorithm is applied to cadastral, geometrical and state statistical data in the Netherlands. By combining the results from RFs with different applied parameters, qualitative validation of building heights predicted within LOD1 city models are done. The results show MAE (maximum absolute error) and RMSE (Root mean square error) for each different RF application. However, it is observed that some of the RF predictors perform even better outside of their original geographical area of context in the example of Leeuwarden.

## 2.4 Evaluation of the conversion

A CityGML file is considered valid by an eXtensible Markup Language (XML) Schema Definition (XSD) validator if all mandatory attributes defined in the CityGML schema exist in the file and all allocated geometry types are allowed in the CityGML version 2.0 schema.

Linking of separate properties to the CityGML database is allowed via code lists. In addition to the data validated by XSD, a CityGML file can have its attributes/semantics defined in a separate linked code list. Using code lists to define relations and attributes between different objects in CityGML reduces data redundancy.

The quality that is sufficient for CityGML is a matter of the user. The requirement for quality is usually in the range of LOD0-LOD3 but for indoor navigation LOD4 is requested. Accuracy of data, intact semantics and surfaces with attributes are not required by most applications currently using high LOD data. Therefore, for current use cases data requirements beyond LOD2 are not common. Most benefits from IFC data come to LOD3 and LOD4 use cases. Thus, the benefits coming from IFC contents remain largely unrealised. However, this does not mean that the development of CityGML or geodata applications has slowed down. With faster data connection speeds over the Internet there are more possibilities for applications to gravitate towards using stored 3D data.

The answer to what is ‘sufficient’ changes with time. Therefore, the future has to be looked into when evaluating IFC to CityGML conversion results and their different aspects. The key insight is that users are able to tell what kind of data is present in the CityGML file. Using the generally accepted convention of LODs is a good start for this. The CityGML modelling instructions created by SIG 3D are a good starting point that could be considered as ‘good quality’. Even well checked data that conforms to the current modelling instructions created by the SIG 3D does not cover all possible future user requirements. The user can nevertheless expect to have at least an idea of how the data in the CityGML file will be like when opening it for example with the indicators of quality introduced in this study.

The common methods that are in use today for assessing the conversion or CityGML quality are:

- XSD validation
- classification of CityGML contents to LODs
- file size
- number of converted objects
- elapsed conversion time
- visual inspection of a CityGML file.

None of these methods are presented in a constant manner in current research papers. This leads to misconceptions about data conversion method quality and the resulting CityGML file. Additionally the IFC to CityGML conversion method may not work with all IFC file structures and versions.

The quality of the conversion and the resulting CityGML file can be evaluated by assessing quality metrics from the executed conversion method. However, getting to the bottom of software and executed processes during the conversion is usually challenging. Making sure that the devised indicators for IFC to CityGML file conversion evaluation results are accessible manually without special software, is a key enabling factor when comparing different conversion quality metrics.

### 2.4.1 Requirements

Requirements for converted IFC data in CityGML are found in many layers. The layers can be authoritative or advisory in nature. Figure 2.8 illustrates the layered structure of CityGML data requirements. The requirements of IFC to CityGML converted data are limited first by active data governing laws and policies. These limitations set the frame for allowed applications in locational context. The next requirement often concerns the classifications of 3D model contents for redistribution. In the European Union (EU) this means aiming to fulfil directives like Infrastructure for Spatial Information in Europe (INSPIRE) and implementing a working national data platform to distribute 3D data (INSPIRE (2014)).

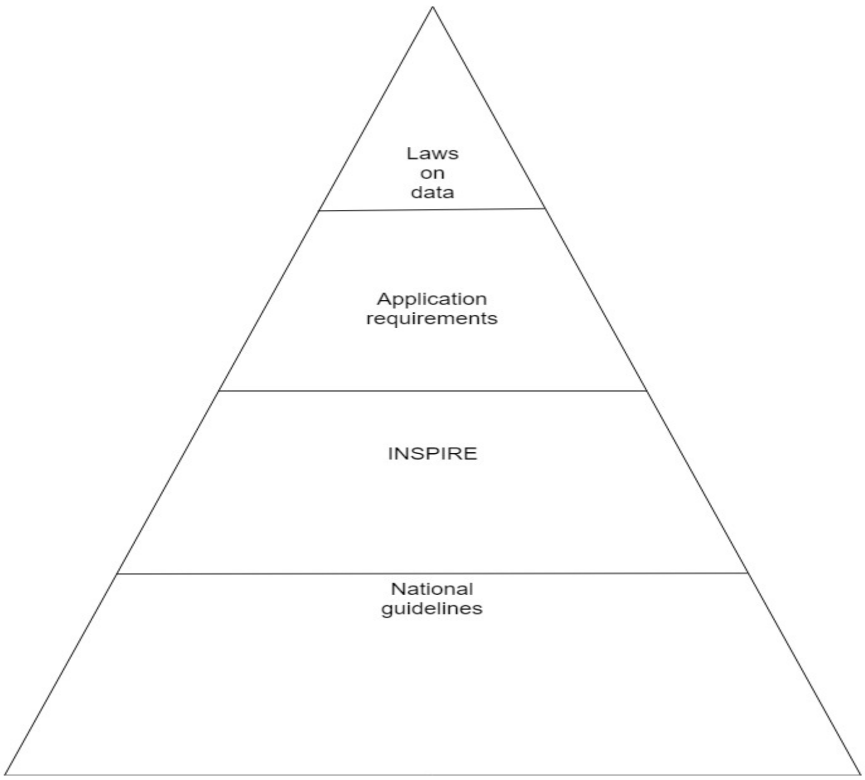



Figure 2.8: Structural presentation of IFC to CityGML converted data requirements.

Lastly, there can be several local guidelines on how the modelled contents should look in CityGML. A popular modelling guide created by the Special Interest Group 3D (SIG 3D) is in use as a basis for CityGML 3D building data in many countries in the EU.

### 2.4.2 Georeferencing

This section concerns the storage and writing of georeferenced data in valid IFC and CityGML files. As noted before in section 2.1.2 CityGML files need to be georeferenced in accordance with their schema definition in order to be considered 'valid'. Thus, it is important that data in the IFC file is interpreted correctly by the conversion tool and used in further assigning CityGML coordinates. IFC files have different possibilities for storing georeferencies depending on the version. Georeferencing features in IFC files do not always produce desired results that can be read and interpreted in an intended way by software. For a valid IFC file the most important/key entities that define these georeferencing attributes are *IfcProject* and if not defined *IfcSite*. In addition unique buildings can be referenced and given related place and address attributes.

**Table 2.1** Georeferencing features by IFC version. Compiled from the IFC4 Documentation.

| | IFC 2x3 TC1 | IFC 4 ADD2 | IFC georeferencing element |
|---|---|---|---|
| Address and other owner attributes | X | X | IFCPOSTALADDRESS |
| EPSG 4326 | X | X | IFCSITE |
| Local coordinates | X | X | IFCLOCALPLACEMENT |
| Other EPSG 4326 based CRS | X | X | IFCGEOMETRICREPRESENTATION CONTEXT |
| Custom EPSG | | X | IFCPROJECTEDCRS |
| Custom CRS Distortion | | X | IFCMAPCONVERSION |

For IFC2x3 this means that the requirement for georeferencing a local coordinate system is writing the wanted georeference x,y and z (or E,N and H) coordinate values into *IfcSite* after an approximate location definition point that is in EPSG 4326. The interpretation of IFC georeferencing attributes is not always supported in an intended way by software. For this reason it could be beneficial to assign values to all ‘extra’ georeferencing features provided by the IFC. According to the IFC2x3 specification coordinates can be provided for *IfcProject* as EPSG:4326 offset (origin in EPSG4326) and true north rotation (*IfcDirection*). Additionally, if *IfcSite* is used then a local (own) coordinate projection system can be georeferenced (point value in EPSG:4326). However, there is no implicit connection between the locations given in *IfcSite*. In addition any entities that can be referenced in IFC like *IfcSite* or *IfcBuilding* can also contain address data written into a separate instance (IfcGeorefChecker – Documentation).

IFC4 contains all the previous georeferencing possibilities or levels and in addition the possibility to assign Coordinate Reference System (CRS) offset values for the whole IFC file using *IfcMapConversion.* In this instance it is possible to define new properties for each X and Y direction as vectors and to use the 'Scale' value to indicate distance distortions. The values found in *IFCMapConversion* are then used as a source coordinate system definition and can be linked to the *IfcProject* entity by *IfcGeometricRepresentationContext* that has two new definable properties in IFC4, called *SourceCRS* and *TargetCRS*. The CRS values need to be linked to this instance by *IfcCoordinateOperation*. The *TargetCRS* can be stored within an *IfcProjectedCRS* instance and it is recommended in the IFC4 schema that values in *TargetCRS* systems are defined with an EPSG-code within *IfcCoordinateRefenceSystem* (IfcGeorefChecker – Documentation).

The Coordinate Reference System (CRS) attribute for a CityGML file is contained within an *gml:Envelope* tag. However, this attribute is independent of the actual coordinate location written into the file as an envelope. Changing either the CRS EPSG-code or the assigned envelope limits (*gml:lowerCorner, gml:upperCorner*) will affect on how the CityGML georeference is interpreted by software. As a consequence the envelope limits and assigned CRS must both be assigned correctly for a successful georeference in CityGML (OGC 2012).

### 2.4.3 Methods of evaluation

The task of evaluating conversion accuracies and performance means first of all that software used to convert IFC to CityGML has to produce consistent results from the IFC files. Because CityGML uses LODs to structure its contents, it is important to establish requirements for each LOD where converted information is considered 'good enough' to pass requirements for all desired applications (if absolute accuracy cannot be derived).

This section presents methods from the literature review that could be applied to evaluating the result of IFC to CityGML conversions. CityGML files have to pass the requirements imposed by the XML schema and are validated with XSD files. Usually this can be done by using an offline or online validation tool. Most tools accept CityGML files with a .gml or .xml file extension. When a CityGML file has been deemed valid by the XSD validation process, it is considered a valid XML file with the right formatting matching the CityGML schema. However, this XML validation process does not tell the file creator and/or user what geometric, topological or important semantic properties are saved in the CityGML presentation. CityGML applications (analysis) require a certain degree of data integrity and can be used in validating CityGML file content properties. Thus, an application can be used as a type of validation method and shortcut when data reuse is limited for CityGML file use cases.

CityGML files that have passed the XSD validation process can further be inspected by using manual or automated methods. The manual method of evaluating the conversion results consists of inspecting multiple aspects of the converted data between the IFC and CityGML formats. Ideally a 'perfect' or full data conversion to LOD4 in CityGML should contain as many entities, relations, or elements as the IFC file that was read in (or witch ever LOD definition is used for the most detailed data 'layer' ). Due to the scale of detailed building models disjoined and/or malformed geometries can be hard to notice when inspecting CityGML files manually. The automated methods consist of programs that can check and repair CityGML file contents. Automated methods can save time and correct mistakes that are a challenge to notice by manual inspection and are likely to improve in the future. Challenges remain in the use and correct implementation when it comes to automatically detecting and fixing IFC to CityGML conversion results. Therefore, this kind of inspection or CityGML validation method is mainly not used in the practical part of this study and is only shown as an example (Figure 3.1).

#### 2.4.4 Related work

Sun et al. (2019) compares accuracies of building models derived from Airborne Laser Scanning (ALS) and 2D footprints from total station measurements to those of BIM. They discovered that the relative absolute accuracy between the two methods was on the scale of decimetres for test sites in the study. Common modelling guidelines are created as a base for CityGML models compared in the study.

Tran et al. (2019) introduce a method for comparing modelled indoor data that is referenced. The transformed indoor data quality is evaluated in terms of three data transformation metrics. One, accuracy (disjoint geometries) by comparing sampled points from the model with median absolute orthogonal distance from the surface plane and setting a cut-off distance for 'good enough' accuracy. Two, completeness ( missing number of elements) by comparing the amount of area that overlaps between the reference and source model element buffers. Three, correctness (extra elements) by calculating the overlapping area of the source model within the reference model buffer using wall thickness as minimum distance.

Giovanella et al. (2019) presents a concept of topological consistency and discusses the conformance of CityGML to ISO 19107. They state that currently making analyses based on topological queries is challenging because of topological inconsistencies in 3D data. This means that the geometry and topological model for example in CityGML contradict each other. Therefore, executing queries based on incidence graphs only is not possible. Topological inconsistencies occur most often in a case where a line segment intersects two polygons. CityGML as a standard is not ISO19107 compliant because it is possible to model valid geometries which contain topological inconsistencies.

## 2.5 Background for GeoBIM benchmark

The GeoBIM benchmarks consists of four separate tasks (GeoBIM benchmark 2019). Task four in the GeoBIM benchmark records hardware specifications for data conversions from IFC to CityGML and/or from CityGML to IFC. In the benchmark there are eight different test data files supplied all together. The choice of LOD is free in the conversions. The data submitted for task four includes recorded conversion time estimate and the converted data upload together with details for executing the conversion. The submissions are classified into different types by software main functionality.

# 3 Conversion quality evaluation

The conversion quality analysis is done by using a series of methods to create quality metrics. The methods presented in this section aim to create key metrics for evaluating the conversion quality. Additionally, there are other ways to assess the resulting conversion output.

For example City Geography Markup Language (CityGML) file contents can also be verified for specific purposes by simply testing the CityGML file as analysis input. Any requirements for data coherency and harmonisation for the analysis have to be listed. Resultant layers from analysed data can reveal gaps in the input CityGML dataset. Analyses that depend on intact Three-dimensional (3D) object surface data and well defined distances are most useful to quickly evaluate whether a CityGML dataset is ‘good enough’ to be taken advantage of in another similar application.

The accuracy of an Industry Foundation Classes (IFC) to CityGML transformation varies depending on the original data collection or creation method of the IFC file and the error introduced by the data conversion process. The resulting CityGML data file accuracy can be calculated as the sum of these two factors. However, in practise the IFC files often have preassigned measurement units that are rounded up to four decimals. Therefore, the only factor for the data conversion full accuracy is the error introduced in the data format conversion or translation process. The calculation of the data conversion accuracy estimate depends on the number of required coordinate system translation parameters. Thus, the data transformed from similar map projections has a smaller total conversion error.

Some errors in CityGML data can be detected automatically using geometry scanners or fixers. *Val3Dity* by Hugo Ledoux (Ledoux, 2013) is an example of a computer program that can detect errors in CityGML primitives (features) using a set of rules from International Standardization Organisation (ISO)19107. The geometric primitive evaluation tool does not take into account topology (separate or otherwise). However, if the extends of a feature can not be solved by *val3Dity* the result is a failed validation (Figure 3.1). While usage of an automated geometry validation or fixing tool for CityGML can potentially decrease the time for evaluating resulting CityGML files, the effects of applying such a tool can be unpredicted since often processing is being done on the CityGML geometries themselves in the validation process. The characteristics and possible signs of conversion quality are therefore lost in such an application of automated error detection.

```
Administrator: Command Prompt
Primitive(s) validated: All
(CityGML/CityJSON have all their 3D primitives validated)
Parameters used for validation:
   snap_tol                  0.001
   planarity_d2p_tol          0.01
   planarity_n_tol              20
   overlap_tol                none
[===================================================] 100%

+++++++++++++++++++ SUMMARY +++++++++++++++++++
Total # of Features:           1
# valid:                       0 (0.0%)
# invalid:                     1 (100.0%)
Types:
  Building
+++++
Total # of primitives:         1
# valid:                       0 (0.0%)
# invalid:                     1 (100.0%)
Types:
  Solid
+++++
Errors present:
  902 -- EMPTY_PRIMITIVE
          (1 primitives)
+++++++++++++++++++++++++++++++++++++++++++++++

Full validation report (in HTML format) saved to "/report\report.html"

E:\vps_conversion_notdel\val3dity-2.1.0-windows-x64\val3dity-2.1.0>
```
Figure 3.1: Val3Dity example.

## 3.1 Requirements of building data

Data requirements and use cases for converted data are important to accommodate the views of different user groups. Currently Level-of-Detail (LOD)3 data is mostly used for visualisation purposes. There are more planned use cases for LOD4, but converting IFC files to this LOD is much more resource and time consuming. The most notable applications for LOD4 include noise simulations and indoor navigation.

The requirements for conversion quality can be divided into four different classes based on the structural hierarchy presented in Figure 2.8.

**Table 3.1** Requirements organised into different classes by source.

| Class | Explanation |
|---|---|
| Requirements and definitions of data structure | In order for LOD3 and LOD4 data to match expectations for the LOD content definition suggestions in the CityGML version 2.0 specification, the distinction between LOD3 and LOD4 has to be specified. LOD3 includes only visible exterior surfaces of buildings. LOD4 also has the interiors modelled in CityGML. The requirements derived from LOD definitions will change depending on what kind of LOD convention is used to divide the CityGML contents. Figure 2.3 depicts is an example of such a convention. |
| General converted data expectations | The general requirements for converted data are:<br>1. Conformance to CityGML schema<br>2. Error free geometry with suitable accuracy tolerances<br>3. Semantics at an acceptable level-of-detail |
| Data requirements for specific applications | The usage of CityGML files in LOD3 and LOD4 can have a set of requirements that are reflected in the file data contents. Usually these requirements are set by the national practitioners unless an official guideline or law can be applied (Figure 2.8). |
| Requirements not meeting general expectations | Applications have specific requirements in align with requirements set by national authorities or 3D modelling practitioners. However, some applications might not require schema conformance, well defined semantics or complete geometry and only require parts of the converted data. |

The GeoBIM benchmark task four does not specify any requirements for converted data, so the focus is on creating CityGML contents that passes the XML Schema Definition (XSD) validation process and therefore conforms to the CityGML version 2.0 schema. An exception from this in the case study is the Special Interest Group 3D (SIG 3D) modelling guide that is followed for the test data in Myran.ifc. The LOD definitions follow the recommendations set forth in the CityGML version 2.0 specification. Other requirements for converted data such as geometry and semantics from class ‘General converted data expectations’, are in control of the conversion tool deployed in converting IFC data to the CityGML format.

## 3.2 Quality metrics

The requirements observed in 2.4.1 as part of the in-depth literature study together with identified quality indicators are used as a basis for creating key metrics to evaluate converted IFC data. The creation process is depicted as Figure 3.2.

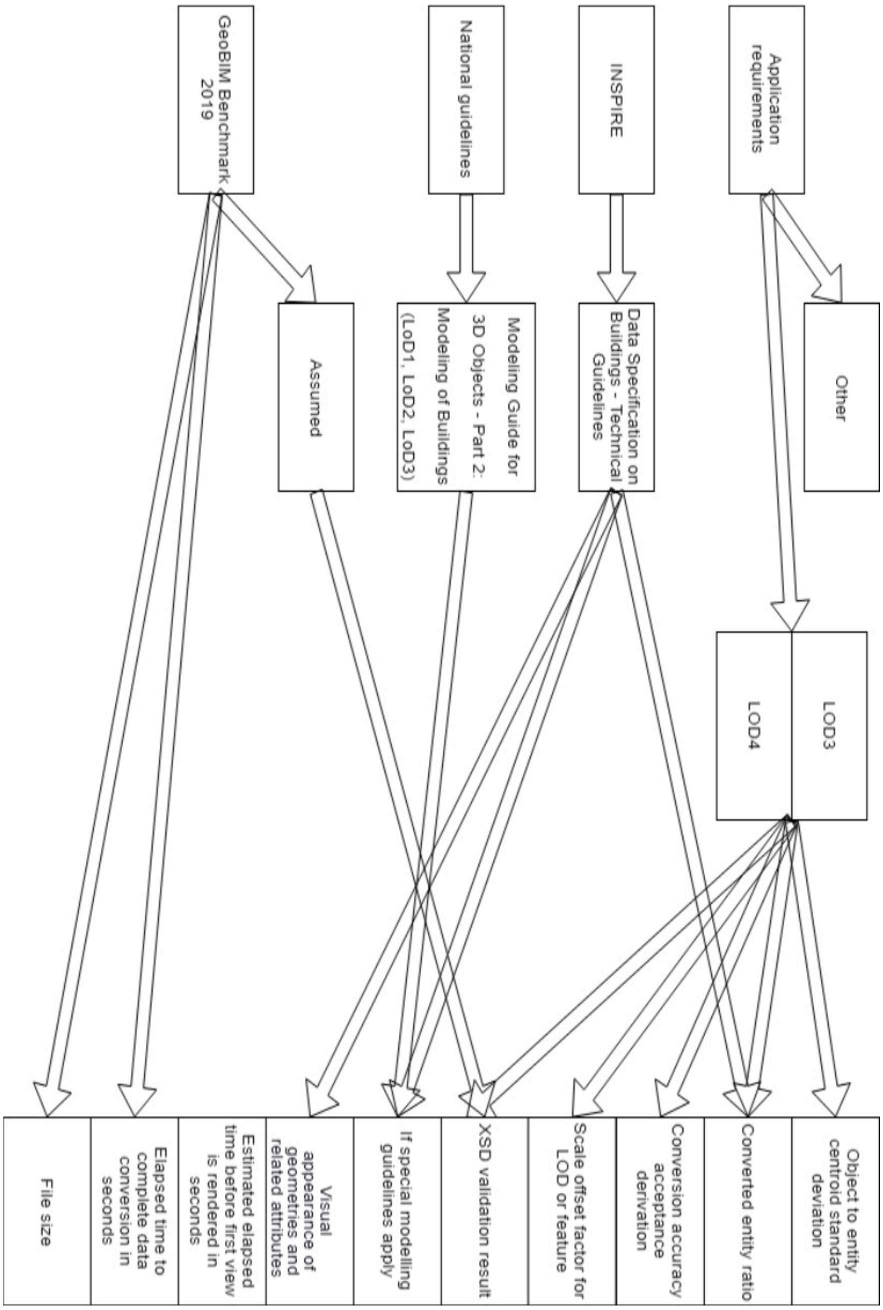



Figure 3.2: Metrics derived from data requirements.

Evaluation of converted data is usually characterised by quality parameters like completeness, accuracy and correctness. Methods for deriving these attributes vary between different studies. The relations of the resulting key metrics to these attributes and methods are shown in Table 3.2.

**Table 3.2** Related metrics, methods and evaluation attributes for data conversions.

| Metric | Method | Quality parameter |
|---|---|---|
| Converted entity ratio | Entity conversion ratio | Completeness, Correctness |
| Object to entity centroid standard deviation | Geometric processing consistency | Accuracy |
| Conversion accuracy acceptance derivation | Estimating transformation positional accuracy | Accuracy |
| Scale offset factor for LOD or feature | Unit scale validation | Accuracy |
| XSD validation result | XSD validation | Consistency |
| If special modelling guidelines apply | Conformance to modelling guidelines | Consistency |
| Visual appearance of geometries and related attributes | Visual inspection | Performance |
| Estimated elapsed time before first view is rendered in seconds | Time elapsed for the first rendered view in seconds | Performance |
| Elapsed time to complete data conversion in seconds | Elapsed conversion time in seconds | Performance |
| File size | Recording file size | Correctness, Performance |

## 3.3 Methods for creating quality metrics

The methods to be deployed in the case study are presented in Table 3.3. Each of the methods corresponds to a quality metric based on the literature review and GeoBIM benchmark task four. As listed in Table 3.2.

**Table 3.3** Methods for creating test quality metrics.

| Method | Description |
|---|---|
| Entity conversion ratio | This method counts how many entities end up being converted to CityGML. The actual mappings and conversion methods between software solutions can vary, and therefore the produced key metric is a ratio of read in and write out. |

| Method | Description |
| --- | --- |
| Geometric processing consistency | The accuracy or consistency of geometric processing is to be tested by calculating or extracting identifiable object centroids and comparing the IFC entity to its CityGML counterpart. The resulting key metric from this test is the standard deviation of coordinates between the IFC entity and CityGML object. |
| Estimating transformation positional accuracy | The accuracy of the data transformation is estimated by calculating the standard deviation of the used transformation. For all models using metric system units the transformation needed is unitary. The accuracy of the applied georeference is found out by comparing the assigned map projection (used in the CityGML file) coordinates to those in the original IFC model. |
| Unit scale validation | An object that has measurement attributes is evaluated against its units by calculating a distance matching its known object measurements. |
| XSD validation | eXtensible Markup Language (XML) file structure can be verified by checking its contents against an XSD file. The purpose of an XSD file much like Document Type Definition (DTD) is to confirm that an XML file paired with it follows the given instructions. However, XSD files use XML schemas that can define custom data types instead of document type definitions. The advantage of using XSD instead of DTD with CityGML is that elements in XML schemas can contain enumerated values. This allows for example the validation of code list contents from CityGML. The CityGML file is validated with an online CityGML schema validator at http://geovalidation.bk.tudelft.nl/schemaCityGML/ . |

| Method | Description |
|---|---|
| Conformance to modelling guidelines | This method creates a CityGML attribute for a guideline. For example, the attribute is assigned a string value ‘SIG 3D’. The metric produced by this method is an attribute indicator to tell the user if set instructions have been influencing the data conversion process. If a conversion result matches expectations from a certain modelling style or guide, this attribute is created into the CityGML file main building component. |
| Visual inspection | CityGML files are compared visually in this method. The strengths of this type of test are instantly obvious when it comes to CityGML file contents. Known problematic or difficult IFC structures are fairly easy to confirm ‘good enough’ for viewing or presenting to audiences. The downside of this method is that opening and rendering large CityGML files takes time and it becomes hard to inspect the files for overlapping geometries or faulty semantics. |
| Time elapsed for the first rendered view in seconds | The user hardware configuration is documented and the active working time (GPU under load in seconds) is recorded to indicate geometric complexity when rendering a first view in CityGML. |
| Elapsed conversion time in seconds | This method is self explanatory and it measures time elapsed from conversion start to finish. |
| Recording file size | The recorded file size from the CityGML file. |

Positional accuracy of IFC to CityGML conversions is evaluated through coordinate dimensions, measurements and unit scaling. Data that is read in from the IFC file as constructed entities will keep its form and dimensions if converted correctly. Units used by the IFC file should scale to assigned Coordinate system in CityGML.

# 4 Case study

This case study describes the execution of test methods to create quality metrics for converted data.

The study is composed of ten methods in total. The methods used in this case study are presented in Table 3.3. The creation of the quality metrics is explained with the selected conversion tools from section 4.1 to 4.4.

## 4.1 Selection of conversion tools

The selection of conversion tools  for the case study is based on availability of software and City Geography Markup Language (CityGML) writers that are able to write out eXtensible Markup Language (XML) conforming to the  CityGML schema.

The tools in use within this thesis for converting Industry Foundation Classes (IFC) files to CityGML are:

- FME by Save Software
- ESRI ArcGIS PRO Data Interoperability extension

Both of these tools use the FME translation engine and transformers.

## 4.2 Overview of the test

The testing of quality metrics in this case study is done by implementing the methods in Table 3.3.

In this section the workflows applied to convert IFC to CityGML are explained shortly. More details about the workflows are available in (Appendix A). The studied data is retrieved from GeoBIM benchmark 2019. The descriptions of the data are available in section 4.3 and in more detail at https://3D.bk.tudelft.nl/projects/geobim-benchmark/data.html .

This study uses two software configurations with multiple conversions in the form of FME originated workflows and converts the IFC files into CityGML data. The different workflow configurations are recorded in Table 4.1 below.

**Table 4.1** FME and ArcGIS Data Interoperability extensions workflows for converting IFC data to CityGML.

| Software | FME 2017 | FME 2017 | FME 2017 | FME 2017 |
|---|---|---|---|---|
| IFC data | Specific IFC (2x3) geometries | Specific IFC 4 geometries | Myran | Up:Town |
| LOD 3, LOD 4 | 4 | 4 | 3, 4 | 4 |
| Implementation | Workbench | Workbench | Workbench | Workbench |
| Workbench version | 2017.0 | 2017.0 | 2017.0 | 2017.0 |
| Additional information | | | | |

| Software | ArcGIS Pro Data Interoperability extension | ArcGIS Pro Data Interoperability extension | ArcGIS Pro Data Interoperability extension | ArcGIS Pro Data Interoperabilit y extension |
|---|---|---|---|---|
| IFC data | Specific IFC (2x3) geometries | Specific IFC 4 geometries | Myran | Up:Town |
| LOD 3, LOD 4 | 4 | 4 | 3, 4 | 4 |
| Implementation | ArcGIS Pro | ArcGIS Pro | ArcGIS Pro | ArcGIS Pro |
| Workbench version | 2018.1 | 2018.1 | 2018.1 | 2018.1 |
| Additional information | Quick import and export LOD 4 | Quick import and export LOD 4 | Quick import and export LOD 4 | Quick import and export LOD 4 |

The conversion workflows in FME workbench are executed with a FME IFC or Revit reader. The feature properties are set to be handled together with geometry in the reader settings. Each reader maps the IFC file entities to multiple FME feature types. The IFC entities are processed into CityGML data by simplifying their geometry types into solids with triangulated surfaces. Finally the mandatory CityGML attributes are created for each CityGML module and send to FME CityGML writer (Figure 4.6).

### 4.2.1 Conversion in Feature Manipulation Engine (FME)

Solids extracted from IFC are converted with the help of a triangulation transformer and then combined into meshes. External shell geometries from the IFC file are transformed into Boundary representation (B-Rep). Geometry installations can be formed using the two aforementioned methodologies in the FME 2017 conversion workflows. Attributes and semantics are imported using FME attribute reader (manager) into FME features and written into CityGML format by FME CityGML writer.

The data conversion from IFC to CityGML in the FME workbench uses a workflow executed in 3 different parts:

1. Read in IFC file
2. Carry out data transformation to FME file storage format
3. Write CityGML from the FME Feature Storage (FFS)

These three parts are in the creation of the data conversion workflow further divided into eight steps:

1. Create IFC or Revit reader
2. Create CityGML writer
3. Connect necessary inspectors for identifying data contents (entities) from IFC feature groups
4. Filter out unwanted features
5. Connect reader to transformers
6. Inspect transformer outputs
7. Connect transformer outputs to CityGML writer
8. Save and run workflow

#### 4.2.2 Conversion in ArcGIS PRO Data Interoperability extension

Conversions in ArcGIS PRO Data Interoperability extension work in principle the same way as they do in FME 2017 workbench (when they are not converted using the quick import and export options in ArcGIS PRO Catalog workspace). The workbench version used for the ArcGIS PRO extension is 2018.1.

The ArcGIS PRO Data Interoperability extension enables options for quick importing and exporting of data formats to and from the ESRI geodatabase. Options for importing IFC files and writing CityGML are nearly identical to those in the FME 2017 workbench (Figure 4.6). The Interoperability extension quick import and export has three possible outcomes that affect how complete the data output is from the IFC file or ESRI geodatabase:
1. Green check mark, import and export completed without errors
2. Yellow triangle warning sign, import and export completed but some features could not be converted or other translation log errors occurred
3. Red cross, import and export terminated and all features could not be completed or written out

Every outcome from the quick import and export option in ‘Catalog’ workspace creates an output file. An exception to this is the instance where the data contents can not be saved to the desired output format by the writer. In this case none of the IFC features are saved to the ESRI geodatabase or CityGML format.

### 4.3 Test data

Data used in this thesis is acquired from the GeoBIM benchmark project. In the case study four IFC files are converted to evaluate created quality metrics. The details of this project and data descriptions can be read from https://3D.bk.tudelft.nl/projects/GeoBIM-benchmark/ . The case study results for different Industry Foundation Classes (IFC) models from Table 4.2 are visible in more detail in Appendix (B).

**Table 4.2** Table of included IFC data.

| Model | Specific IFC (2x3) geometries | Specific IFC 4 geometries | Myran | Up:Town |
|---|---|---|---|---|
| IFC version | 2x3 | 4 | 2x3 | 2x3 |
| Location | - | - | Falun, Sweden | Rotterdam, the Netherlands |
| Exported from | IfcOpenShell 0.6 | IfcOpenShell 0.6 | Autodesk Revit 2018 (ENU) | Autodesk Revit 2015 (ENU) |
| Provided by | T. Krijnen | T. Krijnen | MONDO arkitekter | Municipality of Rotterdam |
| Georeferenced according to the GeoBIM benchmark requirements | Yes | Yes | No | No |

## 4.4 Implementing the methods

The methods from Table 3.3 are implemented as tests in the FME 2017 and ArcGIS PRO Data Interoperability extension. These tests are designed to produce a group of ten varying values as metrics (Figure 3.2). The tests asses varying aspects of features like georeference, modelling guideline conformance and data complexity.

The creation of the quality metrics in Table 3.2 and executing the devised methods can be done by parsing the IFC and CityGML files. However, in reality software like FME already offer tools for this kind of data processing. Manual processing of the XML structure is only useful in cases where such software is unavailable.

In the method 'Geometric processing consistency' an IFC entity and a CityGML object are identified from a data conversion with the centroid extractor transformer (Figure 4.1) or with the 'place marker' option in FME data inspector.

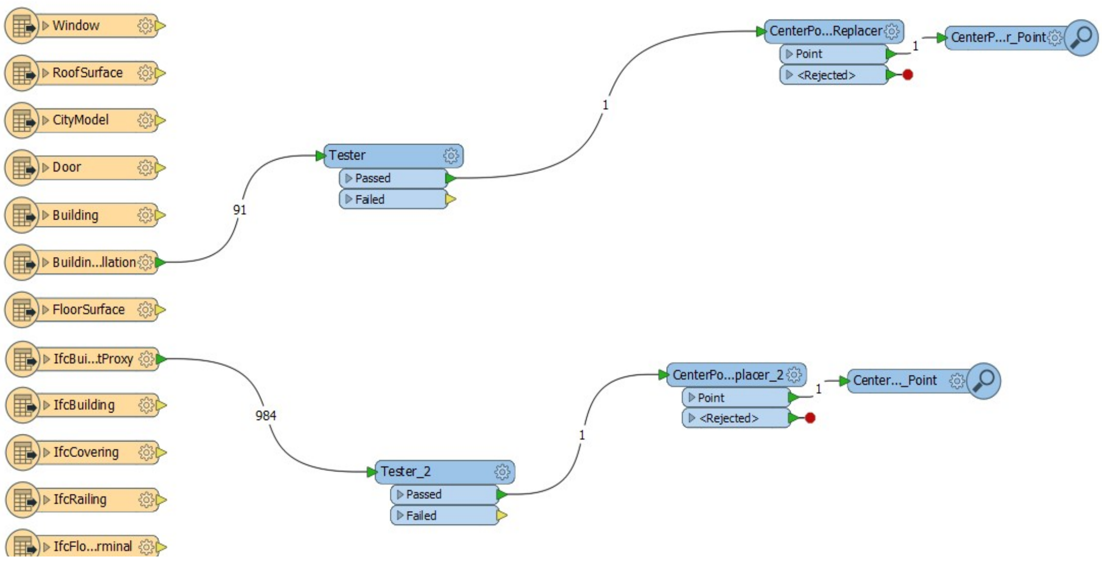



Figure 4.1 Transformation workflow example using *CenterPoint_Replacer.*

The standard deviation between the IFC and CityGML objects is calculated using these centroid locations. For this reason it is important to know what type of transformation is used when converting the data. The conversion units can vary and scale in conversion workflows. In the case of  the IFC Myran model the correct units are in millimetres while the IFC Up:Town model units are in metres.

The number of read in IFC entities and output CityGML objects are counted and expressed as a ratio. The method can be performed by inspecting each file individually in FME data inspector or by saving and looking up the entity and object counts from the FME translation log file (Figure 4.2).

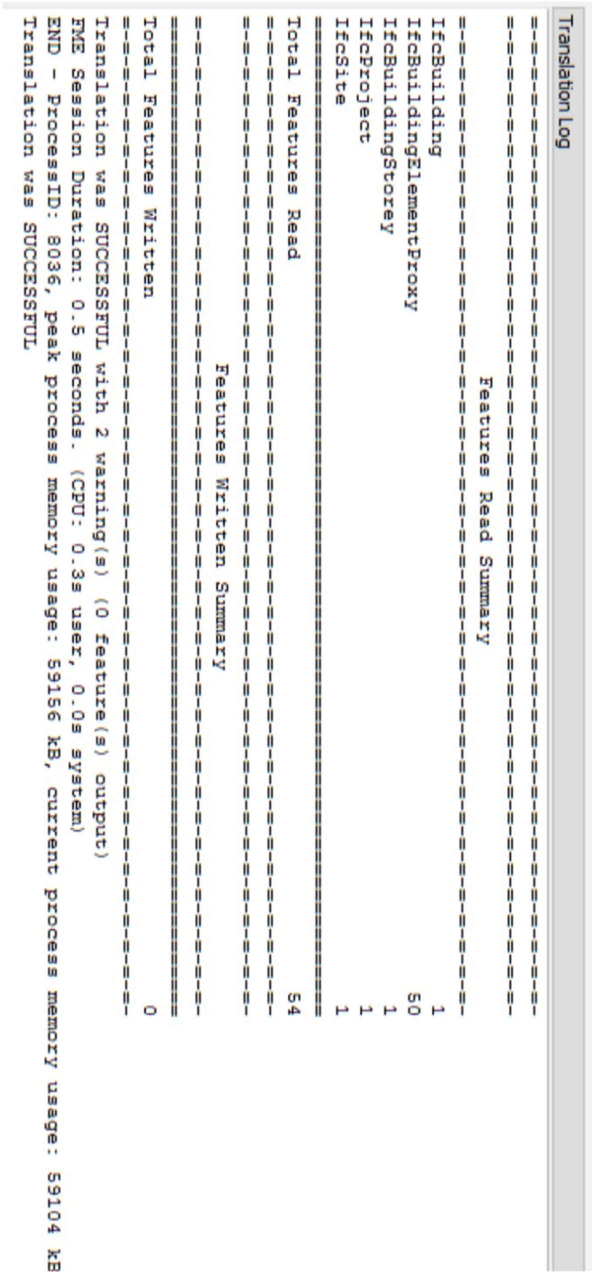



Figure 4.2: The FME translation log.

In the ‘Elapsed time for the data conversion in seconds’ method, the elapsed conversion time is looked up from the FME translation log after a successful data conversion (Figure 4.2).

The ‘Conformance to modelling guidelines’ method is executed by creating an attribute for the CityGML file either during of after the conversion. This attribute is agreed upon at the main CityGML module object level (Figure 4.3). If no such attribute exists, the converted data is not known to conform to any extra modelling instructions or guidelines.

```
</bridge:lod3MultiSurface>
</bridge:RoofSurface>
</core:cityObjectMember>
<core:cityObjectMember>
<bldg:Building gml:id="gml_722cd6ae-5529-44ac-971d-9844b08559ad">
<gml:name>myran_building</gml:name>
<gen:stringAttribute name="modelling_instructions">
<gen:value>SIG3D_Modeling Guide for 3D Objects - Part 2</gen:value>
</gen:stringAttribute>
```

Figure 4.3: Adding attribute for modelling guideline after conversion in Notepad++.

The ‘Unit scale validation’ method is executed inside the FME data inspection tool (since this opens automatically after a completed data conversion with default settings). The measure tool is used to confirm that correct units are processed in the conversion. A CityGML object with a known measurement like width or height is evaluated against the measurement tool or object coordinates (Figure 4.4).

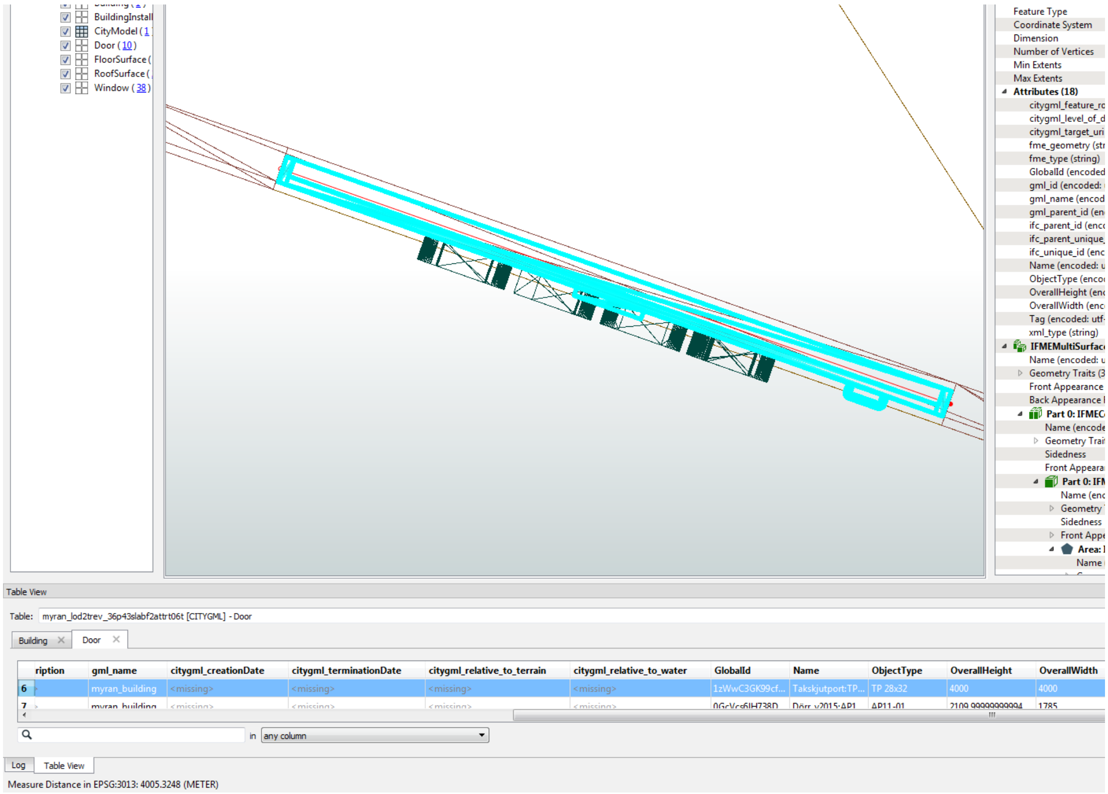



Figure 4.4: FME data inspector measurement tool.

The outcome of the data conversion is evaluated in the FME data inspector (Figure 4.5) with the ‘Visual inspection’ method.

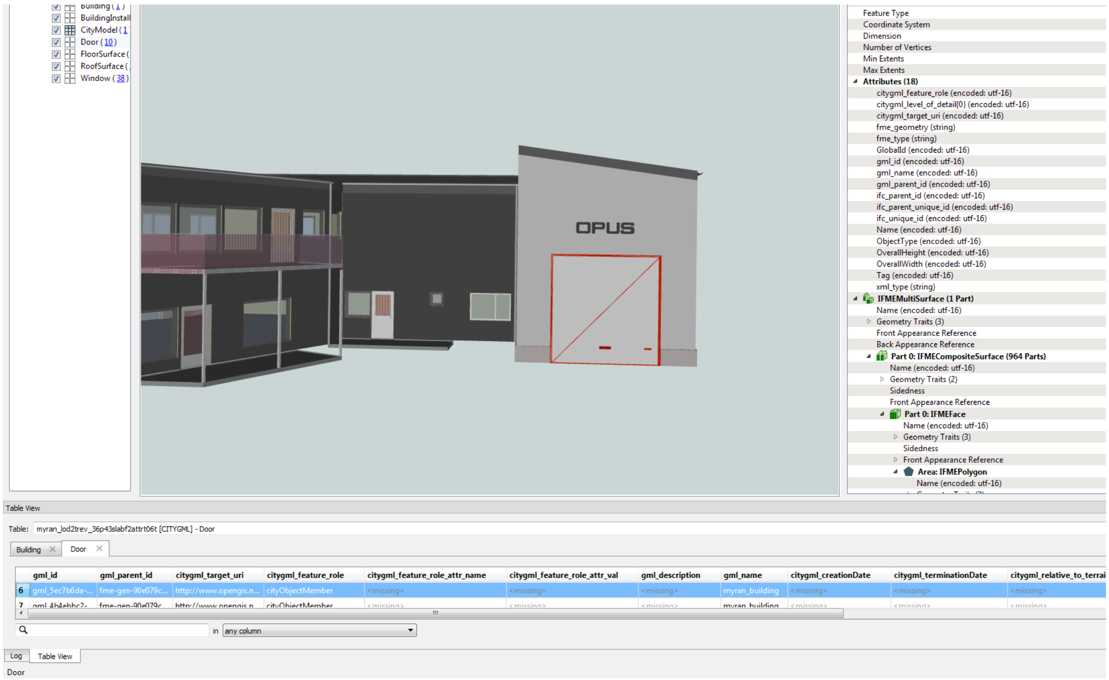


Figure 4.5: FME data inspector.

The elapsed time for the first Three-dimensional (3D) render view is timed manually. With the 'Elapsed time for the first 3D render view' method.

The 'XSD validation' method is executed by uploading the CityGML files to the online XML Schema Definition (XSD) validator at http://geovalidation.bk.tudelft.nl/schemaCityGML/ or by using the FME CityGML writer validation option (Figure 4.6).

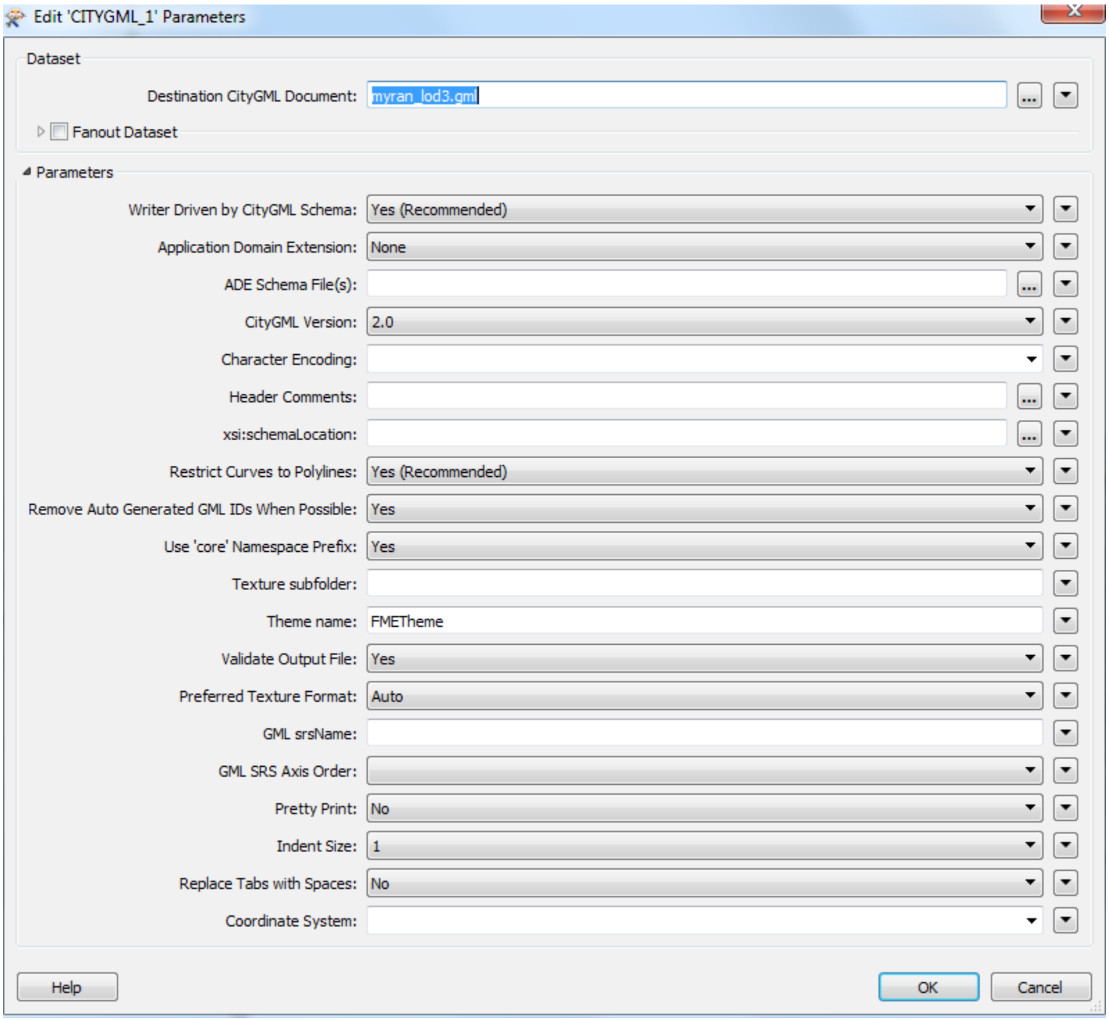


Figure 4.6: FME CityGML writer options.

The 'Estimating transformation positional accuracy' method is executed by inspecting the known envelope definitions for IFC and CityGML coordinate reference systems. The Coordinate Reference System (CRS) definition is given as an EPSG-code. The transformation accuracy estimation is calculated by using the defined envelope centroids as sampling points and calculating the standard deviation of the transformation inside the conversion. The conversions in the benchmark are executed using a unitary transformation (1). During the transformation the XY-planar coordinates and the known height in the IFC model are handled separately (Figure 4.7).

(1) The new projected CityGML coordinates (E and N) are transformed from the current Coordinate Reference System (CRS) with the unitary transformation equation. In the transformation $E_0$ and $N_0$ note the translation and $\alpha$ the rotation in the rotation matrix. The $x$ and $y$ are the existing coordinates in the engineering (IFC) system.

$$\begin{bmatrix} E \\ N \end{bmatrix} = \begin{bmatrix} E_0 \\ N_0 \end{bmatrix} + \begin{bmatrix} \cos\ \alpha & -\sin\ \alpha \\ \sin\ \alpha & \cos\ \alpha \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \tag{1}$$

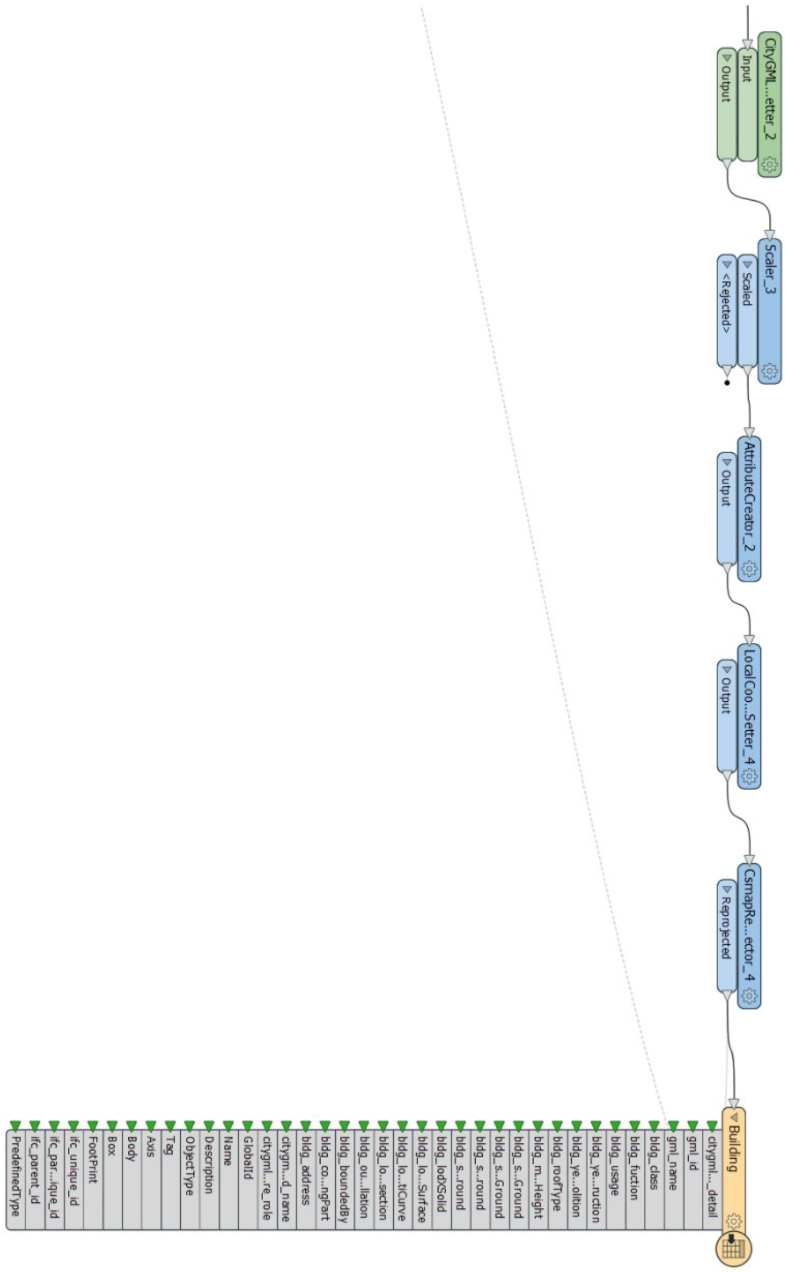



Figure 4.7 The transformation configured in FME workbench.

The ‘File Size’ method is executed by inspecting the original IFC and the resulting converted CityGML files in a file system.

## 4.5 Interpretation of the quality metrics

The evaluation of quality metrics created with the methods from Table 3.3 is explained in Table 4.3. This is to avoid possible confusion or misconceptions in what the metrics should be compared against. Also many of the interpretations of quality in the context of Industry Foundation Classes (IFC) to City Geography Markup Language (CityGML) are not self explanatory.

**Table 4.3** Quality metric evaluation explanations.

| Metric | Explanation | More |
|---|---|---|
| Converted entity ratio | The converted entity to object ratio measures how many of the read IFC entities were converted to CityGML objects during the data conversion. This implies the completeness and correctness of the data conversion. | Take care when assessing the amount of converted IFC entities in relation to CityGML object output. The IFC file can contain other entities besides building data while on the other hand a group of CityGML objects in FME sometimes only counts as a single object. |
| Object to entity centroid standard deviation | The difference between an IFC entity and the corresponding CityGML object is identified. The difference is quantified as standard deviation. If standard deviation is larger than that of the point accuracy processing in the CRS envelope, the feature is likely to be in a wrong location. | This metric is based on the assumption that a feature in the IFC and CityGML files is drawn into 3D using a similar canvas or coordinate approach. The relation of the identified comparison feature to the origin of the system should be retained in the data conversion if geometry is not changed. |

| Metric | Explanation | More |
|---|---|---|
| Conversion accuracy acceptance derivation | The conversion accuracy or georeference is evaluated with the data coordinate transformation standard deviation. Ideally the error introduced by the data transformation is insignificant. | An insignificant error in the data transformations means that the change in position when both files have been georeferenced is so small that it does not change the georeferencing coordinates. The introduced error is depended on the complexity of the data coordinate transformation. In the case study only one pair of sample points is used. |
| Scale offset factor for LOD or feature | The scale of the data conversion geometry is tested by using a measurement tool or comparing coordinates of a feature. A model that has its units given in mm should have a measurement accuracy corresponding to 1/10000 of a metre. | The attributes copied over from IFC files do not change together with the geometric scaling. Therefore, using an object with specified length and or height attribute makes this comparison faster when inspecting converted data content. |
| XSD validation result | A valid CityGML file that has passed XSD validation conforms to the CityGML XML schema definition. | Due to the Tudélft schema validator file size limitation the XSD validator was changed to the corresponding FME functionality. However, the Tudélft validator results were also confirmed within FME. |
| If special modelling guidelines apply | The conformance to a guideline is marked with an added attribute to the first hierarchical (building) CityGML module features. The presence of this attribute means that the CityGML content is converted/modelled after these guidelines. | Although the attribute for guideline conformance can be created during or after the conversion, a data inspection is often needed to confirm actual guideline conformance. |

## 4.6 Result of the metrics in the case study

The results of the case study are constructed in four parts – one part for each IFC file (Table 4.2). The quality metrics 'Visual appearance of geometries and related attributes', 'Scale offset factor for LOD or feature' and 'If special modelling guidelines apply' are producing boolean products but the evaluation process is slightly different for each IFC file and reader.

The results corresponding to the IFC test data model 'Specific IFC (2x3) geometries' are presented in Table 4.4. The first row in the table contains the reader used for data mapping in the conversion software. The example results reveal that one CityGML object (49 instead of 50 in 'Converted entity ratio') is missing from three out of six conversions. As a consequence the 'Visual appearance of geometries and related attributes' metric is 'Fail'.

**Table 4.4** Results part one of four (Specific IFC (2x3) geometries).

| Reader | IFC | IFC OLD | RVT |
|---|---|---|---|
| Conversion Software | FME 2017 Workbench | ArcGIS Pro Data Interoperability extension (trial) | FME 2017 Workbench |
| Converted entity ratio | 50/64 | 50/73 | 50/64 |
| Object to entity centroid standard deviation | Xy=0, h=0 | Xy=0, h=0 | Xy=0, h=0 |
| Conversion accuracy acceptance derivation | 0.0085 | 8.4853 | 0.0085 |
| Scale offset factor for LOD or feature | Pass | Pass | Pass |
| XSD validation result | Pass | Pass | Pass |
| If special modelling guidelines apply | No | No | No |
| Visual appearance of geometries and related attributes | Pass | Pass | Pass |
| Estimated elapsed time before first view is rendered in seconds | 1.8s | 0.1s | 1.8s |
| Elapsed time to complete data conversion in seconds | 2.6s | 6.7s | 8.4s |
| File size | IFC 31kt / CityGML 5872kt | IFC 31kt / CityGML 3525kt | IFC 31kt / CityGML 5866kt |

| Reader | RTV | IFC | RVT |
|---|---|---|---|
| Conversion Software | ArcGIS Pro Data Interoperability extension (trial) | ArcGIS Pro Data Interoperability extension (trial) FME 2018.1 Workbench | ArcGIS Pro Data Interoperability extension (trial) FME 2018.1 Workbench |
| Converted entity ratio | 49/64 | 49/64 | 49/64 |
| Object to entity centroid standard deviation | Xy=0, h=0 | Xy=0, h=0 | Xy=0, h=0 |
| Conversion accuracy acceptance derivation | 8.4853 | 0.0085 | 0.0085 |
| Scale offset factor for LOD or feature | Pass | Pass | Pass |
| XSD validation result | Pass | Pass | Pass |
| If special modelling guidelines apply | No | No | No |
| Visual appearance of geometries and related attributes | Fail | Fail | Fail |
| Estimated elapsed time before first view is rendered in seconds | 1.0s | 1.8s | 1.0s |
| Elapsed time to complete data conversion in seconds | 12.6s | 2.9s | 9.6s |
| File size | IFC 31kt / CityGML 3625kt | IFC 31kt / CityGML 6366kt | IFC 31kt / CityGML 6360kt |

The FME conversion tools allow for three types of readers to be used: IFC, Revit and the backwards compatibility mode for old IFC workflows. The different workflow implementations (Table 4.1) have their results listed with the old IFC reader if the updated version could not produce a valid conversion result in the case study. The sample geometries with IFC2x3 and IFC4 have therefore been tested with this reader and can be seen in the results as an increase in the entity count on read in entities.

The results from the case study parts one and two reveal that 'IfcBooleanClippingResult_1' geometry sample is missing from the conversion results when the workflows are implemented in the ArcGIS Pro Data Interoperability extension. Also results for workflows using the quick import and export functions of this extension indicate a different georeferencing accuracy which is probably due to expected geolocations in the IFC files (EPSG 4326). The other workflows executed in the workbenches of FME are assigning EPSG 3013.

The results from part three indicate unexpected visual results for the ArcGIS Pro Data Interoperability extension workflows. The resulting CityGML file for quick import and export using 'Revit reader' did not produce valid schema validation results for CityGML file from the 'Myran.ifc' file. This is due to the 'Revit reader' exposing extra property sets from the IFC file that are not included in the normal data mappings. In the extension these property sets get set on the CityGML file even if they contain no values.

The 4th IFC model in the case study named Up:Town could not be assessed by all quality metrics because it was too much for the used hardware to handle. However, the workflow provided by FME as a conversion example worked only on the Up:Town IFC model.

As a whole it can be said that the applying of methods worked well and different IFC conversion products did indicate the completeness, correctness, accuracy, consistency and performance when tested on different groups of IFC files.

Detecting the actual missing entities from the conversions is challenging because only the upper level of hierarchical geometries is visible in data inspections. Thus, entities converted into CityGML 2.0 can show a group of objects as one object. For the counting of converted features only the lowest level of hierarchical content is preferred.

# 5 Discussion

Current workflows in the case study for Industry Foundation Classes (IFC) to City Geography Markup Language (CityGML) conversions rely heavily on FME CityGML writer (Appendix C). Thus, the evaluation of truly independent conversion results is probably not possible – at least not in the scope of this master's thesis. The reasons for the missing geometries and semantics are explained in detail. The concept of what is considered 'a valid conversion method' in order to be included into this study is also discussed.

The methods for converting data from IFC to CityGML vary depending on available data mapping model or framework. Sometimes instead of a framework the term semantics is used even if these do not refer to attributes. The data mapping effects on how and in what order the IFC data is being converted to CityGML (Deng et al. 2016).

In FME there are two different readers or 'data mapping modes' where the IFC file contents is either mapped to multiple or to a single output group. Ultimately the selection of groups and mapped data also depend on the selected reader types.

Conversions can in theory be executed individually for each LOD or the data can be thinned down from a complete LOD4 conversion using the FME data mappings. The more familiar and proven concept of updating individual LODs, is used in the case study (Billen et al. 2014).

The conversion for Up:Town quick import and export diverges from the other conversion workflows because a hierarchical mode was applied in the IFC reader. This approach yielded to considerable 54333 converted IFC entities. The hierarchical reader is used because of unusual structures in the IFC file (FME website).

## 5.1 Requirements for CityGML

Because Three-dimensional (3D) City modelling efforts are currently mostly funded by urban planning related actors, the guidelines for building data modelling will be targeted towards the same audience. However, separately licensed versions of this data could become available for the consumer applications.

Although requirements of building data for CityGML LOD3 and LOD4 in phase one (of the overall research method) have multiple classes, the most challenging one to identify is 'Data requirements for specific applications'. The majority of CityGML use cases in urban planning and related domains today require only LOD1 and LOD2 data (Figure 2.8).

The main questions, that should be posed for the future of CityGML, are whether it will reach general acceptance outside the urban planning and related realms and should the data model perhaps include a separate class for application types? What is the future status of subset encodings like CityJSON? Should such encodings be treated as parts of the CityGML data model extensibility or as alternatives to it?

The application requirements for CityGML data are likely to be updated for a number of applications due to the ongoing work for the planned 3.0 version of CityGML (CityGML 3.0 Development).

## 5.2 Key quality metrics and researched methods

Methods 'Entity conversion ratio', 'XSD validation' and 'Visual inspection' are often used when evaluating CityGML data contents. However, the case study results reflect that unless the tools and methods used in the conversion process are well documented it is challenging to determine if all IFC entities have been converted successfully (Appendix C).

The method for creating the 'Converted entity ratio' is software dependent and requires a point of comparison from the same IFC file. The interpretation of this metric is challenging due to CityGML generalisation rules and different conversion frameworks (Biljecki 2017). Two entities read by the conversion software can convert to a multitude of CityGML objects. In a LOD4 conversion the number of converted entities is easier to control and entities modelled in the lowest hierarchical level can be reconstructed according to a near one-to-one equivalency (Deng et al. 2016; El-Mekawy et al. 2012b).

The quality metrics found also prove that it is possible to track the progression and quality of Three-dimensional (3D) building data in CityGML. Although due to varying conversion tools, visual file content inspection is still necessary at some point in the content validation process.

The methods introduced in Table 3.2 are tested on the converted data. Because of the narrow selection of conversion tools and CityGML 2.0 compatible writers, the resulting conversions are inspected within the same conversion environments. Therefore, the resulting quality indication metrics are biased in the sense that they expect any software claiming CityGML 2.0 compatibility to yield to the same inspection results. Thus, the software used to view the CityGML file and to convert it have to be in the same software suit.

The metric 'Estimated elapsed time before first view is rendered in seconds' is not normally included in the metrics but serves for benchmarking the user experience (Figure 3.2).

## 5.3 The test methods to compute the quality metrics

It can be concluded from the results that there are not many functional software options to select from for IFC to CityGML data conversions. With the tools available, it can be said, that from a data storage point of view LOD3 files do not take up much more space than LOD4 files. Although current conversion frameworks favour updating CityGML LODs one by one over using a single workflow, it would be more efficient in a fully automated method for LOD4 to use the single workflow approach like in Deng et al. (2016) or Billen et al. (2014).

*GenericCityObject* is a general module in CityGML that accepts any GML compliant geometries. While the features in this module validate correctly, the downside is that the same geometries are not allowed in other CityGML core modules. For example the quick export function in ArcGIS PRO Data Interoperability extension converts IFC entities into *GenericCityObject* CityGML features (Appendix C). For a single workflow approach to work in this environment the *GenericCityObject* features have to be mapped to other LODs and further down the line into their thematic modules from OGC (2012).

It is observed that method 'Entity conversion ratio' failed to pinpoint any small errors in the data but created a baseline for measuring correctness and completeness when using the same eXtensible Markup Language (XML) formatter. A conversion workflow where all IFC data is converted first into the highest LOD as *GenericCityObjects* before applying the results may improve the accuracy of this method.

Method ‘XSD validation’ correctly revealed errors in the schema definitions. However, the severity of the errors is not assessed. In the case study a file that did not pass XML Schema Definition (XSD) validation could still be read in by FME.

Method ‘Unit scale validation’ revealed errors in conversion scale and measurement units as intended. Data conversion workbench methods did not know how to handle unit precisions in the ArcGIS Data Interoperability extension.

Method ‘Estimating transformation positional accuracy’ correctly identified CityGML files with incorrect georeference coordinates. Unfortunately any incorrect georeference can not be erased from the IFC entities with FME. However, georeferencies in the IFC files can be fixed in a text editor manually without FME transformers or ArcGIS PRO.

The ‘Geometric processing consistency’ method indicated no disturbances in object positions in CityGML. Thus, incorrect measurements in models are not due to errors in geometric processing of converted features.

Method ‘Conformance to modelling guidelines’ marked the converted CityGML file guideline compliant by editing the XML after confirming the conversion results with other methods. The added attribute could be seen when querying the Myran building.

Method ‘Time elapsed for first rendered view in seconds’ works as an indicator to detect the generalisation level of geometries in the converted data. LOD4 data stores caused increased rendering times. This was likely due to the amount of elements since triangulation was used to generalise all geometries in the workflows.

Method ‘Visual inspection’ detected most of the conversion errors. However, errors in the XSD and early versions of the IFC data did not always cause visual inconsistencies or disturbances (Bilejecki et al. 2016a). Incorrect data extends and overlapping objects can go unnoticed. The visual results were used to confirm modelling guideline conformance (SIG 3D).

Method ‘Elapsed conversion time in seconds’ did not reveal important differences in conversion times between comparable data conversions. However, increased conversion times would likely imply that the conversion workflow used is inefficient.

Method ‘Recording file size’ did not reveal any unusual derivation in converted files. Thus, the data conversions were mostly complete.

The methods ‘Entity conversion ratio’, ’XSD validation’ and ‘Visual inspection’ are the most common methods to appear in research literature. Each of these methods has its own faults. The ‘Entity conversion ratio’ method is unable to function properly without multiple comparable conversion results or a true one-to-one documented conversion framework. The ‘XSD validation’ method identifies all CityGML files not conforming to the version 2.0 core schema correctly but is unable to validate CityGML files containing extra properties. The ‘Visual inspection’ method identifies errors in data conversions that are large enough to cause visual disturbance, however, some errors are either not visual in nature or the file structure is too complex and time consuming to be verified manually (Bilejecki et al. 2016a).

## 5.4 Conversion results

Autodesk Revit format reader in FME and ArcGIS PRO Data Interoperability extension does not improve the conversion results if supplementary data is not considered in the case of *GenericCityObjects* (Appendix C).

Georeferencing issues are found in all of the conversion workflows. The overall quality of data conversions was good except for inconsistencies caused by workflows imported from the FME 2017 into the ArcGIS environment for FME 2018 workbench. The results from method 'Estimating transformation positional accuracy' support this finding.

The created metrics for CityGML 2.0 leave some room for interpretation. A more complete analysis on the data transformation components like completeness, correctness and accuracy might not provide any more knowledge without the addition of extensive conversion logs. Developing automated processes to convert IFC data to CityGML can change this.

In the final report of the GeoBIM project (Ohori et al. 2018) the following IFC modelling recommendations are proposed:

- The georeference for IFC files should be set using *IFCSite* so that the offset from *IfcGeometricRepresentationContext* is taken into account. The *TrueNorth* attribute should also be set.
- IFC files should use volumetric objects in the definition of *IfcRepresentation* Item as often as possible.
- IFC files should not contain any intersections if possible. Overlapping objects are bad.
- Empty spaces in IFC should be modelled explicitly as *IfcSpaces*.
- Always use most specific entity type (subclass) for features.

Based on the findings these are the additional suggestions for IFC to CityGML conversions:

1) The IFC files could be complemented by additional information to identify the corresponding CityGML counterparts.
2) The exterior parts of a building should be modelled as closed spaces.
3) To facilitate automatic LOD generation information can be added to IFC building entities about whether they are modelled as a part of volume or surface.
4) The georeference of the IFC file should be checked prior to the conversion and the georeference of the resulting CityGML file verified.
5) Building floors should be separated on actual structural level and have an attribute specifying whether the floor reaches below the ground level.

6) Openings should be included directly into IFC slab and wall entities.
7) Other information normally not modelled in IFC could be added in the form of extra property sets.

# 6 Conclusions

The lack of eXtensible Markup Language (XML) formatters for writing City Geography Markup Language (CityGML) data narrows down the possible software solutions to convert IFC files into CityGML. In order to successfully compete with laser point clouds the IFC to CityGML process needs to be automated. To enable the modelling automatisation proper IFC modelling instructions are required. In any case laser point clouds are required to model areas outside of buildings.

Another conclusion based on the research and the case study work is that currently the effort should concentrate on correctness/reliability of the conversion methods rather than optimising computing performance.

## 6.1 Requirements for data

The requirements for the LOD3 and LOD4 converted data usually follow the structure illustrated in Figure 2.8. For now the major differences in requirements come from how national guidelines and modelling practises evolve to support more application of CityGML. The data generalisation can follow the recommendations set forth in Infrastructure for Spatial Information in Europe (INSPIRE) or in the CityGML encoding documentation. A different LOD scheme for converted data can also be selected or created like in Figure 2.3. In a best case scenario for CityGML 2.0 the semantics of the data are handled using linked custom code lists that are up-to-date. Such an approach requires the prebuilding of the link contents from individual IFC files and general databases.

## 6.2 Quality Metrics

The key quality metrics identification is based on how well the evaluation criterion are performing in the case study to asses the outcome of the data conversions. The quality metrics ‘XSD validation result ‘, ‘Estimated elapsed time before first view is rendered in seconds’ and ‘Elapsed time to complete data conversion in seconds ’ are deemed as secondary metrics since they did not reveal major quality inconsistencies during the case study.

The metrics ‘Conversion accuracy acceptance derivation’, ‘Object to entity centroid standard deviation’ and ‘Visual appearance of geometries and related attributes’ together with ‘If special modelling guidelines apply‘ performed best in the case study. Although many of the CityGML files showed similar characteristics, the aforementioned methods produced very useful information about the quality of the data conversions.

The metric ‘Scale offset factor for LOD or feature’ also performed well and errors found can be corrected afterwards if the error is noticed in time.

The metric ‘File size’ did not find any important file size differences between the workflows except for the usage of RVT reader (extra data) producing slightly larger CityGML files.

The quality metrics ‘Object to entity centroid standard deviation’, ‘Scale offset factor for LOD or feature’, ‘Visual appearance of geometries and related attributes’ and ‘Estimated elapsed time before first view is rendered in seconds’ for Up:Town are missing from the result tables (Appendix B) due to the too high hardware requirements of the Up:Town model.

## 6.3 Methods to compute the quality metrics

The overall computing of test methods to produce the selected quality metrics proved to be successful with the detailed findings below.

Using the old IFC reader increased the entity counts for Specific geometries. Also the ‘IFCBooleanClippingResult_1’ geometry sample could not be converted to CityGML with the workflows executed in ArcGIS PRO Data Interoperability extension.

Unexpectedly the XML Schema Definition (XSD) validation for the Myran CityGML models failed with ArcGIS Pro Data Interoperability extension when testing the quick import and export functions.

The evaluation of all quality metrics could not be completed for the Up:Town model because of hardware limitations in the case study.

The methods tested for FME 2017 and the ArcGIS PRO Data Interoperability extension can create the planned quality metrics expect in the case of Up:Town.

The method ‘Entity conversion ratio’ is unable to differentiate if complex entities in the conversion consist of multiple CityGML objects. Therefore, the ratio created by this method is only comparable against very similar conversions.

## 6.4 Future research

Since converted data can come from multiple sources and the use of conversion frameworks is not limited, CityGML model contents can vary depending on the chosen conversion methodology. Therefore, constant tracking of the different CityGML object versions is important.

Further study is required to specify requirements on the conversion method documentation and on the expected CityGML output in order to enable more accurate estimation of the expected result for certain type of IFC data.

The metrics and methods to measure the quality of the IFC to CityGML data conversions researched in this thesis are likely to be also valid for the CityGML to IFC conversions. This remains to be verified.

## References


AIA LOD Specification 2018. Retrieved 27 February, 2019, from https://bimforum.org/lod/

Arroyo Ohori, K., F. Biljecki, A. Diakité, T. Krijnen, H. Ledoux, and J. Stoter. 2017. TOWARDS AN INTEGRATION OF GIS AND BIM DATA: WHAT ARE THE GEOMETRIC AND TOPOLOGICAL ISSUES? *ISPRS Ann. Photogramm. Remote Sens. Spatial Inf. Sci.,* IV-4/W5**:** 1-8. DOI: 10.5194/isprs-annals-IV-4-W5-1-2017

Bengtsson, J., and M. Grönkvist 2017. Performing Geographic Information System Analyses on Building Information Management Models. PhD Thesis

Biljecki, F., J. Stoter, H. Ledoux, S. Zlatanova, and A. Çöltekin. 2015. Applications of 3D city models: State of the Art Review. 4**:** 2842.

Biljecki, F., H. Ledoux, X. Du, J. Stoter, K. H. Soon, and V. H. S. Khoo. 2016a. THE MOST COMMON GEOMETRIC AND SEMANTIC ERRORS IN CityGML DATASETS. *ISPRS Ann. Photogramm. Remote Sens. Spatial Inf. Sci.,* IV-2/W1**:** 13-22. DOI: 10.5194/isprs-annals-IV-2-W1-13-2016

Biljecki, F., H. Ledoux, and J. Stoter. 2016b. An improved LOD specification for 3D building models. *Computers, Environment and Urban Systems,* 59**:** 25-37. DOI: https://doi.org/10.1016/j.compenvurbsys.2016.04.005

Biljecki, F., H. Ledoux, and J. Stoter. 2016c. GENERATION OF MULTI-LOD 3D city modelS IN CityGML WITH THE PROCEDURAL MODELLING ENGINE RANDOM3DCITY. ISPRS Ann. Photogramm. Remote Sens. Spatial Inf. Sci., IV-4/W1: 51-59. DOI: 10.5194/isprs-annals-IV-4-W1-51-2016

Biljecki F. 2017. Level of detail in 3D city models. Doctoral dissertation. Netherlands: Delft University of Technology.

Biljecki F, Ledoux H, Stoter J (2017): Generating 3D city models without elevation data. Computers, Environment and Urban Systems, 64:1–18. doi:10.1016/j.compenvurbsys.2017.01.001

Billen, R., A.-F. Cutting-Decelle, O. Marina, J.-P. de Almeida, C. M., G. Falquet, T. Leduc, C. Métral, et al. 2014. 3D city models and urban information: Current issues and perspectives. I-118. DOI: 10.1051/TU0801/201400001

BIMTalk. URL:https://bimtalk.co.uk/. Retrieved 14 March 2019.

BuildingSMART International Limited. (2016.) IFC4 Documentation. Retrieved 14 February, 2019, from http://www.buildingsmart-tech.org/ifc/IFC4/Add2/html/

CityGML 3.0 Development. URL:https://GitHub.com/opengeospatial/CityGML-3.0. Accessed 27 February 2019.

Chowdhury, S. N., and S. Dhawan. 2016. Evaluation of key performance indicators of smartcities by Delphi analysis. In 2016 IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT), 337-342.

de Laat, R., and L. van Berlo. 2011. Integration of BIM and GIS: The development of the CityGML GeoBIM extension. 211-225 pp.

Dallasega, P., E. Marengo, W. Nutt, L. Rescic, D. Matt, and E. Rauch. 2015. *Design of a Framework for Supporting the Execution-Management of Small and Medium sized Projects in the AEC-industry*.

Deng, Y., J. C. P. Cheng, and C. Anumba. 2016. Mapping between BIM and 3D GIS in different levels of detail using schema mediation and instance comparison. Automation in Construction, 67: 1-21. DOI: https://doi.org/10.1016/j.autcon.2016.03.006

Donkers, S. 2013. Automatic generation of CityGML LoD3 building models from IFC models. PhD Thesis

Donkers, S., H. Ledoux, J. Zhao, and J. Stoter. 2016. Automatic conversion of IFC datasets to geometrically and semantically correct CityGML LOD3 buildings. 20**:** 547-569. DOI: 10.1111/tgis.12162

El-Mekawy, M., A. Östman, and I. Hijazi. 2012. A Unified Building Model for 3D Urban GIS. 1: 120.

El-Mekawy, M., A. Östman, and I. Hijazi. 2012b. An Evaluation of IFC-CityGML Unidirectional Conversion.

Floros, G., I. Pispidikis, and E. Dimopoulou. 2017. INVESTIGATING INTEGRATION CAPABILITIES BETWEEN IFC AND CityGML LOD3 FOR 3D CITY MODELLING. *Int. Arch. Photogramm. Remote Sens. Spatial Inf. Sci.,* XLII-4/W7**:** 1-6. DOI: 10.5194/isprs-archives-XLII-4-W7-1-2017

FME website. URL: https://www.safe.com. Accessed 4 March 2019.

GeoBIM 2017. URL:https://3D.bk.tudelft.nl/projects/GeoBIM/. Accessed 4 March 2019.

GeoBIM benchmark 2019. URL:https://3D.bk.tudelft.nl/projects/GeoBIM-benchmark/. Accessed 9 May 2019.

Giovanella, A., P. Bradley, and S. Wursthorn. 2019. Evaluation of Topological Consistency in CityGML.

IfcGeorefChecker – Documentation. URL:http://www.dd-bim.org/?page_id=31 . Retrieved 4 May 2019.

IfcOpenShell_CGAL. URL:https://GitHub.com/aothms/IfcOpenShell_CGAL. Accessed 4 March 2019.

Ifcwiki 2007. URL: http://www.ifcwiki.org/index.php/IfcExplorer_CityGML_Export. Accessed 4 March 2019.

Ifc2CityGML. Automatic conversion of IFC models 2x3 to CityGML LOD 1/2/3/4. URL: https://GitHub.com/tudelft3D/ifc2CityGML. Accessed 4 March 2019.

INSPIRE (2014). Data Specification on Buildings – Technical Guidelines, Version 3.0, European Commission Joint Research Centre, Retrieved 27 February, 2019, from http://inspire.ec.europa.eu/id/document/tg/bu

Isikdag, U., and S. Zlatanova. 2009. Towards Defining a Framework for Automatic Generation of Buildings in CityGML Using Building Information Models. 79-96 pp.

KIT FZKViewer. URL:https://www.iai.kit.edu/1302.php. Accessed 4 March 2019.

Kutzner, T., and T. Kolbe. 2018. CityGML 3.0: Sneak Preview.

Laakso, M., and A. Kiviniemi. 2012. *The IFC Standard - A Review of History, Development, and Standardization*.

Ledoux, H. 2013. On the Validation of Solids Represented with the International Standards for Geographic Information. Computer-Aided Civil and Infrastructure Engineering, 28. DOI: 10.1111/mice.12043

Maas, H.-G., and G. Vosselman. 1999. Two algorithms for extracting building models from raw laser altimetry data. ISPRS Journal of Photogrammetry and Remote Sensing, 54: 153-163. DOI: https://doi.org/10.1016/S0924-2716(99)00004-0

McGlinn, D. K., H. Wicaksono, w. lawton, M. Weise, N. Kaklanis, i. petri, and D. Tzovaras. 2016. *Identifying Use Cases and Data Requirements for BIM Based Energy Management Processes*.

Mohanty, S. P., U. Choppali, and E. Kougianos. 2016. Everything you wanted to know about smart cities: The Internet of things is the backbone. IEEE Consumer Electronics Magazine, 5: 60-70. DOI: 10.1109/MCE.2016.2556879

OGC 2012. OGC City Geography Markup Language (CityGML) encoding standard version 2.0.0.

OGC 2015. OGC Smart Cities Spatial Information Framework. Retrieved 09 May, 2019, from https://portal.opengeospatial.org/files/?artifact_id=61188

OGC 2016. CityGML Quality Interoperability Experiment Engineering Report (2016). Retrieved 27 February, 2019, from http://www.opengeospatial.org/pressroom/pressreleases/2458

Ohori, K., A. Diakité, H. Ledoux, and J. Stolter. 2018. GEOBIM project Final report. Delft University of Technology, Retrieved 9 May, 2019, from https://3D.bk.tudelft.nl/projects/geobim/#final-reports

Prieto, I., J. L. Izkara, and R. Béjar. 2017. A continuous deployment-based approach for the collaborative creation, maintenance, testing and deployment of CityGML models. *International Journal of Geographical Information Science,* 32**:** 282-301. DOI: 10.1080/13658816.2017.1393543

Rahm, E., and P. A. Bernstein. 2001. A survey of approaches to automatic schema matching. The VLDB Journal. 10: 334-350. DOI: 10.1007/s007780100057

Saran, S., K. Oberai, P. Wate, A. Konde, A. Dutta, K. Kumar, and A. J. J. o. t. I. S. o. R. S. Senthil Kumar. 2018. Utilities of Virtual 3D city models Based on CityGML: Various Use Cases. Springer India. *Journal of the Indian Society of Remote Sensing,* 46**:** 957-972. DOI: 10.1007/s12524-018-0755-5

SIG 3D / Quality Working Group. 2017. Modeling Guide for 3D Objects - Part 2: Modeling of Buildings (LoD1, LoD2, LoD3). ULR:http://en.wiki.quality.sig3D.org/index.php/Modeling_Guide_for_3D_Objects_-_Part_2:_Modeling_of_Buildings_(LoD1,_LoD2,_LoD3) . Accessed 10 May, 2019.

Song, Y., X. Wang, Y. Tan, P. Wu, M. Sutrisna, J. C. P. Cheng, and K. Hampson. 2017. Trends and Opportunities of BIM-GIS Integration in the Architecture, Engineering and Construction Industry: A Review from a Spatio-Temporal Statistical Perspective. 6**:** 397.

Steinhage, V., J. Behley, S. Meisel, and A. Cremers. 2010. *Automated updating and maintenance of 3D city models*.

Stouffs, R., H. Tauscher, and F. Biljecki. 2018. Achieving Complete and Near-Lossless Conversion from IFC to CityGML. 7: 355.

Sun J., Olsson P.-O. Eriksson H., and Harrie L., 2019. Evaluating the Geometric Aspects of Integrating BIM Data into City Models. Journal of Spatial Science. DOI: https://doi.org/10.1080/14498596.2019.1636722

Suveg, I., and G. Vosselman. 2004. Reconstruction of 3D building models from aerial images and maps. ISPRS Journal of Photogrammetry and Remote Sensing, 58: 202-224. DOI: https://doi.org/10.1016/j.isprsjprs.2003.09.006

van Berlo, L., T. Dijkmans, and J. Stoter. 2013. EXPERIMENT FOR INTEGRATING DUTCH 3D SPATIAL PLANNING AND BIM FOR CHECKING BUILDING PERMITS. *ISPRS Ann. Photogramm. Remote Sens. Spatial Inf. Sci.,* II-2/W1**:** 279-284. DOI: 10.5194/isprsannals-II-2-W1-279-2013

Qing, L., G. Tao, and W. J. Ping. 2014. Study on Building Lifecycle Information Management Platform Based on BIM.

# Appendix A

# Data conversion workflows for GeoBIM benchmark

As a reference to what the different blocks in the FME workbench do and how they work please use the software online help page at https://www.safe.com/fme/ .
The descriptions of data are available at https://3D.bk.tudelft.nl/projects/geobim-benchmark/data.html .

Based on the requirements, metrics and methods revealed in the literature study a series of methods for measuring converted IFC data in CityGML is devised. Extracting the metrics for creating resulting quality indicators can vary for each different data conversion implementation and software. Despite this, the goal of the methods is still to be independent from actual conversion software.

The Inspire 2013 data specification document recommends absolute accuracies of <=2 m for LODs zero and one and 1 m for LOD two. However, national standards and modelling instructions for building data might differ from these recommendations. The main criteria for LOD classification is based on visual representations of CityGML data. Other types of LOD classification criterion besides accuracy are equally valid methods of dividing CityGML data into LODs. The baseline for CityGML conversion quality in this study is the guidelines of CityGML LOD modelling released by SIG 3D. These guidelines are made together with the German geoportal to satisfy the needs of 3D modelled cadastral data in Germany.

## FME 2017 Workbench

FME offers two ways to read the data into IFC files. The de-facto way is to use IFC reader to fetch and modify important information. The alternative is to use Revit reader that is meant to be used with .rvt format files and allows for more advanced data inclusions when used with IFC files. The saved feature data and IFC properties differ depending on witch reader is used in FME 2017 to read in the IFC file. Workflows containing properties found only from one reader realisation do not necessarily work when data is read in using another one. Certain properties in common with both readers are immune to breaking, even if different readers are used in the same FME workbench to access IFC files. The common properties found in IFC and Revit readers are:

- fme_geometry
- fme_type
- GlobalId
- Name
- Tag

Using the common properties within a transformer in FME 2017 only stops the workflow from breaking. Actual geometries or properties of read in objects can differ depending on which reader is used. Therefore, the contents mapped to ‘IFC reader’ and ‘Revit reader’ objects may be different depending on the IFC file contents.

FME has a Translation Log that shows how many features get read in and written out but it is not good enough for the purpose of counting objects from IFC to CityGML conversions. This is due to the fact that objects which are ‘null’ or do not contain geometries are not accepted by CityGML validators. Instead the translator GeometryFilter in FME 2017 is used for this purpose (Figure A1). Other metrics like time elapsed since process start to finish and parameters in use by FME transformers are also in the translation log.

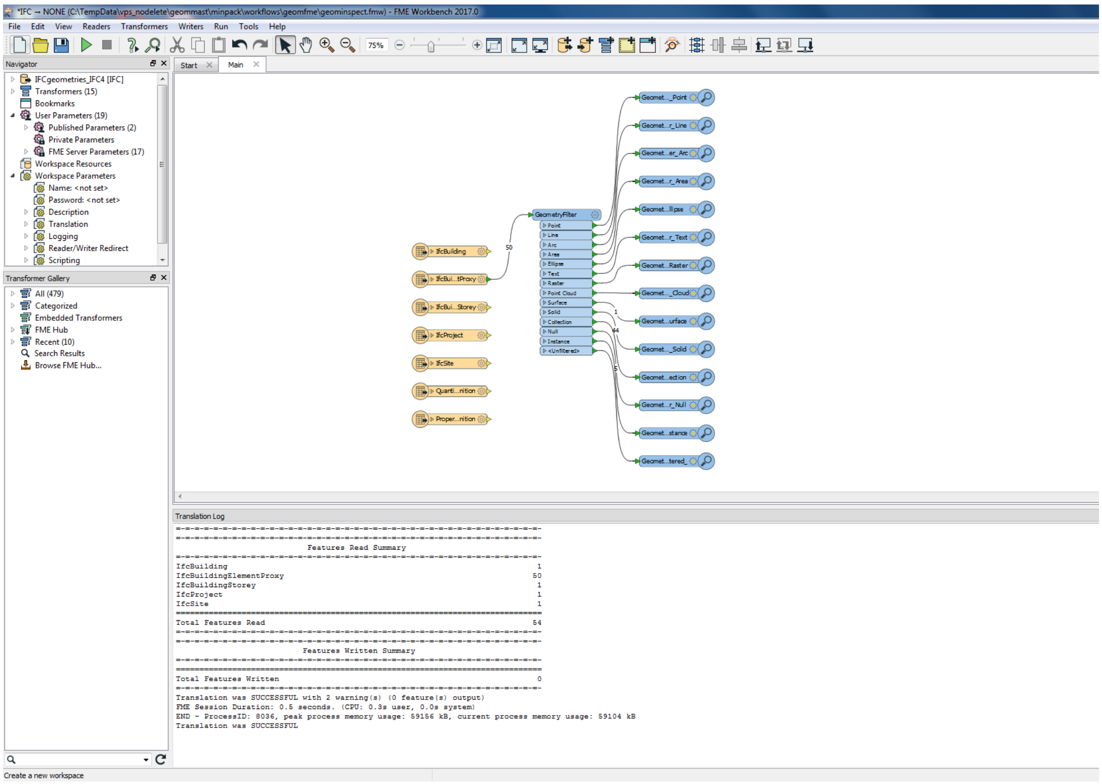


Figure A1: The GeometryFilter in FME 2017 that is used to count conversion objects/ entities (FME screen capture).

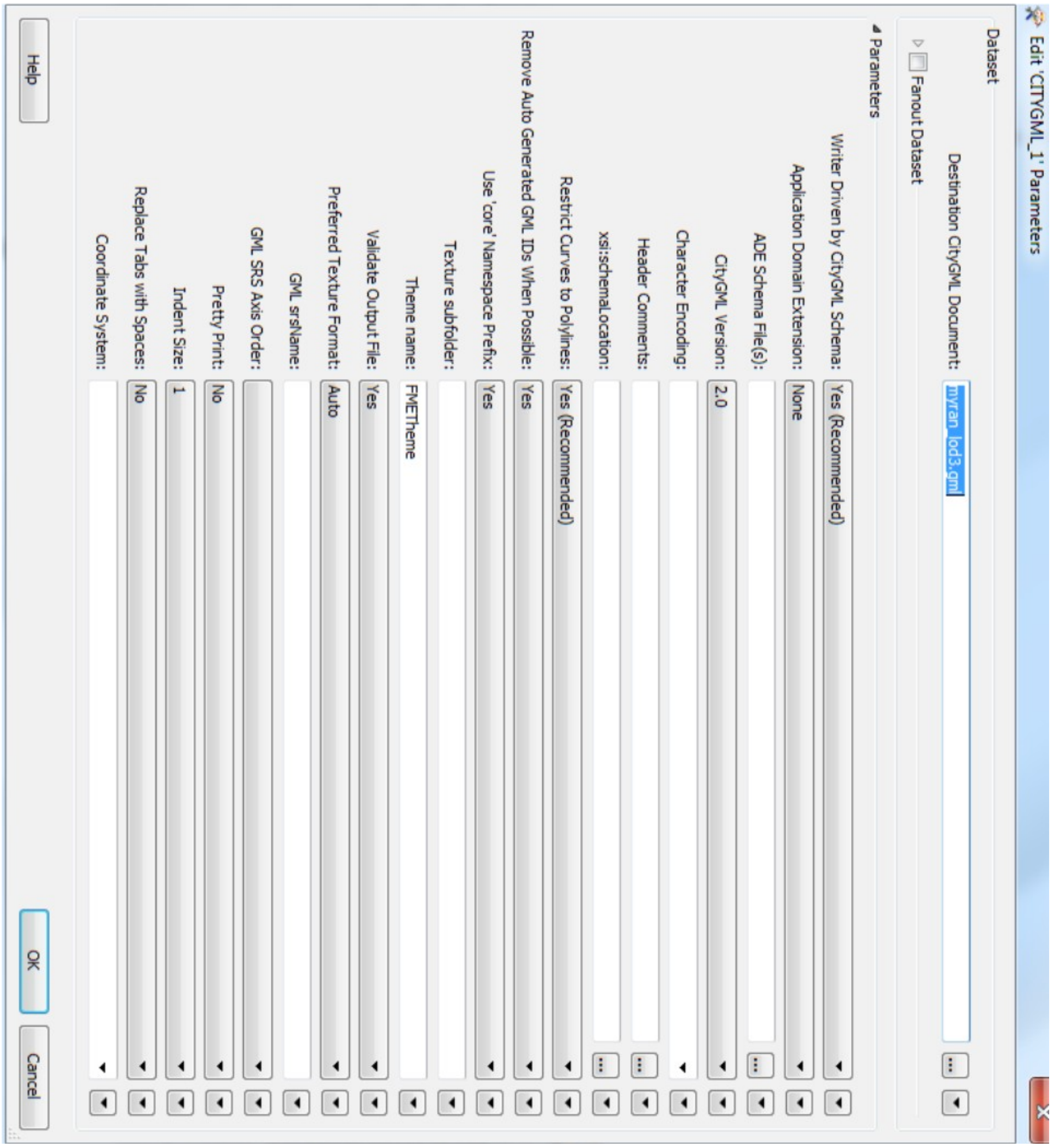


Figure A2: CityGML writer options in FME 2017.

Sometimes geometries in the IFC format need to be instantiated first so that they can be processed and filtered by FME. The transformer ‘Geometry filter’ is used for this purpose. Features that do not contain geometry can not be geometrically processed by FME and thus, the required attributes and geometry traits mapped to ‘null’ need to be copied to the converted geometry.

Processed geometries are not always directly compatible with allowed CityGML module geometry properties and have to be processed into another feature type by an additional transformer. Because of this, a *GeometryCoercer* FME transformer is applied in order to extract surfaces for deaggregation.

Most geometrical traits or attributes can only be tested with a single feature class. An exception to this is the *AttributeFilter* transformer that can filter multiple feature types based on individual inputs.

## ArcGIS PRO Data Interoperability extension

The Data Interoperability extension provided by ESRI (Environmental Research Institute) is an ArcGIS PRO extension that allows for geodatabase data integration with most formats supporting FME translations. The quick import and export are features within this extension that facilitate fast data handling within ArcGIS Pro Catalog. The quick import and export features work with a plug and play mindset. Thus, no actual workflow construction is required to convert IFC data. Read and write settings are enough for the conversion features.

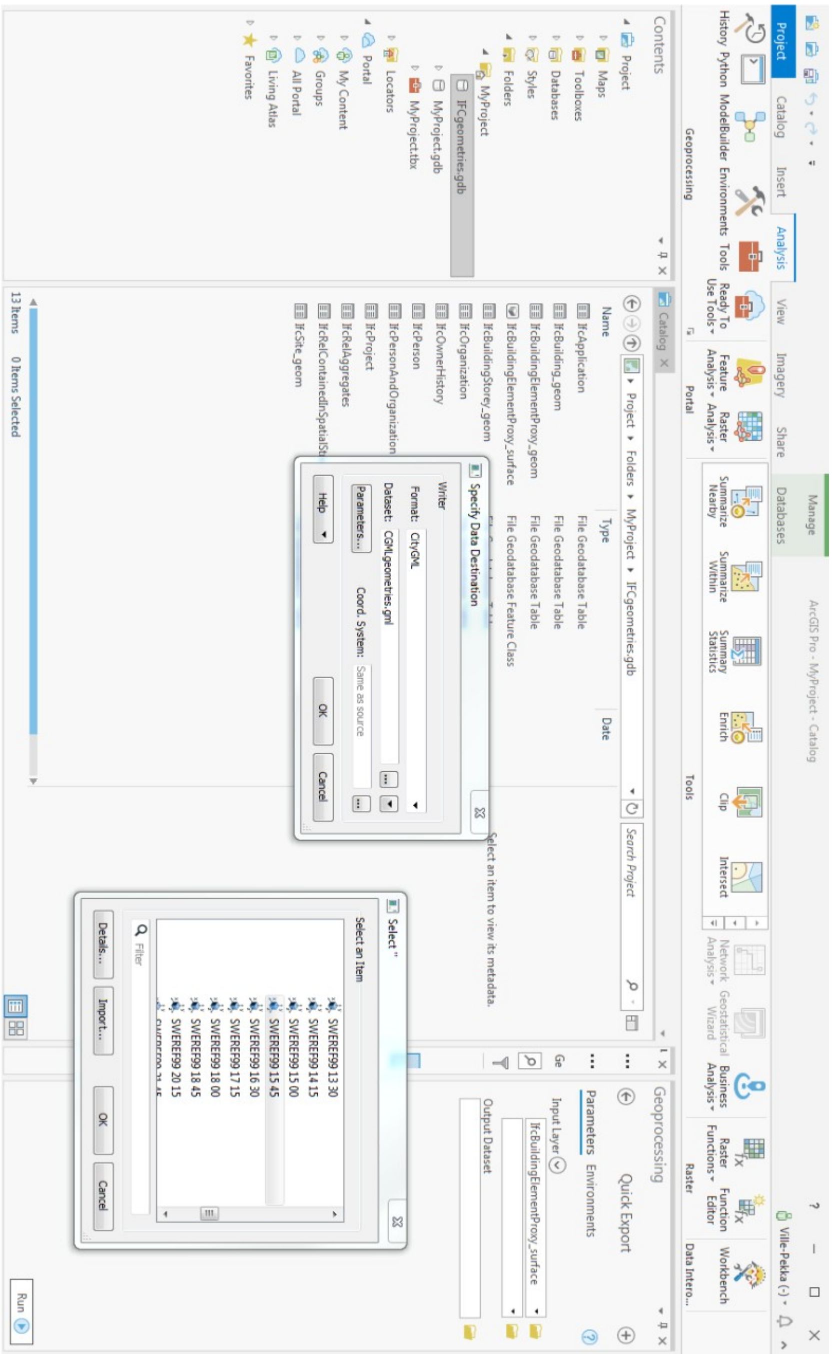


Figure A3: Read and write settings in ArcGIS PRO quick CityGML export.

The amount of available data depends mainly on the success of the IFC data quick import to a geodatabase. Selecting the right settings for the IFC file reader creation is important.

The next step in the data interoperability quick import and export workflow is the quick exporter. The CityGML writer options are similar to that of the FME 2017 product. The workbench version in the ArcGIS Pro extension is 2018.1. Figure A4 depicts the usual settings for exporting (writing out) CityGML 2.0 data. The FME 2017 workbench workflows are imported into the FME 2018.1 version inside ArcGIS PRO data  interoperability extension for testing.

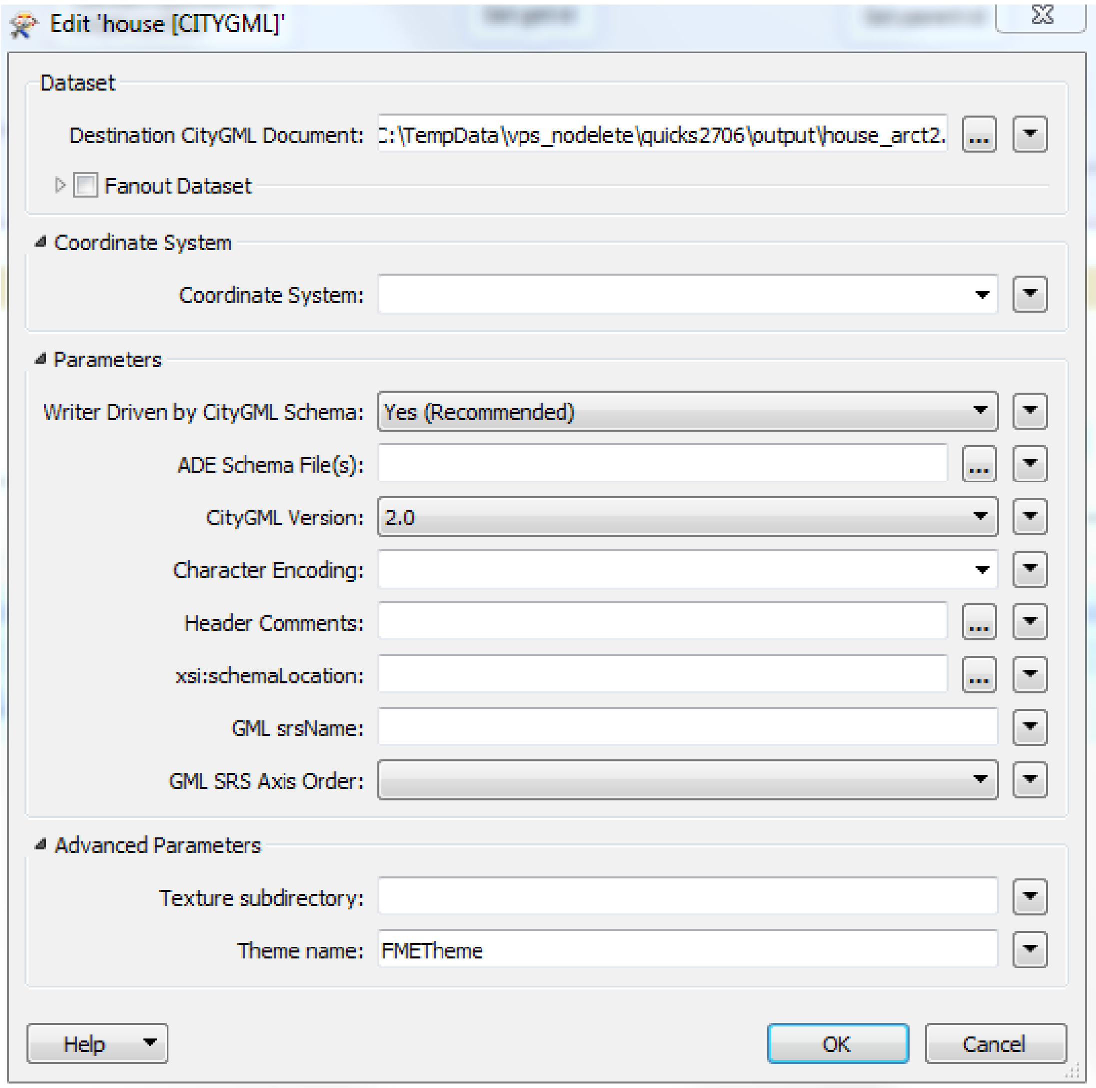



Figure A4: Figure of workbench 2018 CityGML export.

The XSD validation had to be switched from the online (http://geovalidation.bk.tudelft.nl/schemaCityGML/) validator in middle of the conversion tests. The FME XSD validator is used instead.

## Workflows for Myran

The workflow for Myran is large in the FME workbench and is therefore provided only as an accompanying zip file for the GeoBIM benchmark submit.

The prebuild conversion workflows produced uncompleted results in FME 2017 and had to be modified for the RVT reader (Figure A5; Figure A6). These modifications followed the same guidelines for modelling from SIG 3D. However, the transformers used to separate different faces from the floor mesh did not work the same way that they did with the IFC reader data mappings. The glass panels in the exterior Myran elements also had this problem.

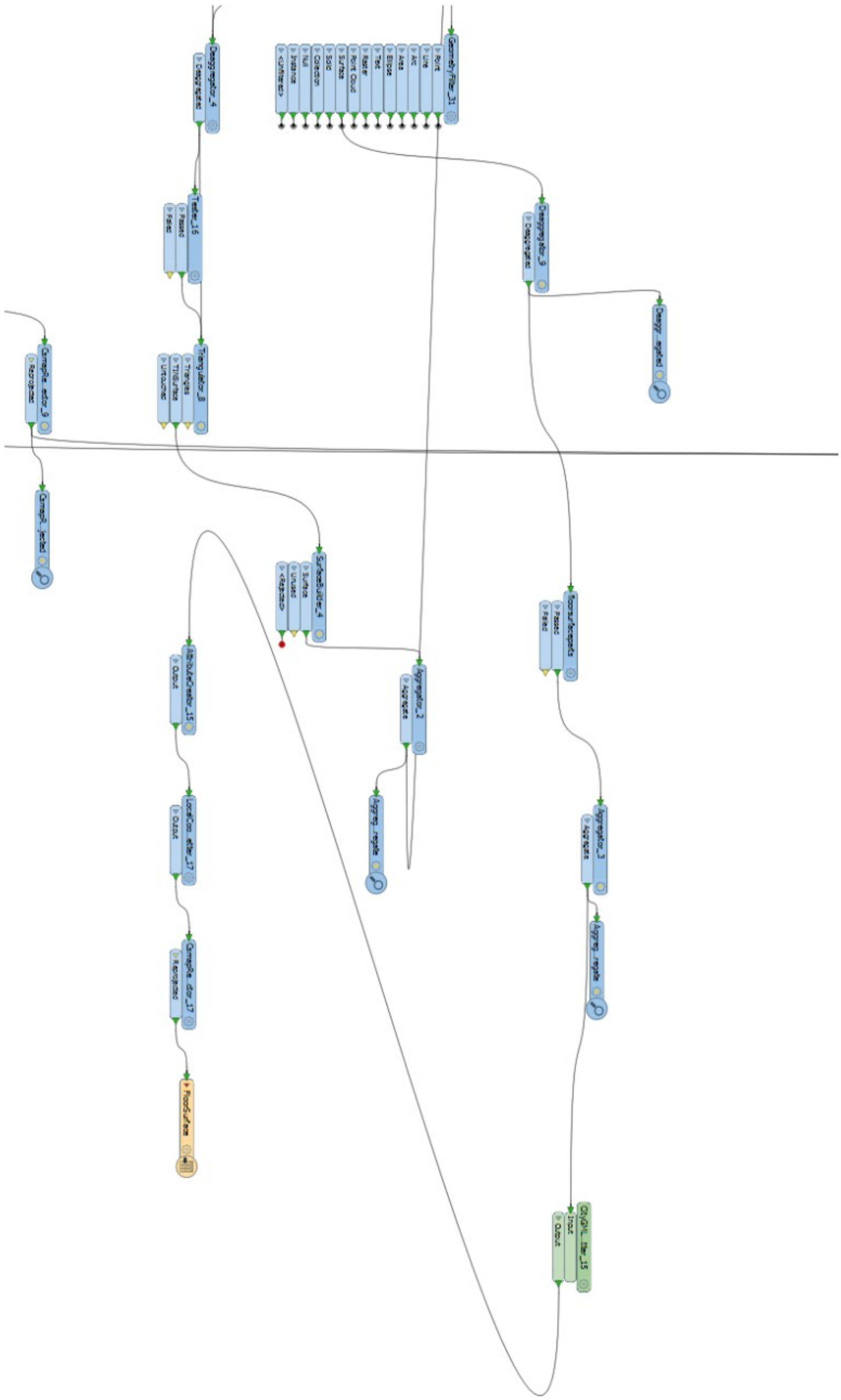


Figure A5: Floor.

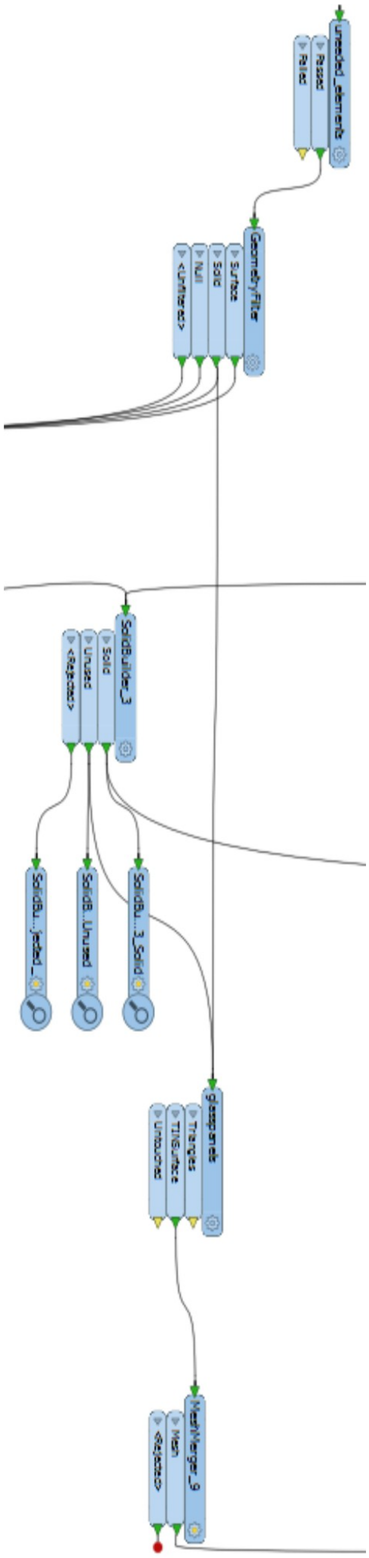



Figure A6: Glass.

## More details on data interoperability quick import and export workflow construction and methodology

The IFC OLD reader (Figure A7) is being used in converting the sample geometries with ArcGIS PRO Data Interoperability extension. Although the normal IFC reader can import data into .gdb with the quick export functionality, it is unable to convert the contents into CityGML.

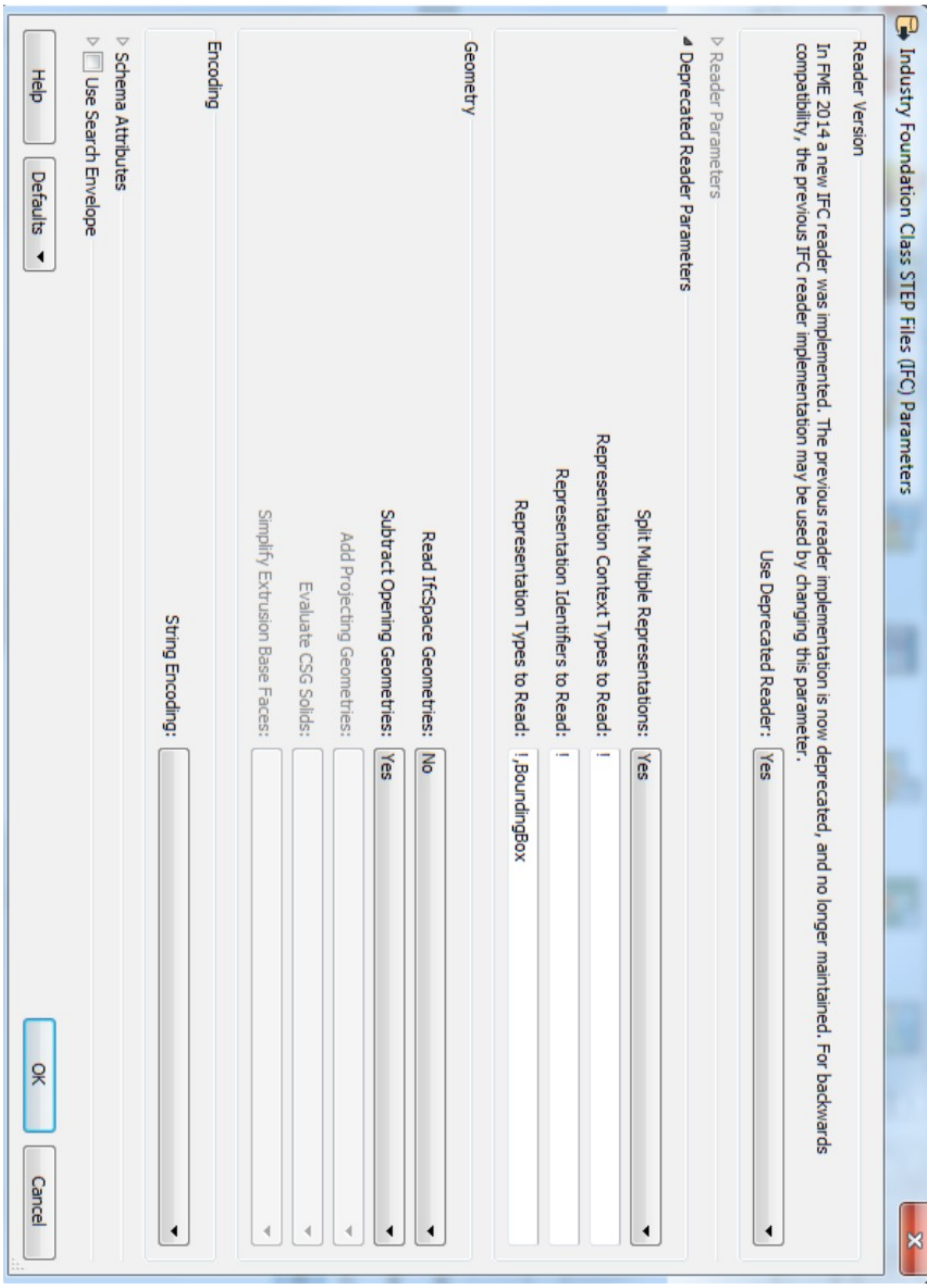


Figure A7: Old IFC settings.

## Elapsed time in processing estimation

The elapsed processing time of the conversion from the IFC file to the resulting CityGML file is recorded. The elapsed time is summed up from different software processing logs. An example constructed from the ArcGIS Pro interoperability quick export and import logs can be seen in Figure A8.

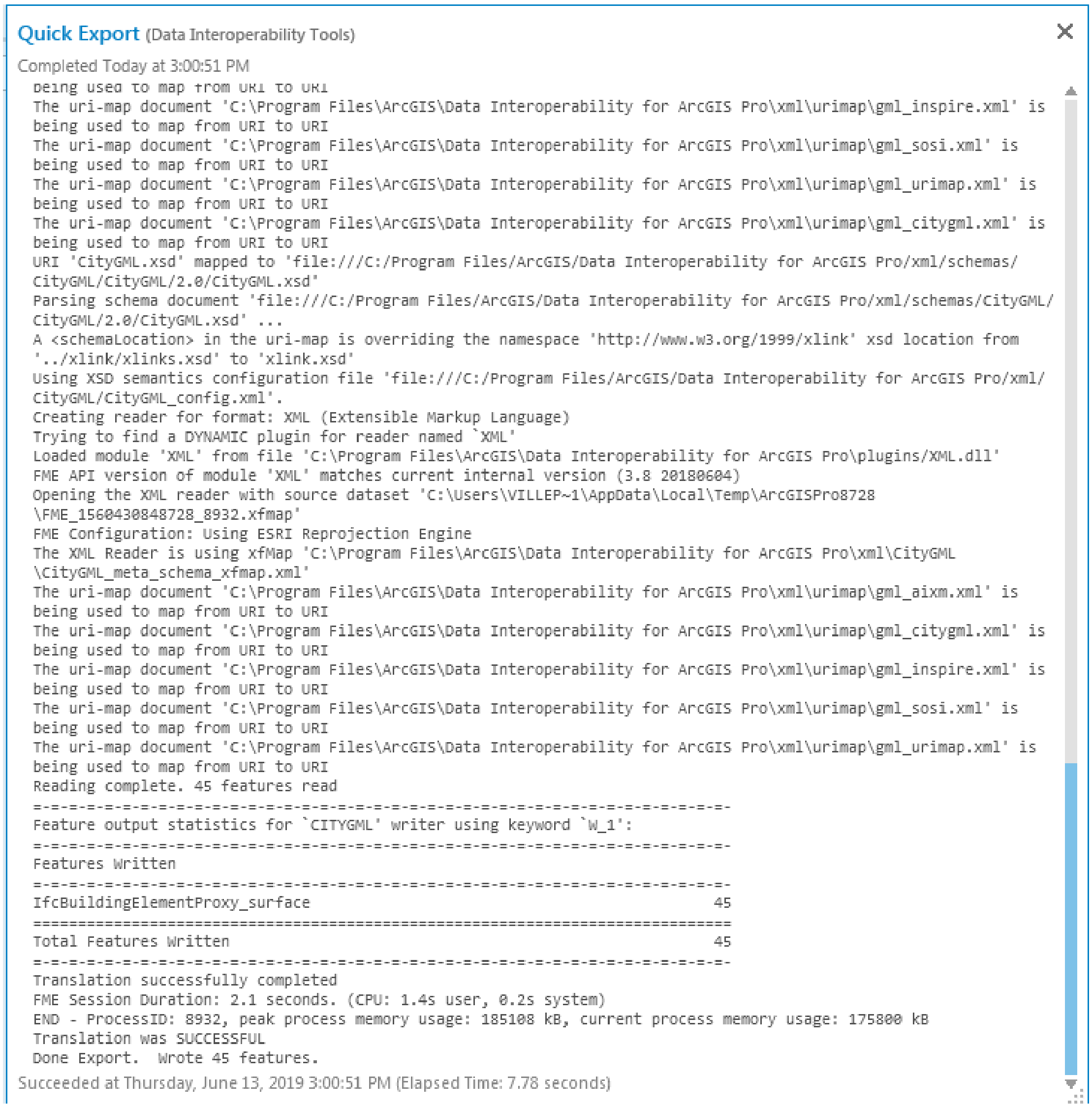



Figure A8: Example picture showing ArcGIS PRO logs for IfcGeometries_IFC4.

The total elapsed conversion time in this case is calculated as a sum of the quick import and export toolbox processes in ArcGIS Pro.

## Example conversions

The IfcGeometries are used as a basis to construct the data conversion methods in line with the examples provided in the FME help pages at https://knowledge.safe.com/articles/591/bim-tutorial.html .

The first IFC file in Task 4 of the GeoBIM benchmark involves converting IFCGeometries.ifc (which is provided in 2X3 TC1 and IFC4 ADD2 versions). The IFC files are read in using IFC (.ifc) and Revit(.rtv) readers and a separate workflow is constructed for each reader in FME 2017 workbench.

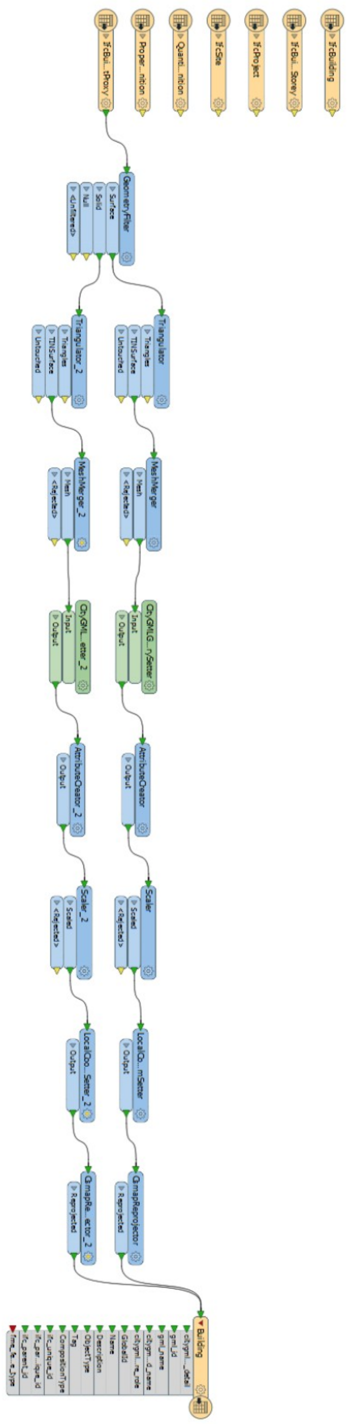


Figure A9: FME 2017 workflow for IFCGeometries.ifc 2x3 TC1 and IFC4 ADD2 with IFC Reader.

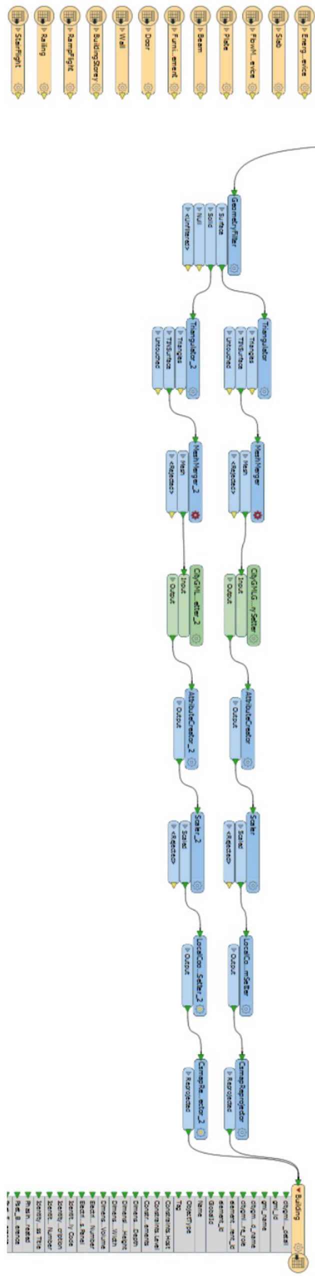


Figure A10: FME 2017 workflow for IFCGeometries.ifc 2x3 TC1 and IFC3 ADD2 with Revit(.rtv) Reader..

Because some of the methods have to be executed as part of the IFC to CityGML translation workflow in FME 2017, the IFCgeometries conversion results are used to demonstrate the conversion evaluation methodologies needed for counting the number of reads in entities from IFC files (Figure A1). The resulting CityGML 2.0 file is validated using external XSD validation from http://geovalidation.bk.tudelft.nl/schemaCityGML/ . A comparison using georeferenced data and IFC units is done to evaluate the placement of converted IFC entities (a georeference must be assigned to CityGML geometries) and to confirm FME reader and transformer data processing setting assumptions. In the end FME

2017 IFCgeometries conversion results are visually compared with the read in data from FME Data Inspector 2017.

## Georeference method and semantics

CityGML geometries need to be georeferenced on their geometry or parent entity level. Two different ways of setting georeference are used in the workflows. The first method consists of two transformers *LocalCoordinateSystemSetter* and *CsmapReprojector*. Together these transformers are used to perform a unitary transformation. Note that the coordinate values for *LocalCoordinateSetter* correspond to *IFCSite* WGS84 (SWEREF99) Origin. The Scaler transformer is only used to correct unit scaling for the transformation. Heights for the reprojection to EPSG:3013 are reconstructed automatically in the CityGML geometry. This combination of transformers can be used to write different georeference to individual geometries in CityGML. The second method is to use CityGML writer in FME 2017 to set a coordinate system for the city model.

Attributes for the CityGML model are set manually with the recommendations in the SIG 3D modelling guide in mind. Only attributes carried over from the IFC file or added are written out. Entries and tags for empty values are not created in the CityGML file for attributes (features in FME feature storage (FFS)). The CityGML object table entries are called traits in FME and need to be exposed or created into the CityGML file separately for an edit.

# Appendix B

# Case study conversion results

**Table B1** Results part 1 of 4 (Specific IFC (2x3) geometries).

| Reader | **IFC** | **IFC OLD** | **RVT** |
|---|---|---|---|
| Conversion Software | FME 2017 Workbench | ArcGIS Pro Data Interoperability extension (trial) | FME 2017 Workbench |
| Converted entity ratio | 50/64 | 50/73 | 50/64 |
| Object to entity centroid standard deviation | Xy=0, h=0 | Xy=0, h=0 | Xy=0, h=0 |
| Conversion accuracy acceptance derivation | 0.0085 | 8.4853 | 0.0085 |
| Scale offset factor for LOD or feature | Pass | Pass | Pass |
| XSD validation result | Pass | Pass | Pass |
| If special modelling guidelines apply | No | No | No |
| Visual appearance of geometries and related attributes | Pass | Pass | Pass |
| Estimated elapsed time before first view is rendered in seconds | 1.8s | 0.1s | 1.8s |
| Elapsed time to complete data conversion in seconds | 2.6s | 6.7s | 8.4s |
| File size | IFC 31kt / CityGML 5872kt | IFC 31kt / CityGML 3525kt | IFC 31kt / CityGML 5866kt |

| Reader | **RTV** | **IFC** | **RVT** |
|---|---|---|---|
| Conversion Software | ArcGIS Pro Data Interoperability extension (trial) | ArcGIS Pro Data Interoperability extension (trial) FME 2018.1 Workbench | ArcGIS Pro Data Interoperability extension (trial) FME 2018.1 Workbench |
| Converted entity ratio | 49/64 | 49/64 | 49/64 |
| Object to entity centroid standard deviation | Xy=0, h=0 | Xy=0, h=0 | Xy=0, h=0 |
| Conversion accuracy acceptance derivation | 8.4853 | 0.0085 | 0.0085 |
| Scale offset factor for LOD or feature | Pass | Pass | Pass |
| XSD validation result | Pass | Pass | Pass |
| If special modelling guidelines apply | No | No | No |
| Visual appearance of geometries and related attributes | Fail | Fail | Fail |
| Estimated elapsed time before first view is rendered in seconds | 1.0s | 1.8s | 1.0s |
| Elapsed time to complete data conversion in seconds | 12.6s | 2.9s | 9.6s |
| File size | IFC 31kt / CityGML 3625kt | IFC 31kt / CityGML 6366kt | IFC 31kt / CityGML 6360kt |

**Table B2** Results part 2 of 4 (Specific IFC 4 geometries).

| Reader | **IFC** | **IFC OLD** | **RVT** |
|---|---|---|---|
| Conversion Software | FME 2017 Workbench | ArcGIS Pro Data Interoperability extension (trial) | FME 2017 Workbench |
| Converted entity ratio | 45/54 | 45/63 | 45/54 |
| Object to entity centroid standard deviation | Xy=0, h=0 | Xy=0, h=0 | Xy=0, h=0 |
| Conversion accuracy acceptance derivation | 0.0085 | 8.4853 | 0.0085 |
| Scale offset factor for LOD or feature | Pass | Pass | Pass |
| XSD validation result | Pass | Pass | Pass |
| If special modelling guidelines apply | No | No | No |
| Visual appearance of geometries and related attributes | Pass | Pass | Pass |
| Estimated elapsed time before first view is rendered in seconds | 1.0s | 1.0s | 1.2s |
| Elapsed time to complete data conversion in seconds | 2.5s | 6.2s | 8.1s |
| File size | IFC 27kt / CityGML 5012kt | IFC 27kt / CityGML 3002ktkt | IFC 27kt / CityGML 5006kt |

| Reader | **RVT** | **IFC** | **RVT** |
| --- | --- | --- | --- |
| Conversion Software | ArcGIS Pro Data Interoperability extension (trial) | ArcGIS Pro Data Interoperability extension (trial) FME 2018.1 Workbench | ArcGIS Pro Data Interoperability extension (trial) FME 2018.1 Workbench |
| Converted entity ratio | 44/54 | 44/54 | 44/54 |
| Object to entity centroid standard deviation | Xy=0, h=0 | Xy=0, h=0 | Xy=0, h=0 |
| Conversion accuracy acceptance derivation | 8.4853 | 0.0085 | 0.0085 |
| Scale offset factor for LOD or feature | Pass | Pass | Pass |
| XSD validation result | Pass | Pass | Pass |
| If special modelling guidelines apply | No | No | No |
| Visual appearance of geometries and related attributes | Fail | Fail | Fail |
| Estimated elapsed time before first view is rendered in seconds | 1.0s | 1.8s | 1.5s |
| Elapsed time to complete data conversion in seconds | 12.7s | 2.8s | 9.4s |
| File size | IFC 27kt / CityGML 3091kt | IFC 27kt / CityGML 5437kt | IFC 27kt / CityGML 5432kt |

**Table B3** Results part 3 of 4 (Myran).

| Reader | **IFC** | **RVT** | **IFC** |
|---|---|---|---|
| Conversion Software | FME 2017 Workbench | FME 2017 Workbench | ArcGIS Pro Data Interoperability extension (trial) |
| Converted entity ratio | 164/2264 | 165/2245 | 2109/4702 |
| Object to entity centroid standard deviation | xy=1,607015867998819730384889580 6185e-6 h=0,001 | xy=1,607015867998819730384889580 6185e-6 h=0,001 | xy=9.9972e-006, h=0 |
| Conversion accuracy acceptance derivation | 6.5574e+009 | 6.5574e+009 | 6.574e+009 |
| Scale offset factor for LOD or feature | Pass | Pass | Pass |
| XSD validation result | Pass | Pass | Pass |
| If special modelling guidelines apply | Yes | No | No |
| Visual appearance of geometries and related attributes | Pass | Pass | Pass |
| Estimated elapsed time before first view is rendered in seconds | 30.5s | 11.2s | 90s |
| Elapsed time to complete data conversion in seconds | 54.4s | 71.1s | 81.5s |
| File size | IFC 27788kt / CityGML 33093ktkt | IFC 27788kt / CityGML 33219kt | IFC 27788kt / CityGML 102993kt |

| Reader | **RVT** | **IFC** | **RVT** |
|---|---|---|---|
| Conversion Software | ArcGIS Pro Data Interoperability extension (trial) | ArcGIS Pro Data Interoperability extension (trial) FME 2018.1 Workbench | ArcGIS Pro Data Interoperability extension (trial) FME 2018.1 Workbench |
| Converted entity ratio | 1888/2093 | 136/2092 | 164/2578 |
| Object to entity centroid standard deviation | xy=9.9972e-006, h=0 | xy=3,00001666662 0370627570230352 255e-6, h=0 | xy=3,00001666662 0370627570230352 255e-6, h=0 |
| Conversion accuracy acceptance derivation | 6.574e+009 | 6.5574e+006 | 6.5574e+006 |
| Scale offset factor for LOD or feature | Pass | Fail | Fail |
| XSD validation result | Fail | Pass | Pass |
| If special modelling guidelines apply | No | Yes | No |
| Visual appearance of geometries and related attributes | Pass | Fail | Fail |
| Estimated elapsed time before first view is rendered in seconds | 79s | 25.6s | 27.4s |
| Elapsed time to complete data conversion in seconds | 83.6s | 100.4s | 103.2s |
| File size | IFC 27788kt / CityGML 106306kt | IFC 27788kt / CityGML 33155kt | IFC 27788kt / CityGML 35152kt |

**Table B4** Results part 4 of 4 (Up:Town).

| Reader | **IFC** | **RVT** | **IFC** | **IFC** |
|---|---|---|---|---|
| Conversion Software | FME 2017 | ArcGIS Pro Data Interoperability extension (trial) | ArcGIS Pro Data Interoperability extension (trial) | ArcGIS Pro Data Interoperability extension (trial) FME 2018.1 Workbench |
| Converted entity ratio | 20879/46615 | 38431/46615 | 54333/104163 | 17986/46615 |
| Object to entity centroid standard deviation | - | - | - | - |
| Conversion accuracy acceptance derivation | 4.8260e+004 | 5.1915e+004 | 4.8554e+004 | 4.8260e+004 |
| Scale offset factor for LOD or feature | - | - | - | - |
| XSD validation result | No | No | No | No |
| If special modelling guidelines apply | No | No | No | No |
| Visual appearance of geometries and related attributes | - | - | - | - |
| Estimated elapsed time before first view is rendered in seconds | - | - | - | - |
| Elapsed time to complete data conversion in seconds | 3491.6s | 3581.7s | 3050.2s | 3169.7s |
| File size | IFC 246824kt / CityGML 1160765kt | IFC 246824kt / CityGML 1551832kt | IFC 246824kt / CityGML 1422229kt | IFC 246824kt / CityGML 1111980kt |

# Appendix C

## Pictures related to conversion results

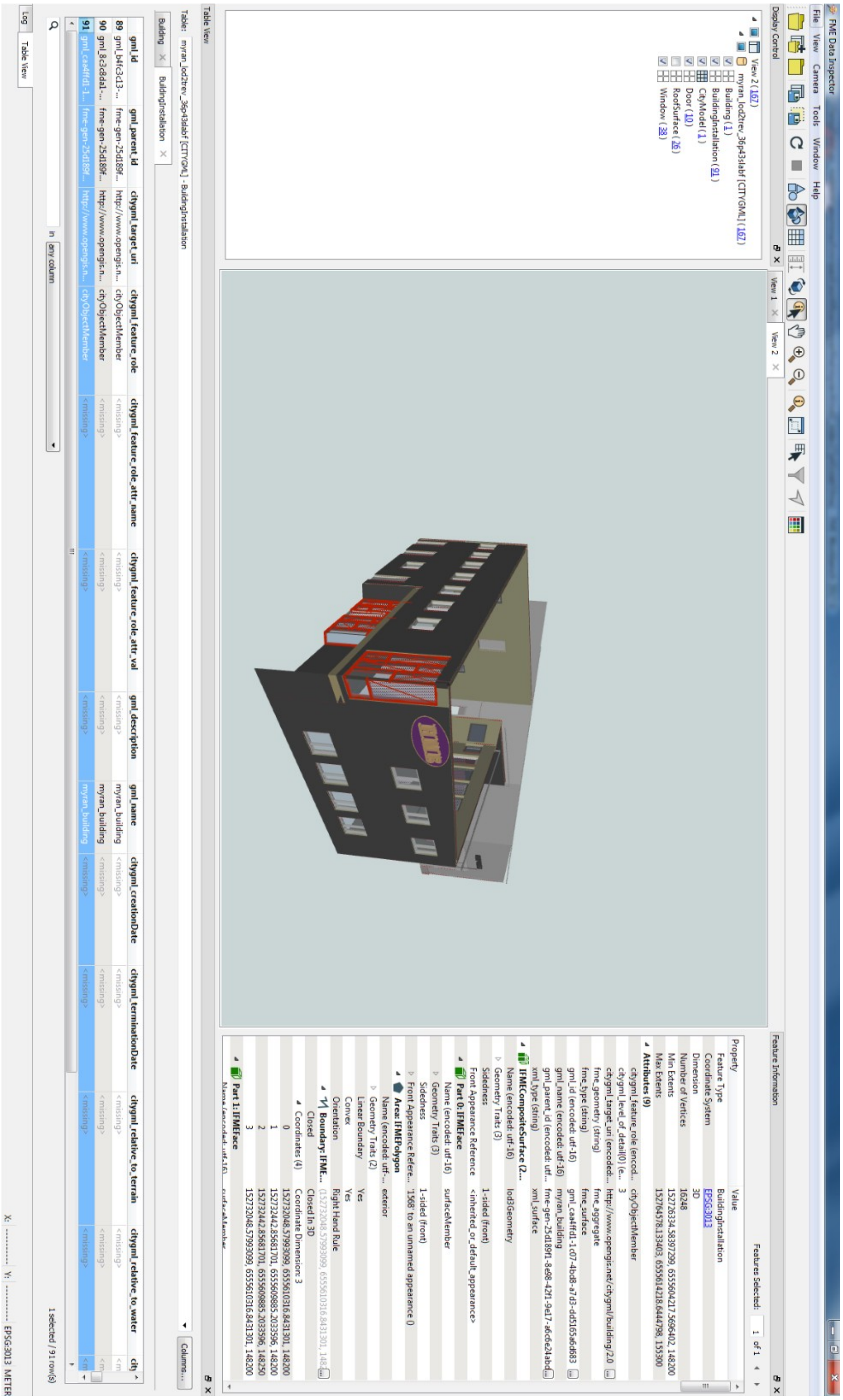
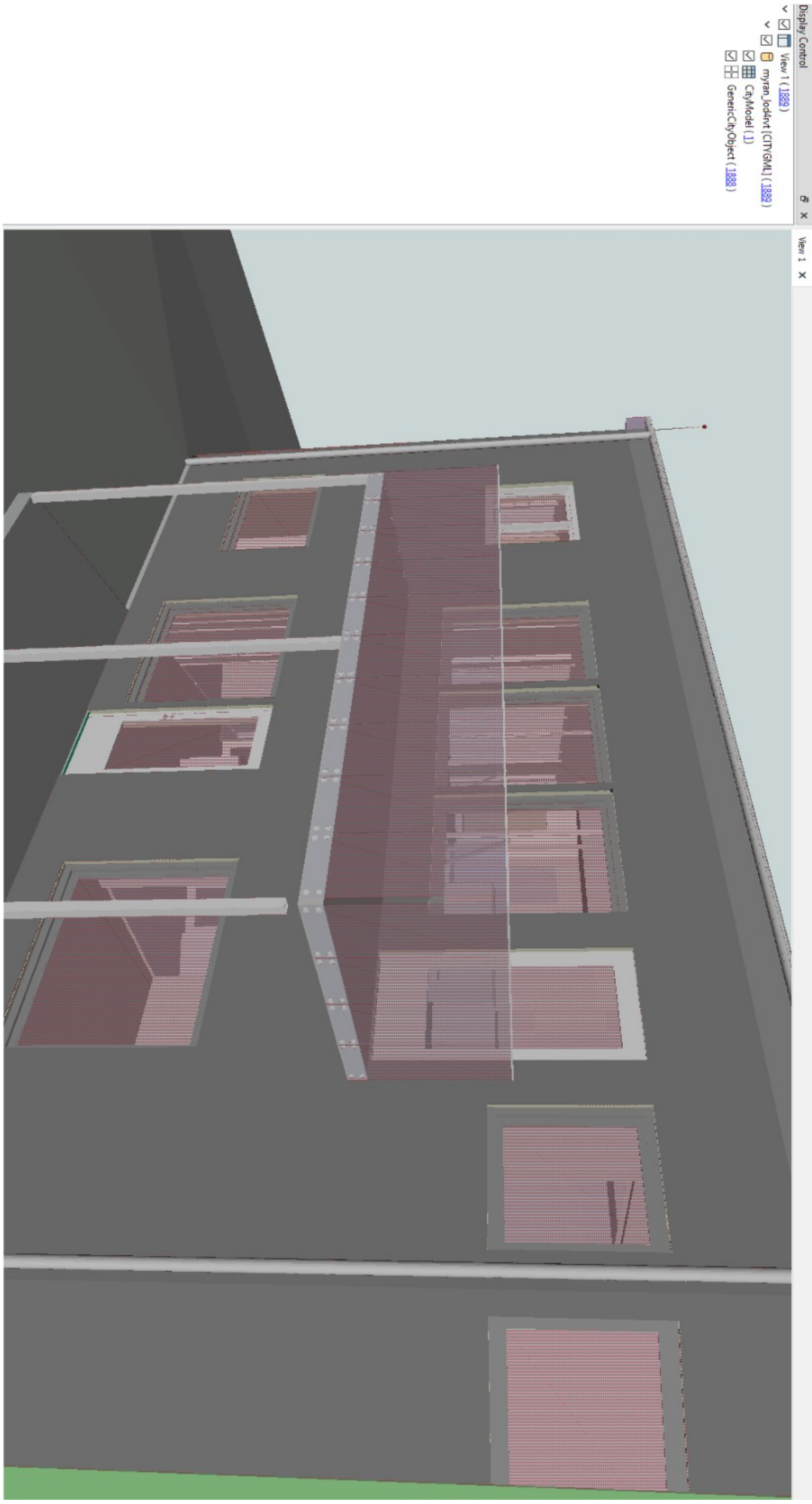


Figure C1: Unified Frames from a test conversion workflow.


Display Control
View 1 ( 1889 )
myran_lod4rvt [CITYGML] ( 1889 )
CityModel ( 1 )
GenericCityObject ( 1888 )
View 1

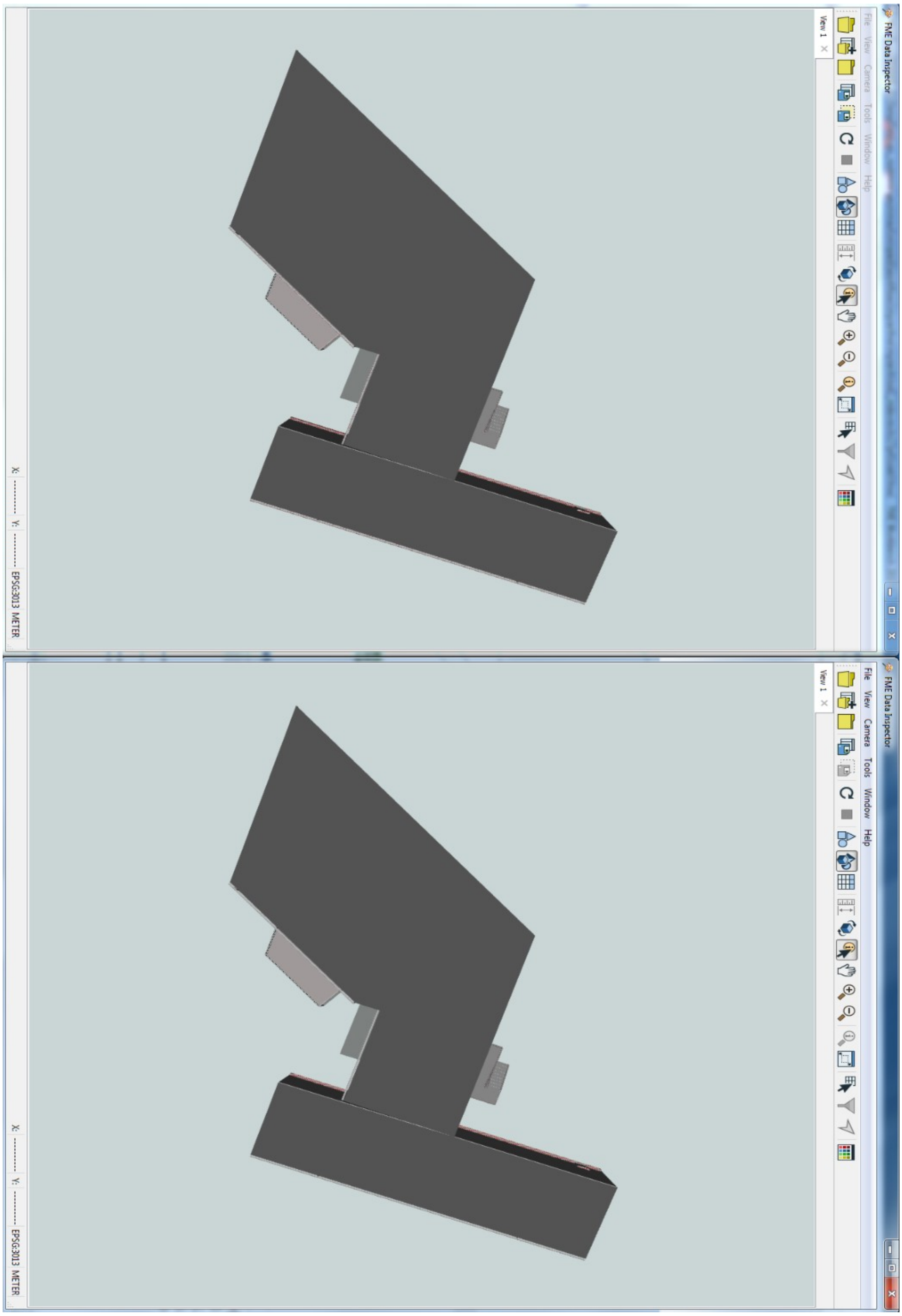
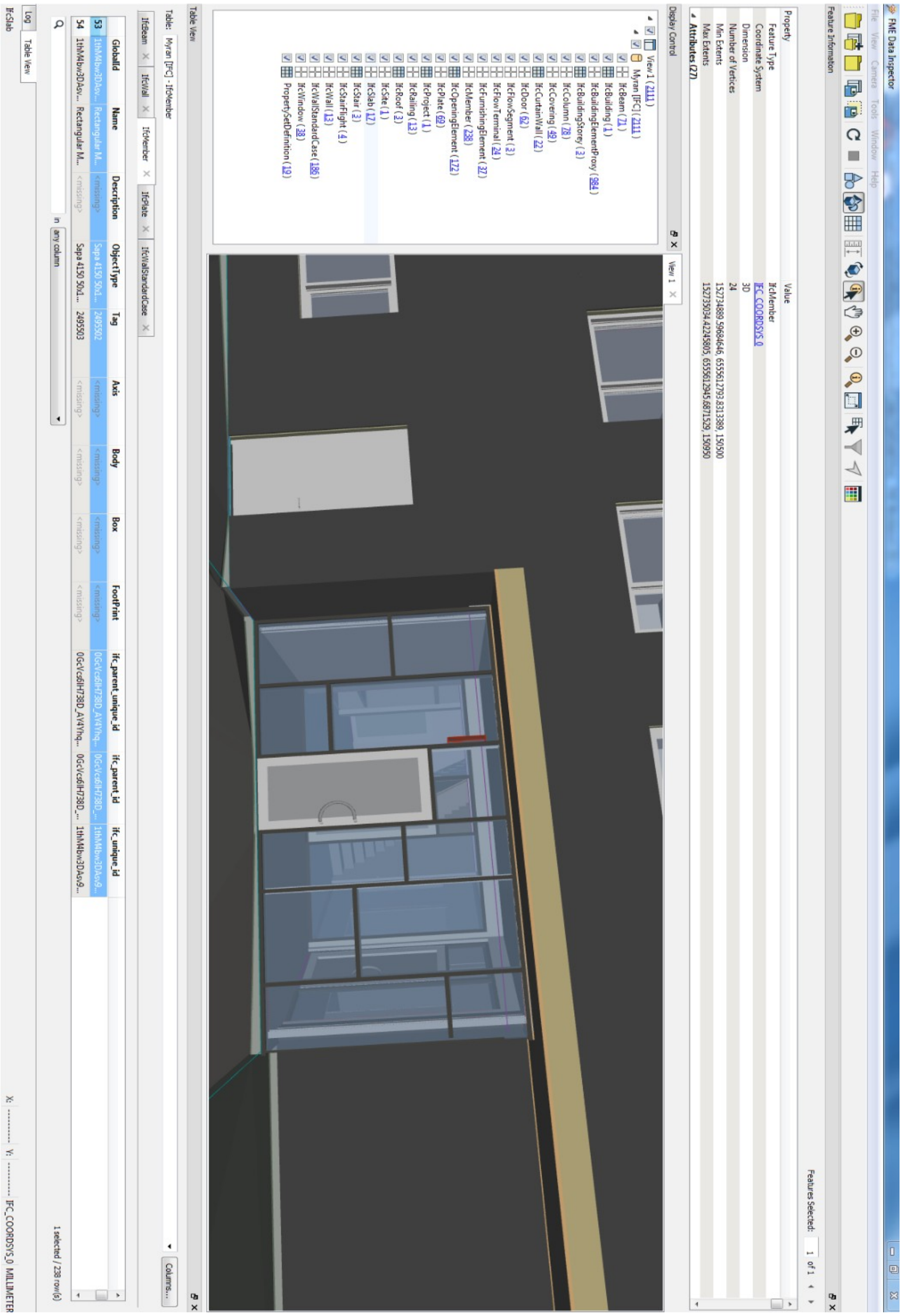


Figure C2: Myran results from FME 2017 workflows (left, IFC reader; right, RVT reader).

Figure C3: Myran results from FME 2017 workflows (left, IFC reader; right, RVT reader).


FME Data Inspector
Feature Information
Display Control
Table View

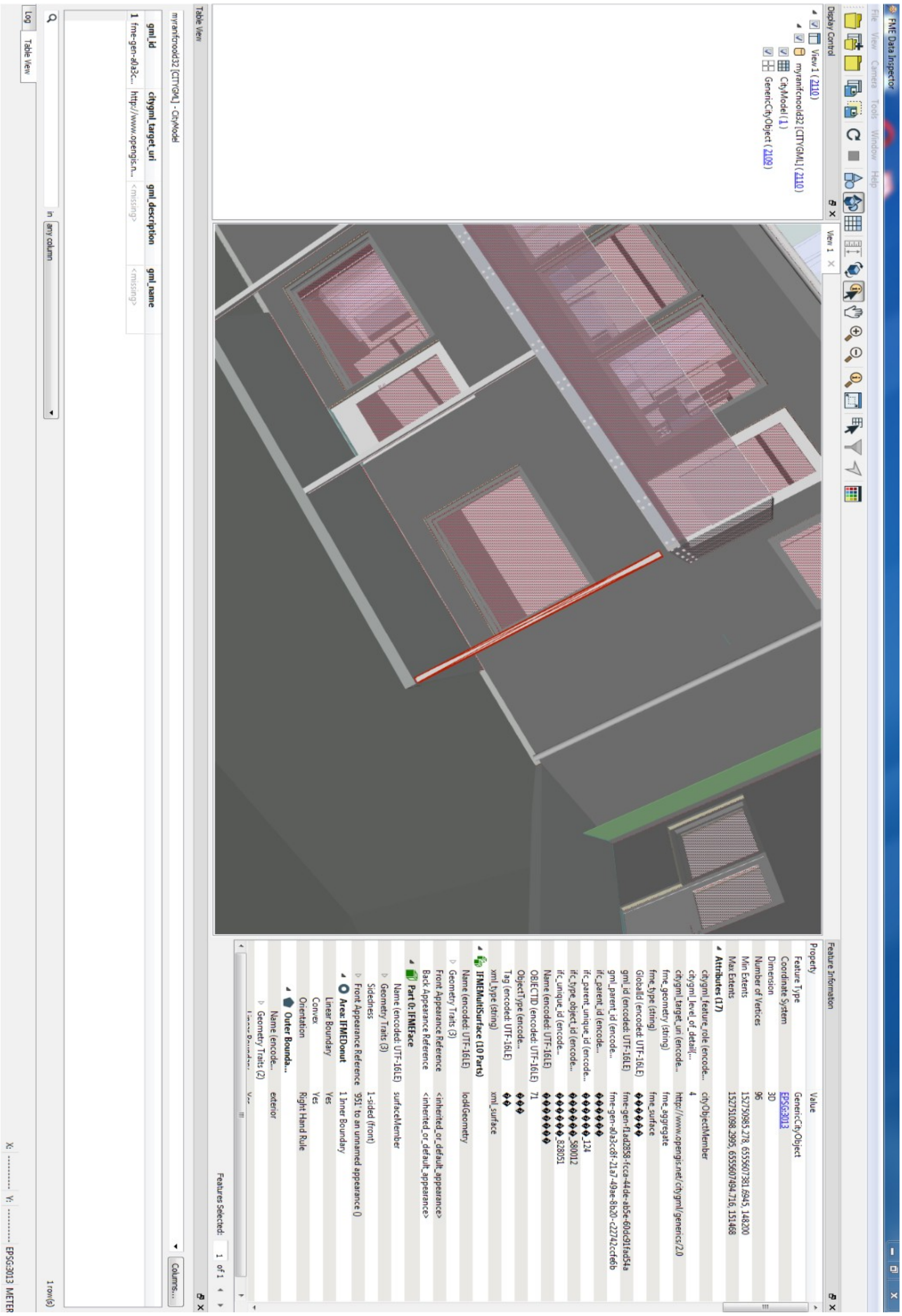


Figure C4: There are some weird entities in the IFC files. These entities are not errors in the CityGML data.

Figure C5: LOD4 data from ArcGIS PRO extension using RVT data mappings.

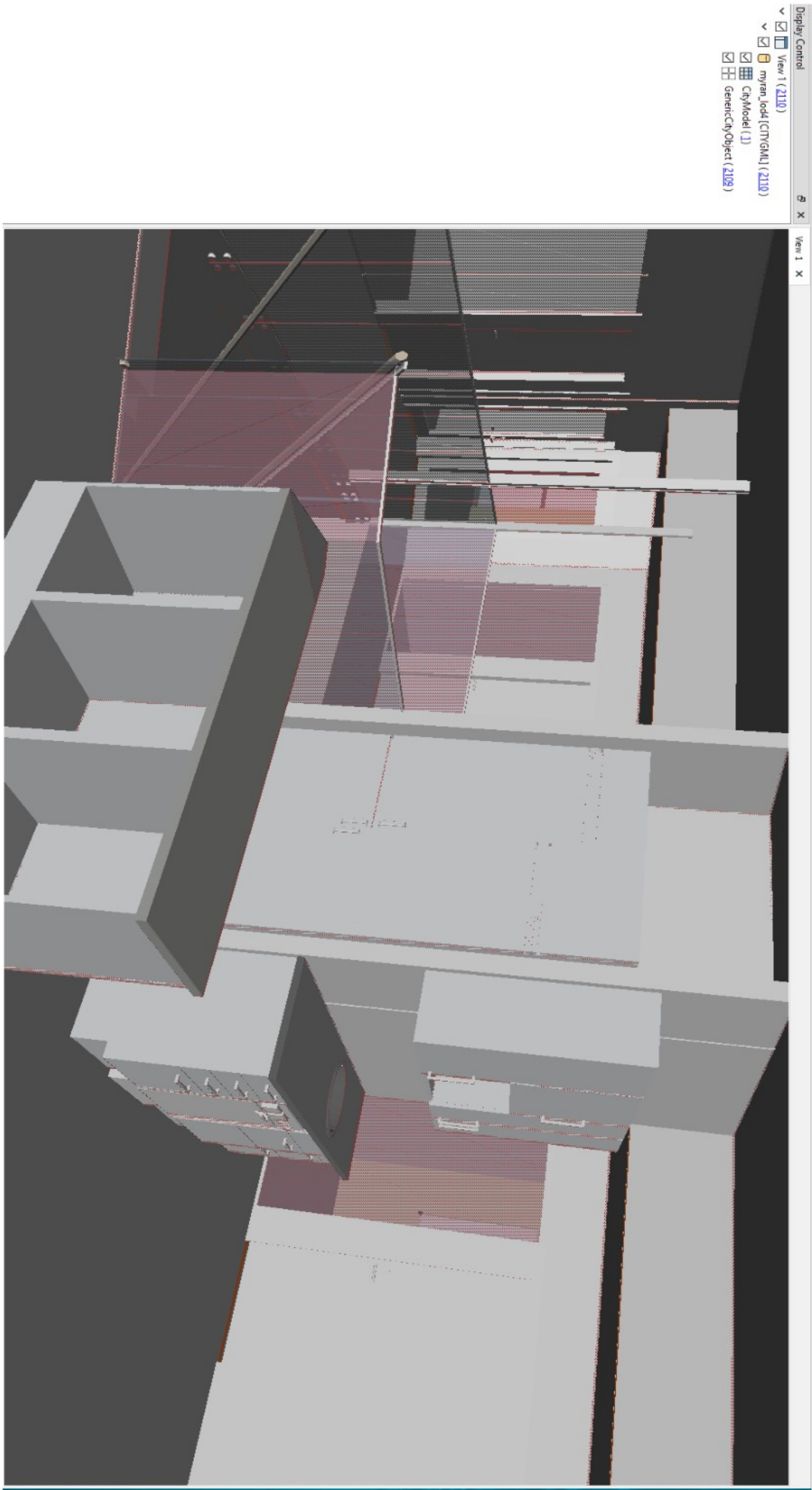


Figure C6: Myran LOD4 IFC reader result comparison.

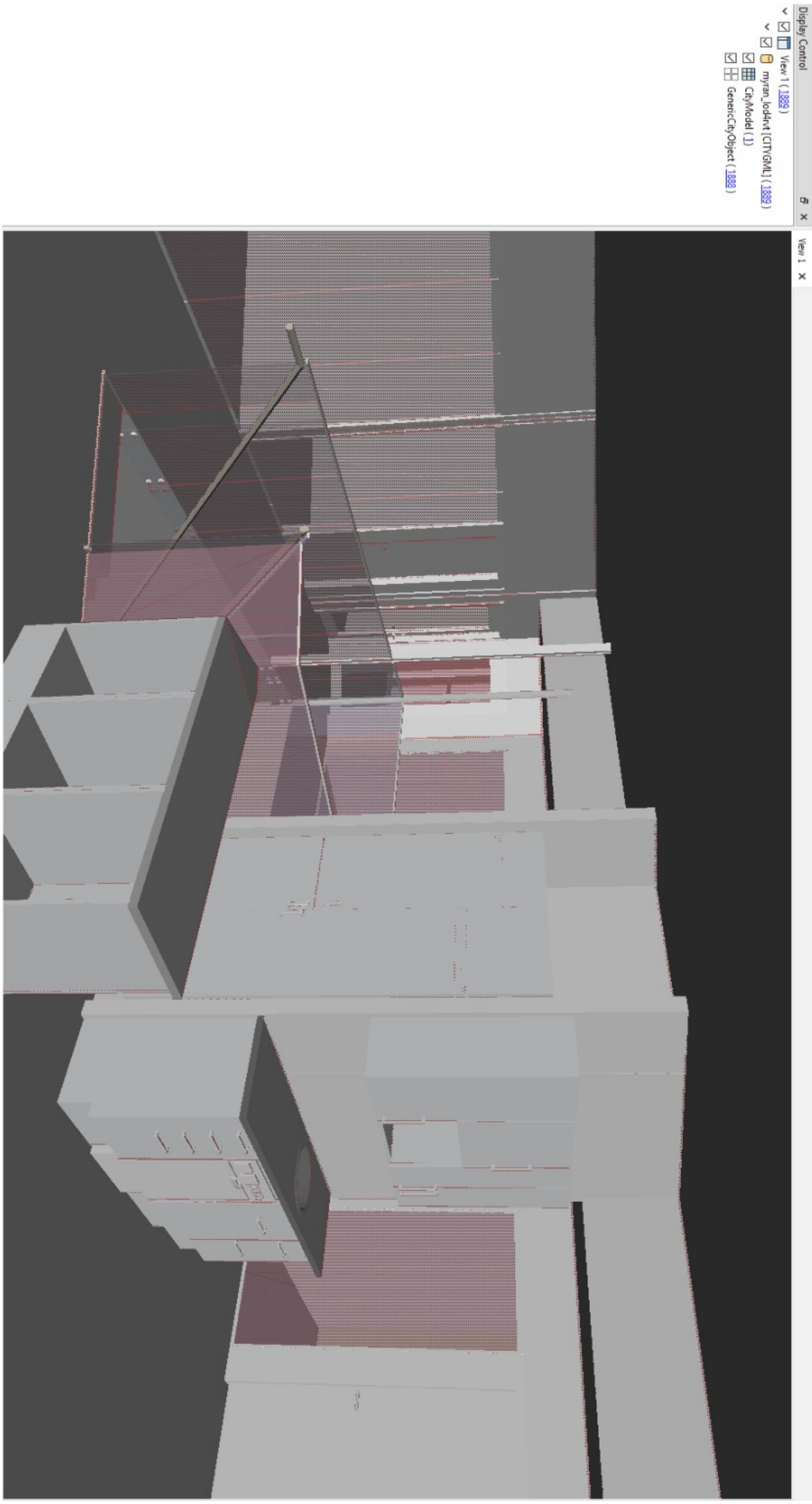



Figure C7: Myran LOD4 RVT reader result comparison.

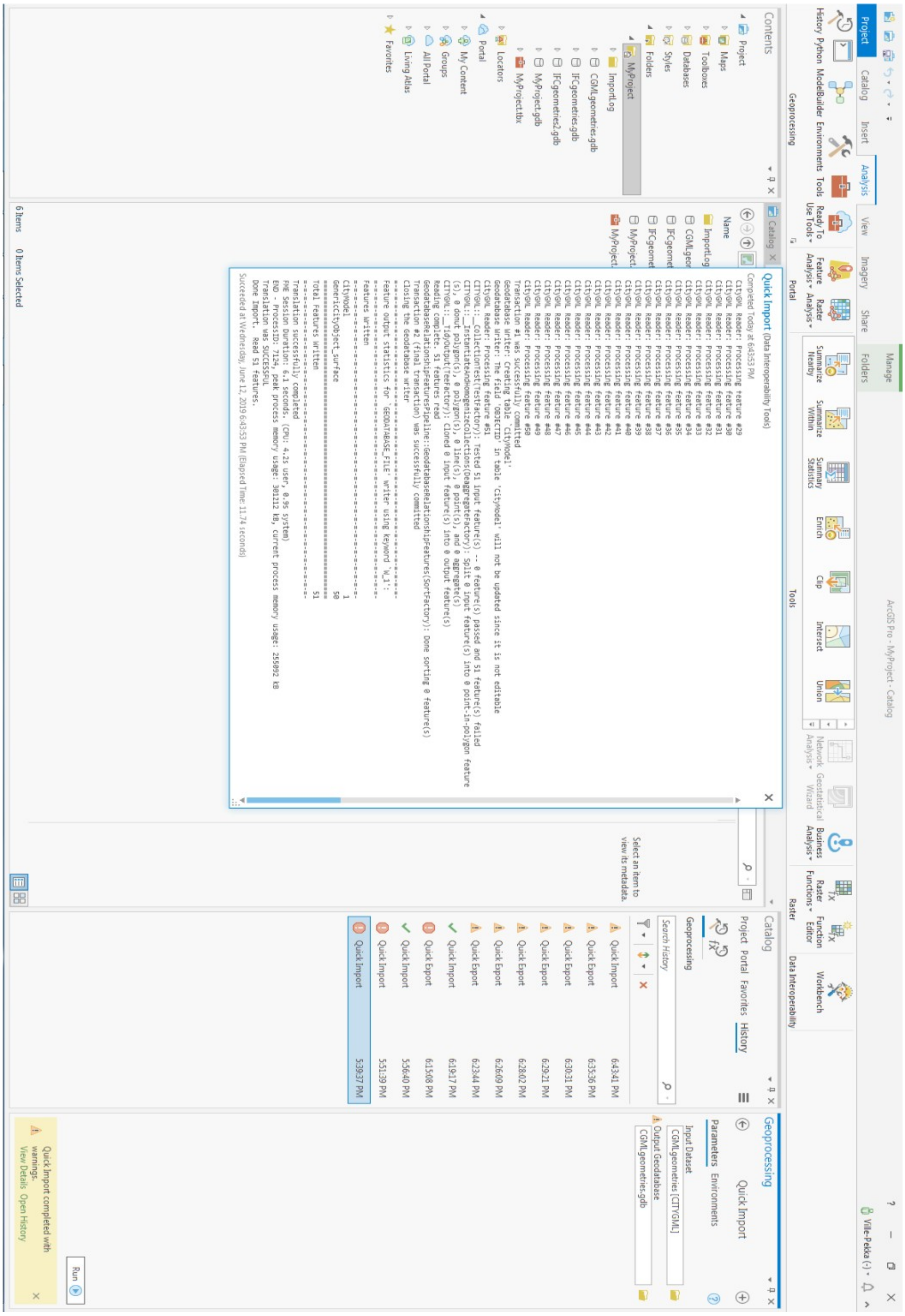


Figure C8: The conversion of IFC entities into *GenericCityObjects* is dependent on how the features are read into the .gdb.

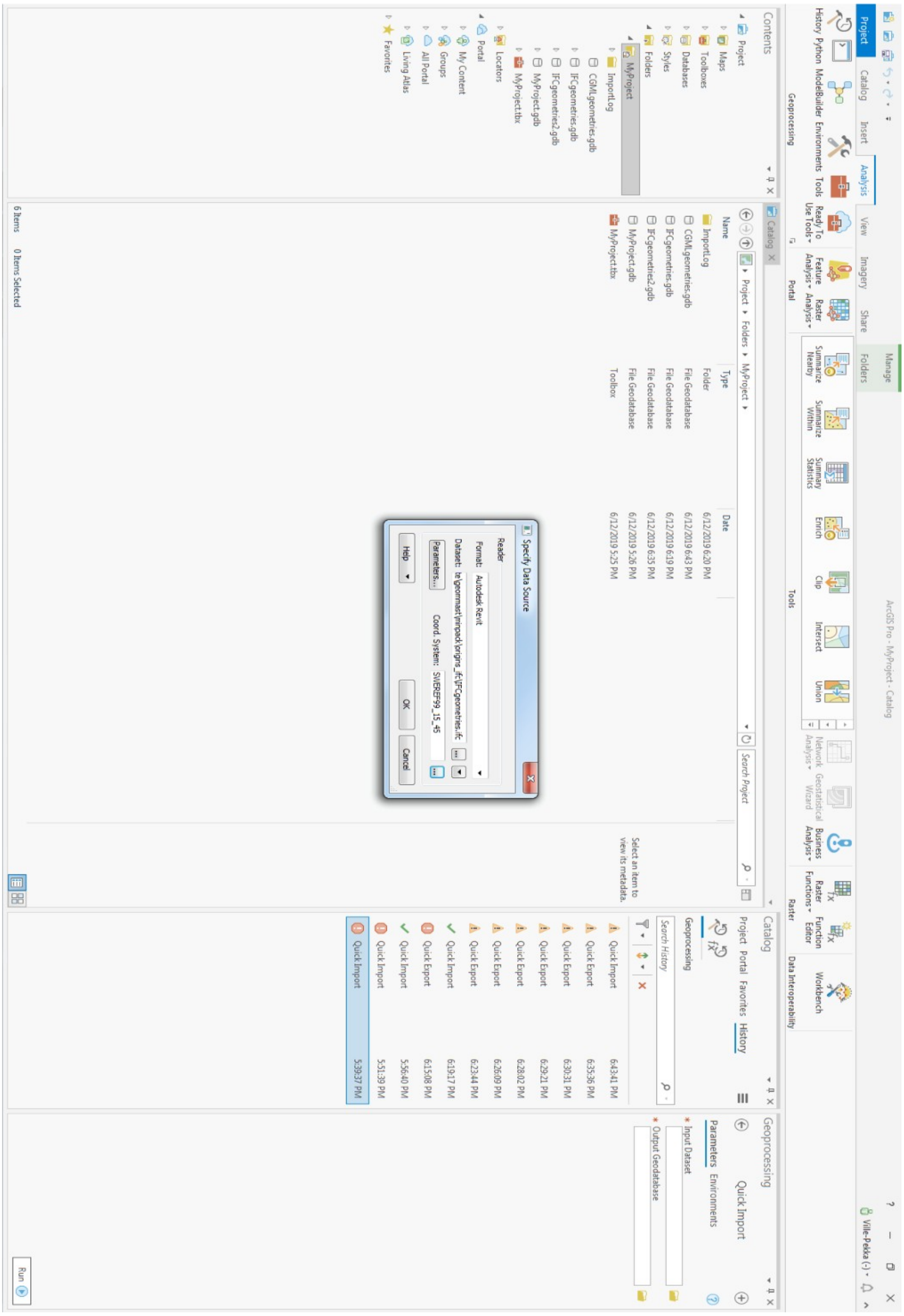


Figure C9: The conversion of IFC entities into *GenericCityObjects* is dependent on how the features are read into the .gdb (2).

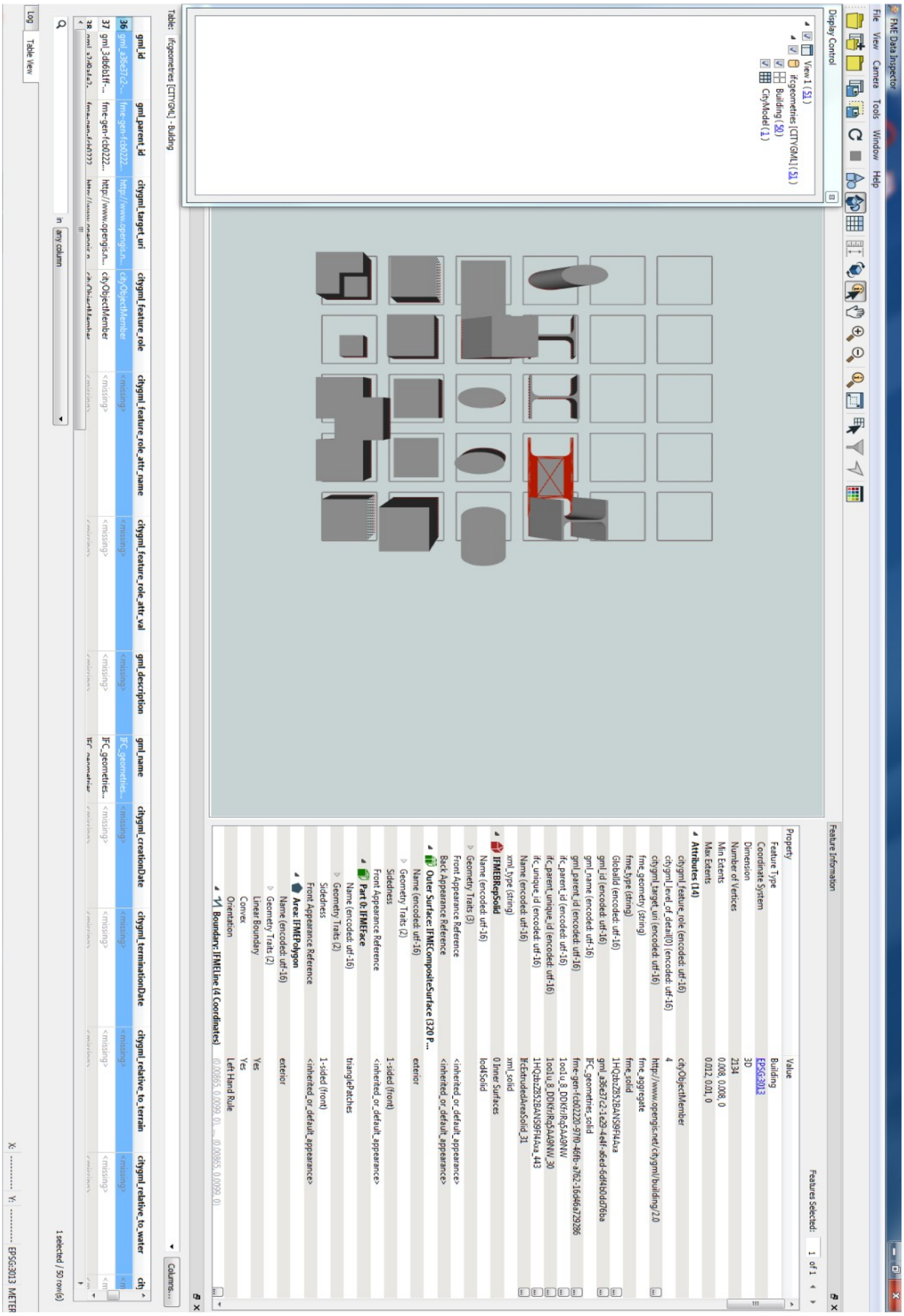


Figure C10: IFC reader results for IFC 2x3.

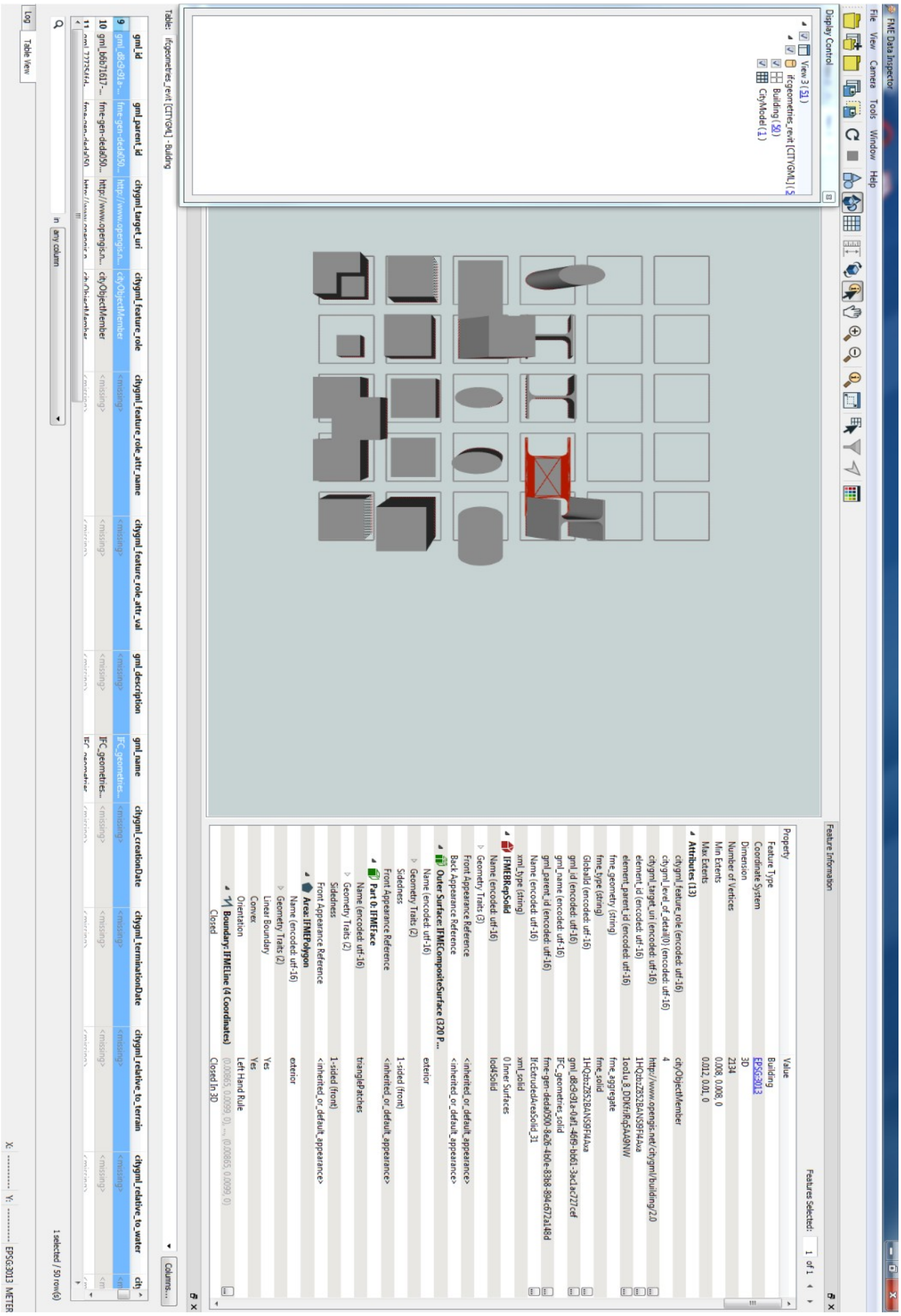


Figure C11: RVT reader results for IFC 2x3.

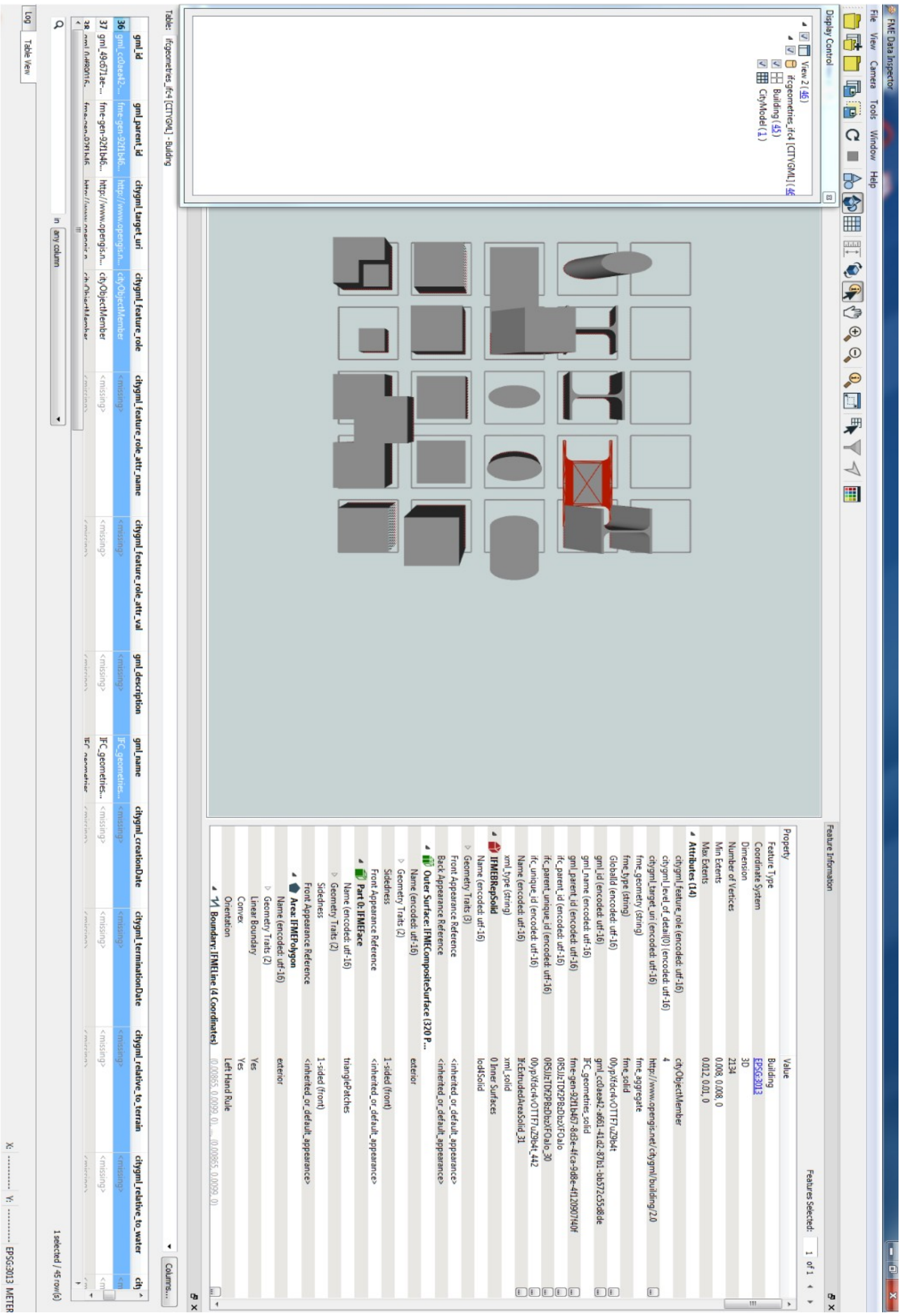


Figure C12: IFC reader for IFC4.

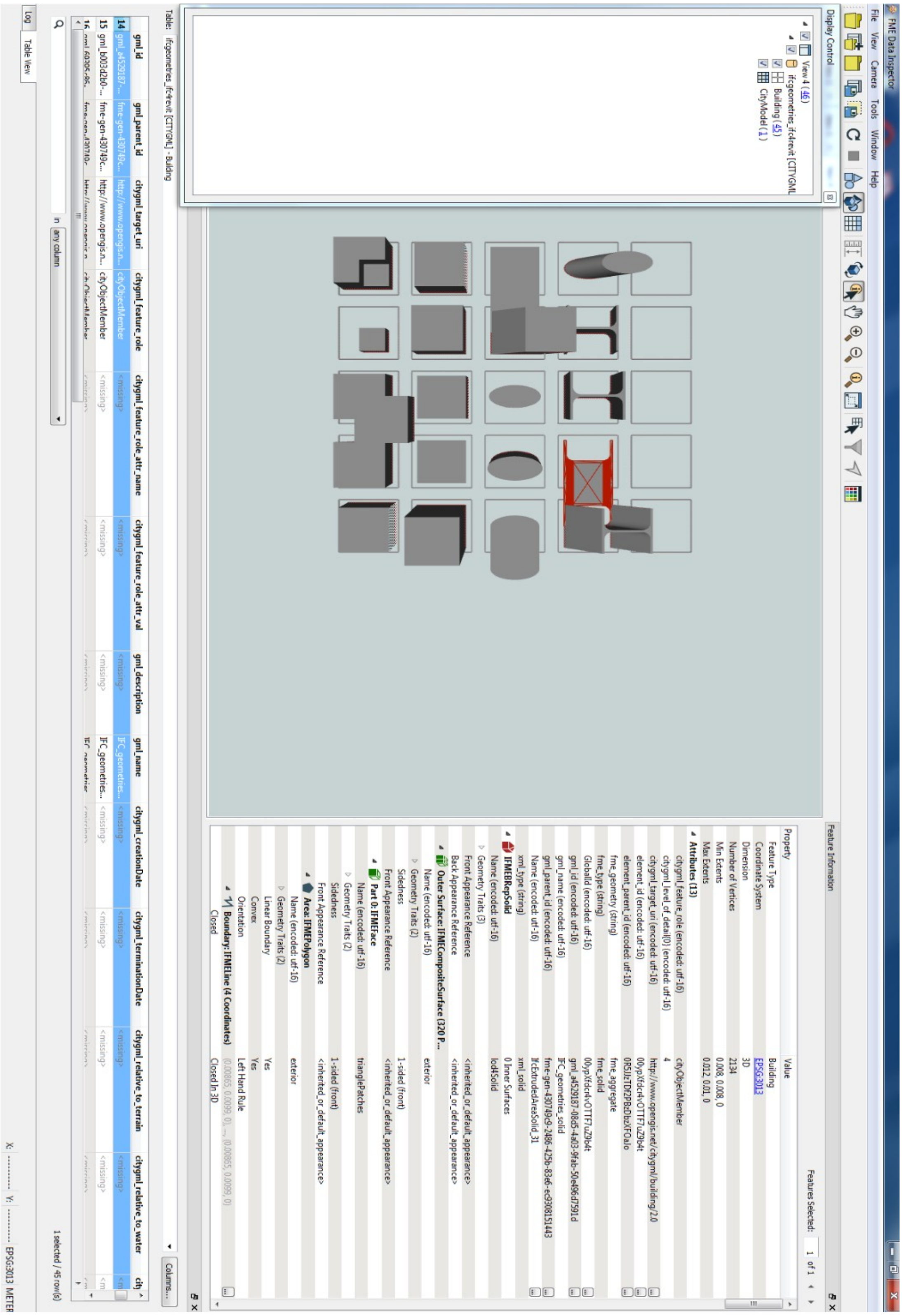


Figure C13: IFC reader for IFC4.