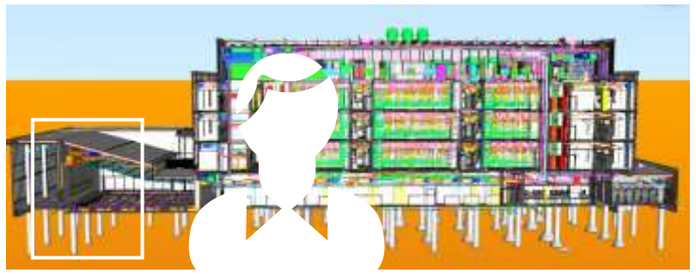
Overview of GeoBIM project

Jantien Stoter, Ken Arroyo Ohori, Thomas Krijnen,
Hugo Ledoux,

GeoBIM conference
Amsterdam 24 November 2017

GeoBIM:

We want to integrate because we model the same real world, but



3D Geo

Differences

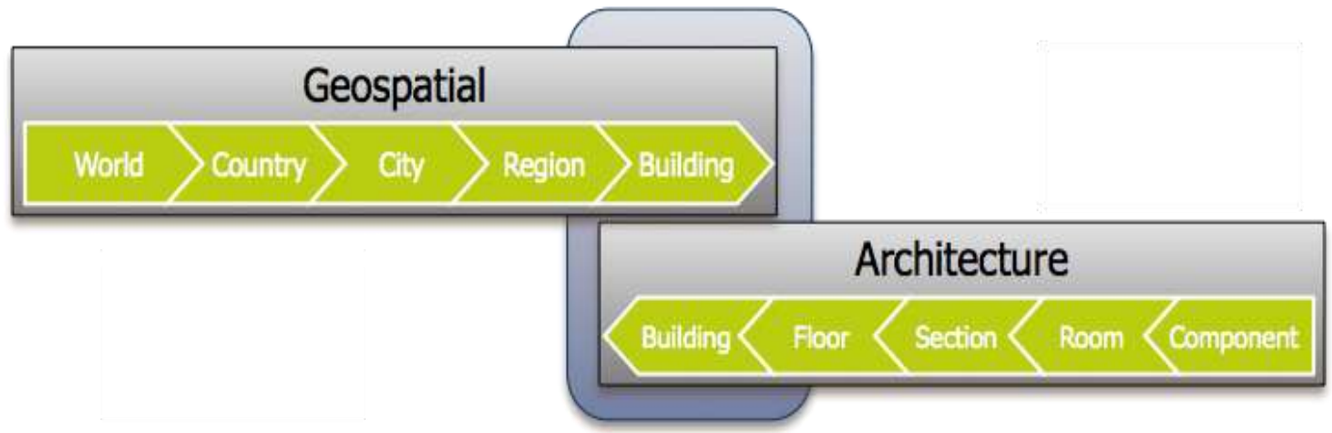
Spatial data is a digital plan for design and construction	Spatial data is source of inf on
Industry dominated	Government dominated
Sharing data difficult (copy rights); benefits for sharing are not always clear	Open data and sharing data is common aim (both sender and receiver benefit)
Focus on data possibilities in native software	Attention for data quality, validation and data responsibilities; data flows outside software
Site based data management	Theme based data management; integration of data from different sources
Data is designed (parametrized)	Data is measured (Brep)
Detailed (mm precision)	Less detailed (cm/m precision)

GeoBIM

frequently used term

But not a common understanding

- Strict boundary?



- What is GeoBIM?:
  - Viewing/querying both in one environment? Data integration? Harmonising two standards?
  - Data conversion and import? 3D?

- What are GeoBIM applications?
  - Urban planning, geology, asset & life cycle management, 3D cadastre, energy?

- What are the GeoBIM issues to solve?
  - geometry, semantics, links, import/export, very detailed versus global, measured versus designed

Our GeoBIM project

TU Delft — Delft University of Technology

TU/e — Technische Universiteit Eindhoven University of Technology

Gemeente Rotterdam

Gemeente Den Haag

kadaster

Rijkswaterstaat

GEONOVUM

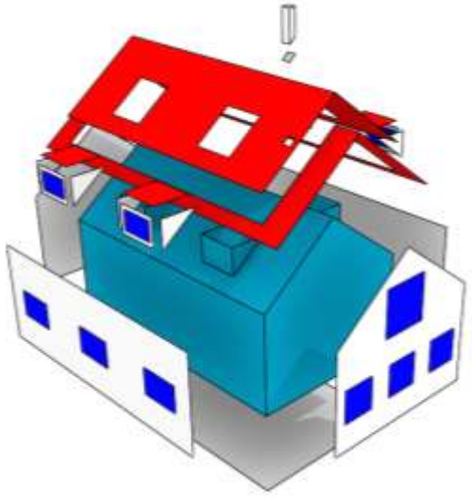
TNO innovation for life

BIM Loket

Our goal

Develop an interface between the two worlds to prepare for a fundamental solution to bridge the gap

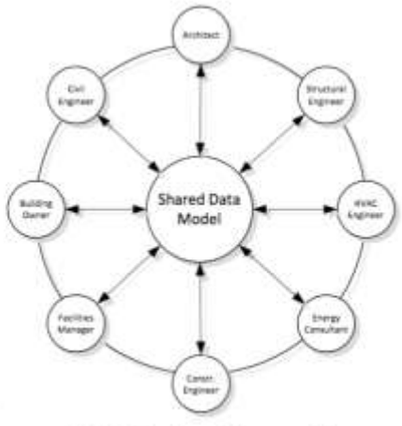
1. Open-source API to represent IFC (BIM) + CityGML (Geo) with the same data structure
2. Recommendations for future integration



CGAL

\+

(A) buildingSMART standards for BIM



(B) IFC, a shared data model

IfcOpenShell

the open source ifc toolkit and geometry engine

3DGIS==CityGML

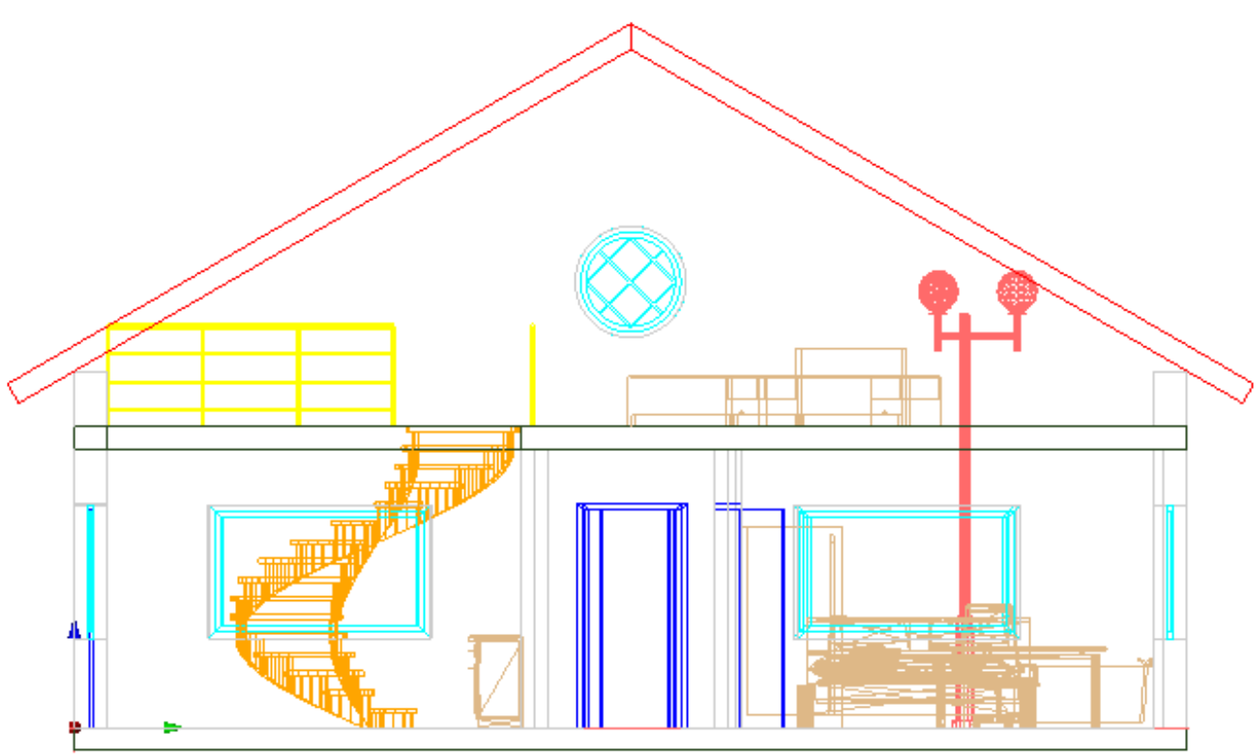
Bridging the gaps between IFC and CityGML

unlike most other similar initiatives, **geometry** is our main concern

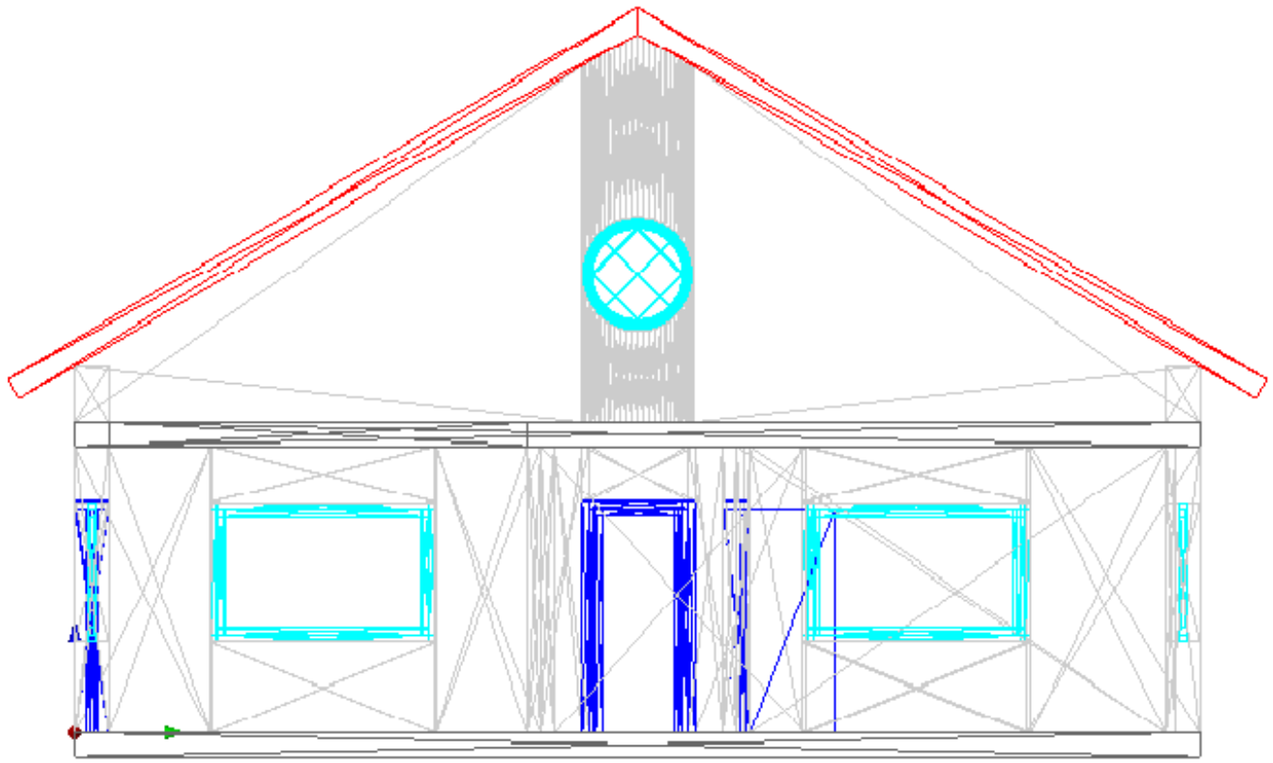
1. gap in semantic
2. gap in geometry/topology
3. gap in coordinate reference systems

Results with current commercial software

most geometries are converted



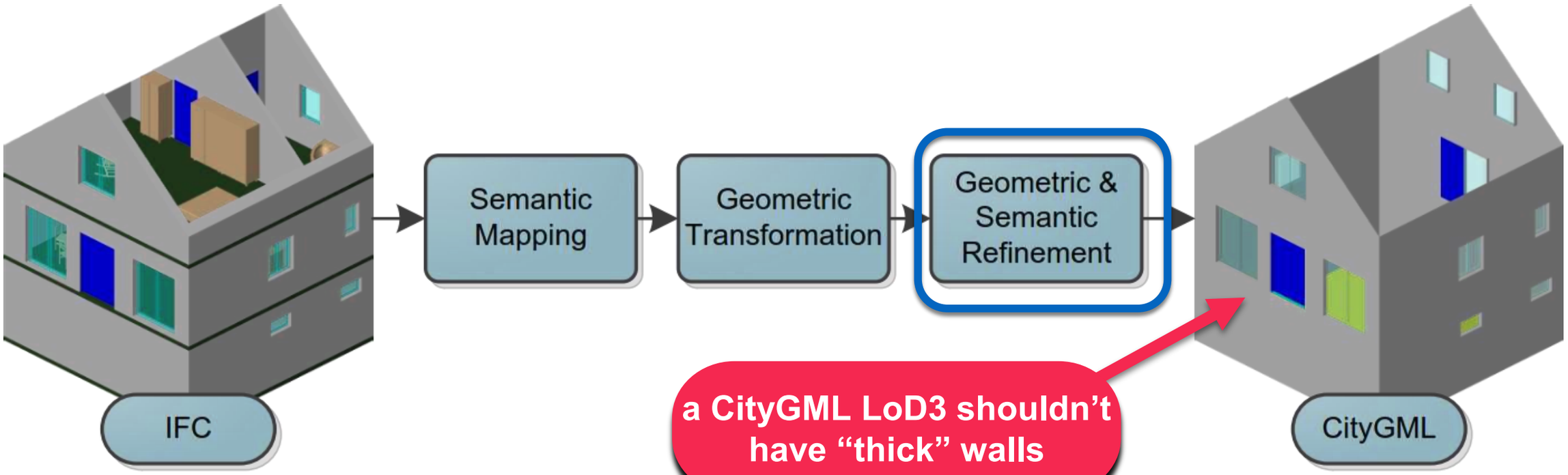
IFC

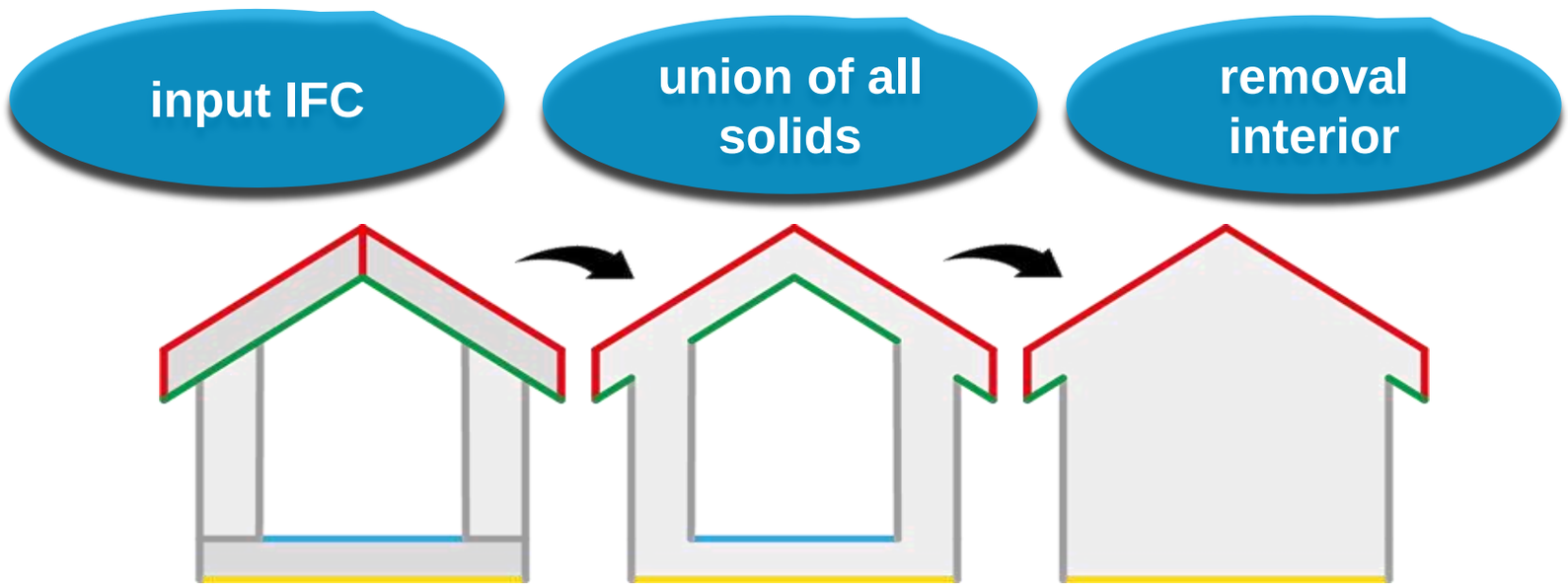


CityGML

(Donkers, 2013)

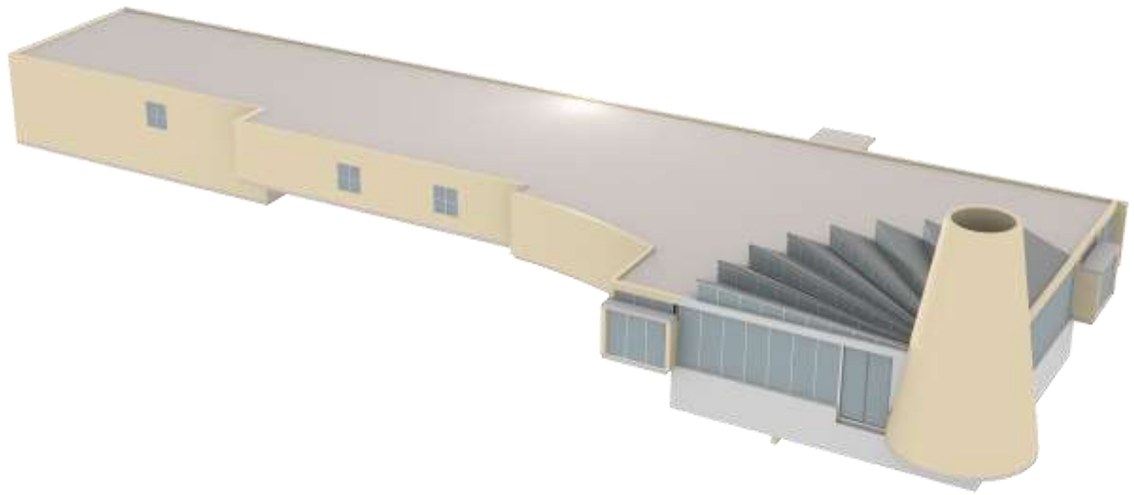
Our conversion methodology

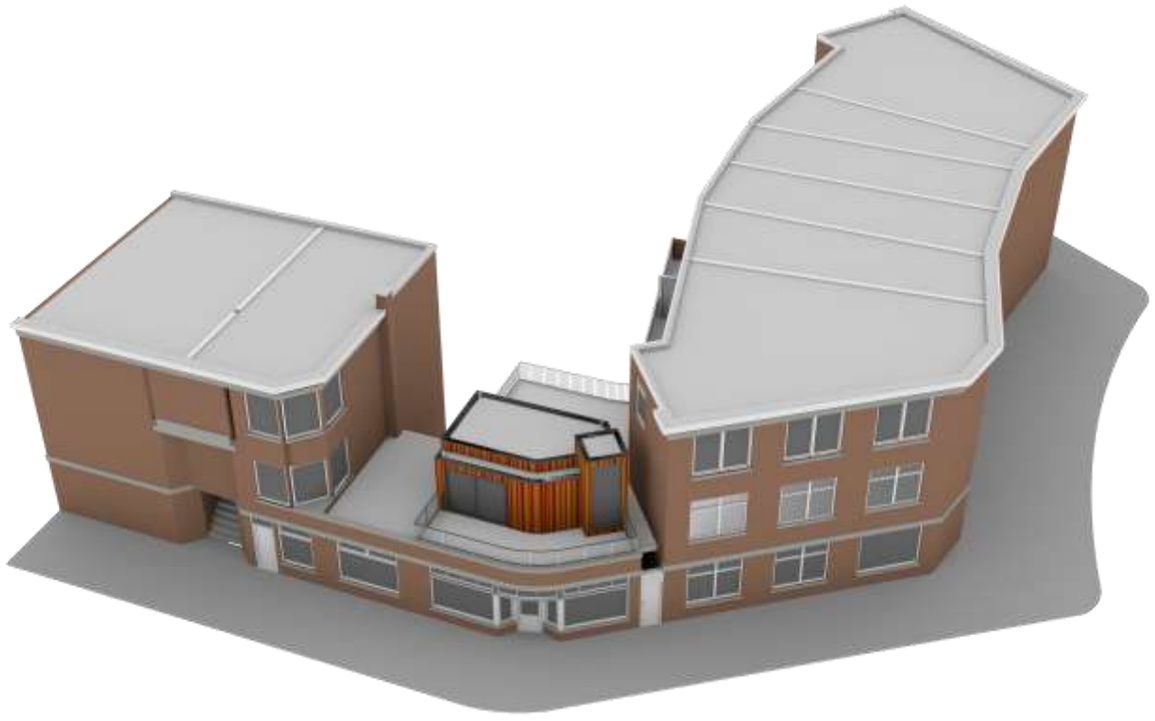




Sjors Donkers, Hugo Ledoux, Junqiao Zhao, Jantien E. Stoter: Automatic conversion of IFC datasets to geometrically and semantically correct CityGML LOD3 buildings. Trans. GIS 20(4): 547-569 (2016)

Developed and tested our approach on newly designed buildings in NL (no academic data)





Main conclusion 1

IFC models are rigged with errors, and we don’t see this improving soon…

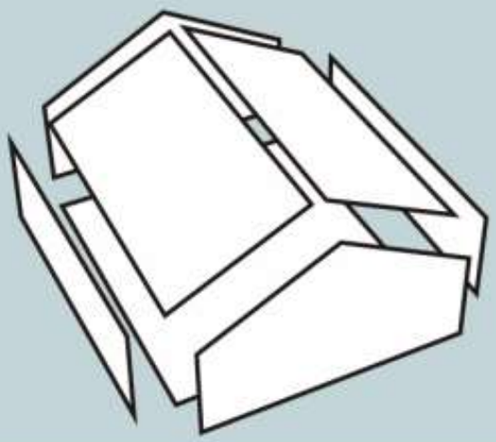
Data interest designers/constructors is minor (not part of their job)

(not clear were errors are created: in native software or IC export)

Representations of the geometry

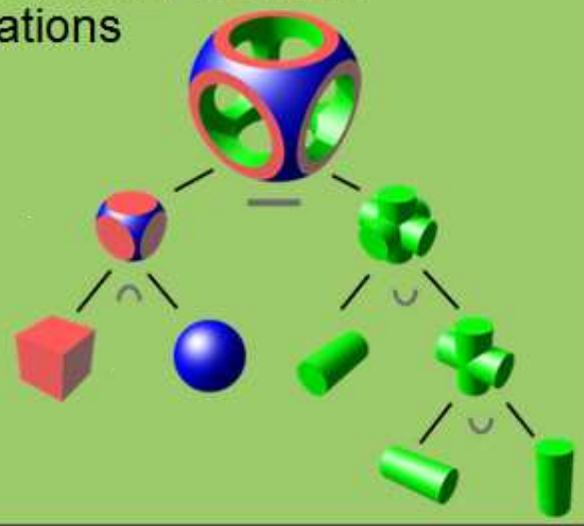
Boundary Representation

Aggregation of boundary surfaces, which enclose the body completely



Constructive Solid Geometry

- 3D primitives
- Combined by boolean operations



Sweep Volume

Extrusion of a face along a defined path

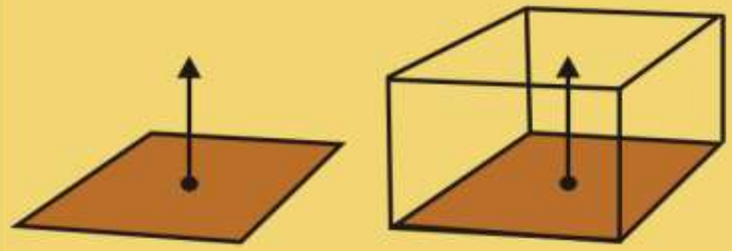


Figure adapted from Kolbe and Plümer (2

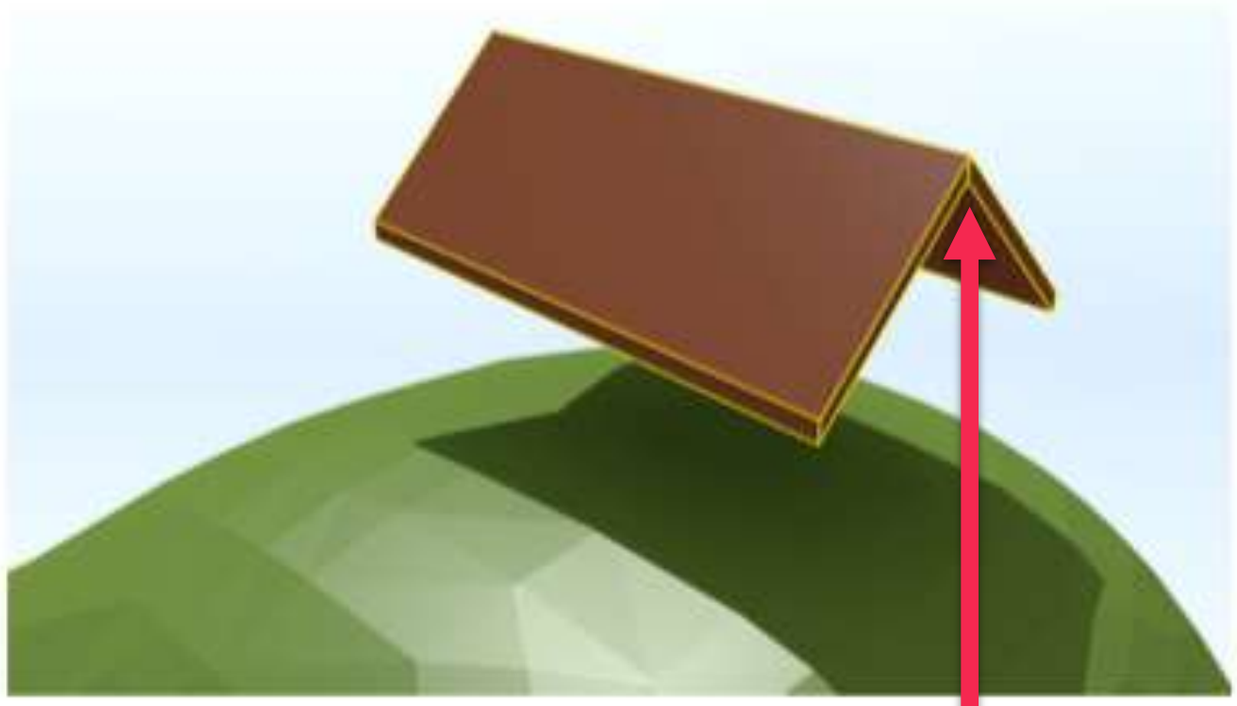
explicit representation

implicit representation
(need to be discretised to be manipulated with GIS objects)

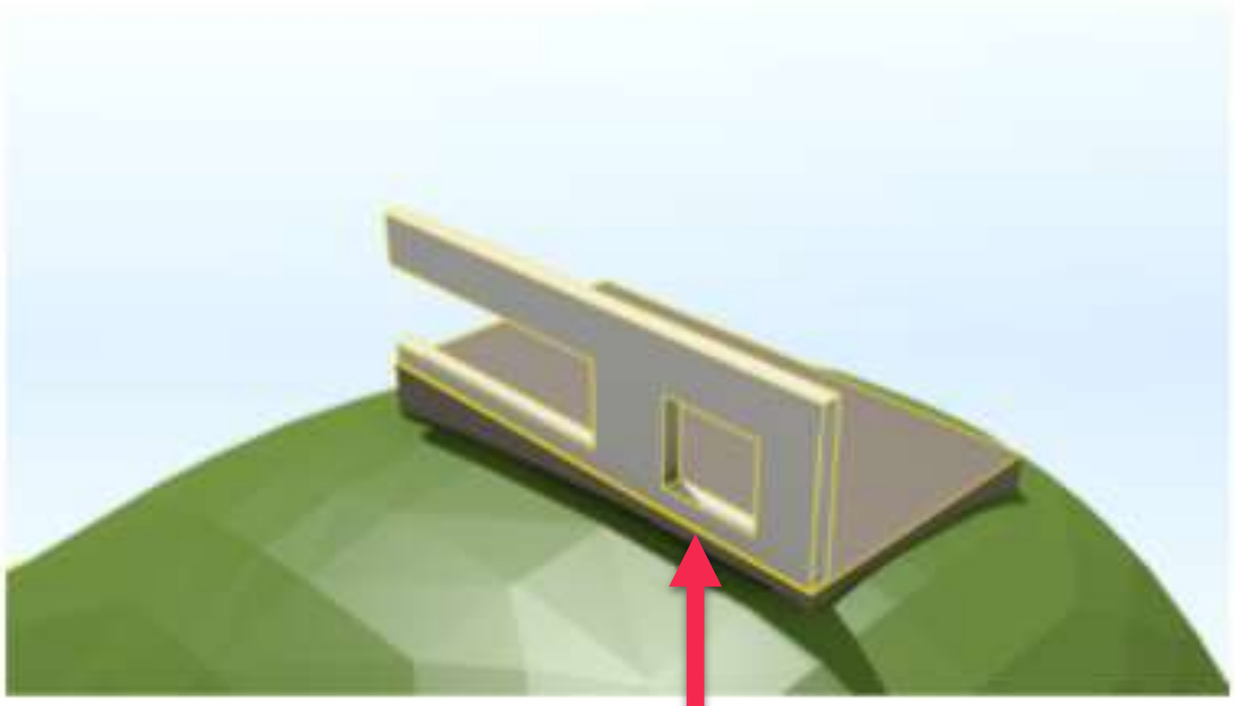
3D GIS

IFC

IFC models = CSG computations that can go bad…



2 roof parts do not touch



wall and ground overlap

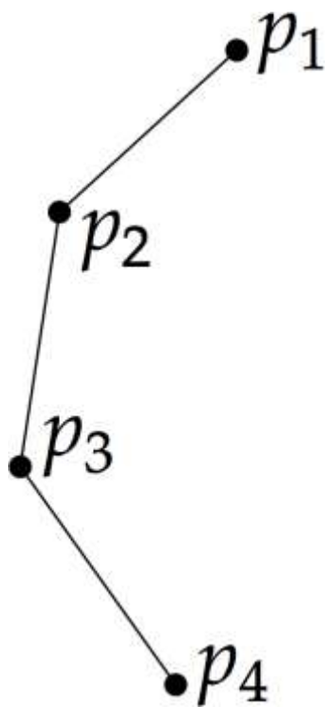
Main conclusion 2

IFC has many (geometry) classes

More than 1 way to model a situation & transform a situation

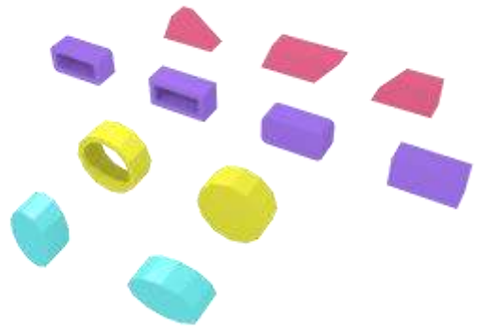
It is not possible to develop transformation for:
any class and any modelling situation

Curves/wires

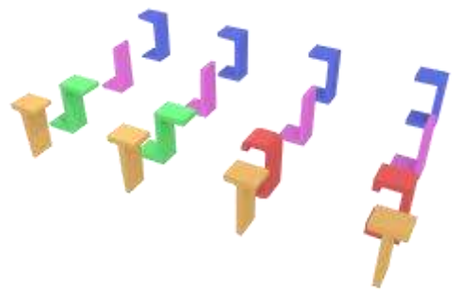


IfcCircle
IfcEllipse
IfcLine
IfcEdge
IfcOrientedEdge
IfcEdgeLoop
IfcPolyLoop
IfcPolyline
IfcCompositeCurve
IfcTrimmedCurve

Faces

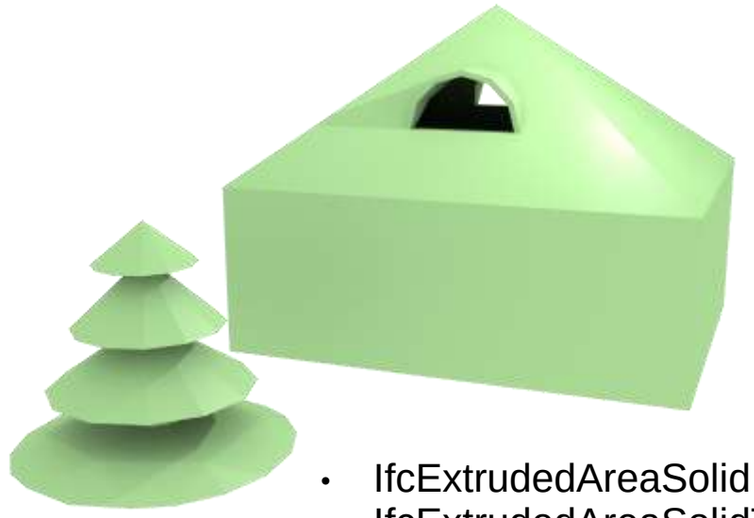


- IfcArbitraryClosedProfileDef
- IfcArbitraryProfileDefWithVoids
- IfcRectangleProfileDef
- IfcRoundedRectangleProfileDef
- IfcRectangleHollowProfileDef
- IfcTrapeziumProfileDef
- IfcCircleProfileDef
- IfcCircleHollowProfileDef
- IfcEllipseProfileDef
- IfcFace



IfcCShapeProfileDef
IfcLShapeProfileDef
IfcIShapeProfileDef
IfcTShapeProfileDef
IfcUShapeProfileDef
IfcZShapeProfileDef
IfcDerivedProfileDef

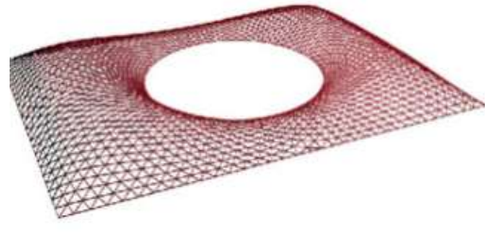
Volumetric shapes



- IfcExtrudedAreaSolid
- IfcExtrudedAreaSolidTapered
- IfcConnectedFaceSet
- IfcCsgSolid
- IfcBlock
- IfcBooleanResult
- IfcSphere
- IfcRectangularPyramid
- IfcRightCircularCylinder
- IfcRightCircularCone
- IfcTriangulatedFaceSet
- IfcHalfSpaceSolid

Abstract shapes

IfcRepresentation
IfcGeomatricSet
IfcShellBasedSurfaceModel
IfcManifoldSolidBrep
IfcMappedItem
IfcFaceBasedSurfaceModel



IFC (Industry Foundation Classes)

IfcActuatorType
IfcAirTerminalBoxType
IfcAirTerminalType
IfcAirToAirHeatRecoveryType
IfcAlarmType
IfcAnnotation
IfcBeam
IfcBoilerType
IfcBuildingElementPart
IfcBuildingElementProxy
IfcBuildingStorey
IfcCableCarrierFittingType
IfcCableCarrierSegmentType
IfcCableSegmentType
IfcChillerType
IfcCoilType
IfcColumnType
IfcCompressorType
IfcCondenserType
IfcControllerType
IfcCooledBeamType
IfcCoolingTowerType
IfcCovering
IfcCurtainWall
IfcDamperType
IfcDistributionChamberElementType
IfcDistributionControlElement
IfcDistributionElement
IfcDistributionFlowElement
IfcDoorType
IfcDuctFittingType
IfcDuctSegmentType
IfcDuctSilencerType
IfcElectricApplianceType
IfcElectricFlowStorageDeviceType
IfcElectricGeneratorType
IfcElectricHeaterType
IfcElectricMotorType
IfcElectricTimeControlType
IfcElementAssembly

IfcEnergyConversionDevice
IfcEvaporativeCoolerType
IfcEvaporatorType
IfcFanType
IfcFastenerType
IfcFilterType
IfcFireSuppressionTerminalType
IfcFlowController
IfcFlowFitting
IfcFlowInstrumentType
IfcFlowMeterType
IfcFlowMovingDevice
IfcFlowSegment
IfcFlowStorageDevice
IfcFlowTerminal
IfcFlowTreatmentDevice
IfcFooting
IfcFurnishingElement
IfcFurnitureType
IfcGasTerminalType
IfcHeatExchangerType
IfcHumidifierType
IfcJunctionBoxType
IfcLampType
IfcLightFixtureType
IfcMechanicalFastenerType
IfcMemberType
IfcMotorConnectionType
IfcOpeningElement
IfcOutletType
IfcPile
IfcPipeFittingType
IfcPipeSegmentType
IfcPlateType
IfcProtectiveDeviceType
IfcPumpType
IfcRailing
IfcRamp
IfcReinforcingBar
IfcReinforcingMesh

IfcRoof
IfcSanitaryTerminalType
IfcSensorType
IfcSite
IfcSlab
IfcSpace
IfcSpaceHeaterType
IfcStackTerminalType
IfcStair
IfcSwitchingDeviceType
IfcSystemFurnitureElementType
IfcTankType
IfcTransformerType
IfcTransportElementType
IfcTubeBundleType
IfcUnitaryEquipmentType
IfcValveType
IfcWall
IfcWasteTerminalType
IfcWindowType

1000+ in total

OGC®
Making location count.

Future City Pilot-1: Using IFC/CityGML in Urban Planning Engineering Report

Publication Date: 2016-10-03

Approval Date: 2017-08-17

Posted Date: 2017-06-27

Reference number of this document: OGC 16-097

Reference URL for this document: http://www.opengis.net/doc/PER/FCP1-UPrules

Category: Public Engineering Report

Editor: Mohsen Kalantari

Title: Future City Pilot 1: Using IFC/CityGML in Urban Planning Engineering Report

OGC Engineering Report

COPYRIGHT

Copyright © 2017 Open Geospatial Consortium. To obtain additional rights of use, visit http://www.opengeospatial.org/

Main conclusion:
“integration was not possible due to inconsistent coding of IFC elements that made transformation to CityGML complicated”

-> “a clear set of specification needs to be set for the preparation of IFC files”

We're making specific recommendations

To make IFC data Geo/CityGML ready

1. Use most specific Ifc class as possible (beam, chimney, door, roof etc) instead of the more generic IfcBuildingElementProxy

2. Guidelines when IfcSpaces should be used for enclosed spaces

3. How to construct valid volumetric objects

4. How to avoid self-intersections

5. How to correctly georeference

6. Follow:
   - existing IFC standard
   - implementation guidelines (define entities; allowed attribute values; schema's for consistent naming of objects etc)
   - external guidelines like BIM Basic IDM
     http://bimloket.nl/upload/documents/downloads/BIMbasisILS/BIM%20basic%20IDM%20(A4).pdf

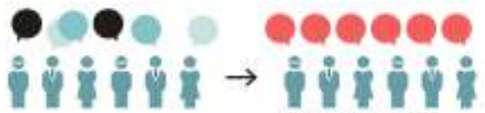
Validation is important!

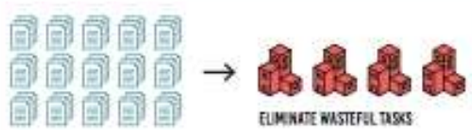
BIM Basic Information Delivery Model

BIM BASIC INFORMATION DELIVERY MANUAL (IDM)

1. WHY ARE WE SHARING THIS INFORMATION UNAMBIGUOUSLY?

in order to secure and reuse information more efficiently and effectively.





2. HOW ARE WE GOING TO SHARE THIS INFORMATION UNAMBIGUOUSLY?

Knowledge and practical experiences have shown that there is a significant common denominator.
We are not developing something new, but rather using existing structures, based on openBIM IFC.

3. WHICH STRUCTURE WILL WE USE?

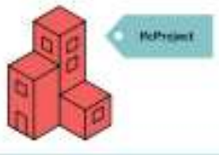
The agreements listed below help ensure that every involved party will always be able to find and supply the right information in the right place.

Checklist basic information delivery manual

3.1 FILE NAME

- ✔ Ensure that uniform and consistent naming is used for (discipline) models within the project.

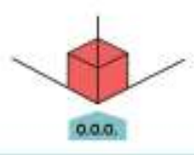
example: <Building>_<Discipline>_<Component>



3.2 LOCAL POSITION AND ORIENTATION

- ✔ The local position of the building is coordinated and close to the origin.

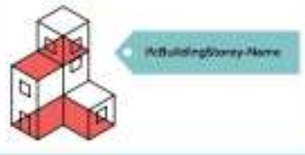
tip: use a physical object as point of origin, positioned at 0.0.0., and also export this to IFC.



3.3 BUILDING STOREYS AND NAMING

- ✔ Name Building Storeys only as ifcBuildingStorey-Name.
- ✔ Allocate all objects to the correct level.
- ✔ Within a project, ensure that all involved parties consistently use exactly the same naming, that can be numerically sorted with a textual description.

example 1: 00 ground floor
example 2: 01 first floor



3.4 CORRECT USE OF ENTITIES

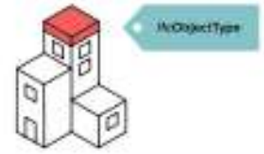
- ✔ Use the most appropriate type of BIM entity, both in the source application and the IFC entity.

example: slab = IfcSlab, wall = IfcWall, beam = IfcBeam, column = IfcColumn, stair = IfcStair, door = IfcDoor etc.

3.5 STRUCTURE AND NAMING

- ✔ Consistently structure and name objects.
- ✔ Correctly enter the object TYPE (IfcType, IfcObjectType or IfcObjectTypeOverride).
- ✔ Where applicable, also correctly enter the Name (IfcName or NameOverride).

example: roof insulation, type: glass fibre



3.6 CLASSIFICATION SYSTEM

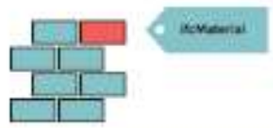
- ✔ Apply the existing classification system used in the relevant country. In the Netherlands this is the NL-SfB.
- ✔ Allocate to each object a four-digit NL-SfB variant element code.

example: 22.11

3.7 OBJECTS WITH CORRECT MATERIALIZATION

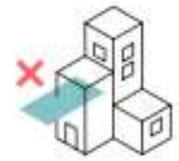
- ✔ Allocate objects with a material description (IfcMaterial).

example: limestone



3.8 DUPLICATES AND INTERSECTIONS

- ✔ There are no duplicates or intersections permitted. Make sure this is checked in IFC.



LEARNING TO SPEAK THE SAME LANGUAGE IS SOMETHING WE DO TOGETHER.

When naming objects, consider whether the name meets the following criteria. Double-check this, and know what information you are sharing.

- ✔ Significant
- ✔ Understandable
- ✔ Logical
- ✔ Insightful
- ✔ Consistent
- ✔ Recognizable

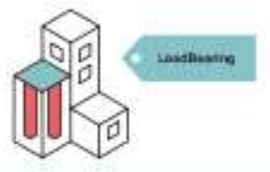
4. HOW CAN WE SECURE OTHER/FUTURE OBJECT INFORMATION?

Object information is secured in the correct properties and property sets as defined in IFC.

example: for beams, the properties FireRating, LoadBearing and IsExternal are part of the Pset_BeamCommon.

4.1 LOADBEARING

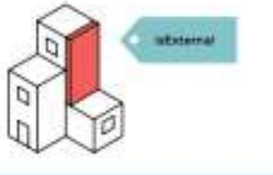
- ✔ Allocate objects, when applicable, with the property LoadBearing (True/False).



4.2 IS EXTERNAL

- ✔ Allocate objects, when applicable, with the property IsExternal (True/False).

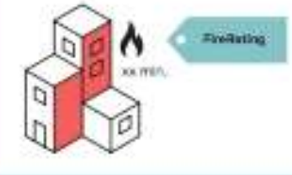
tip: both inner and outer faces of the façade have the property IsExternal True.



4.3 FIRERATING

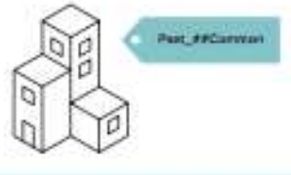
- ✔ Allocate objects, when applicable, with the property FireRating.

example: Apply the existing standard used in the relevant country.



4.4 PROJECT SPECIFIC

- ✔ Define which IFC properties you are using for each specific project.



They will be published soon

A proposed set of IFC guidelines for further processing in Geo software

August 31, 2017

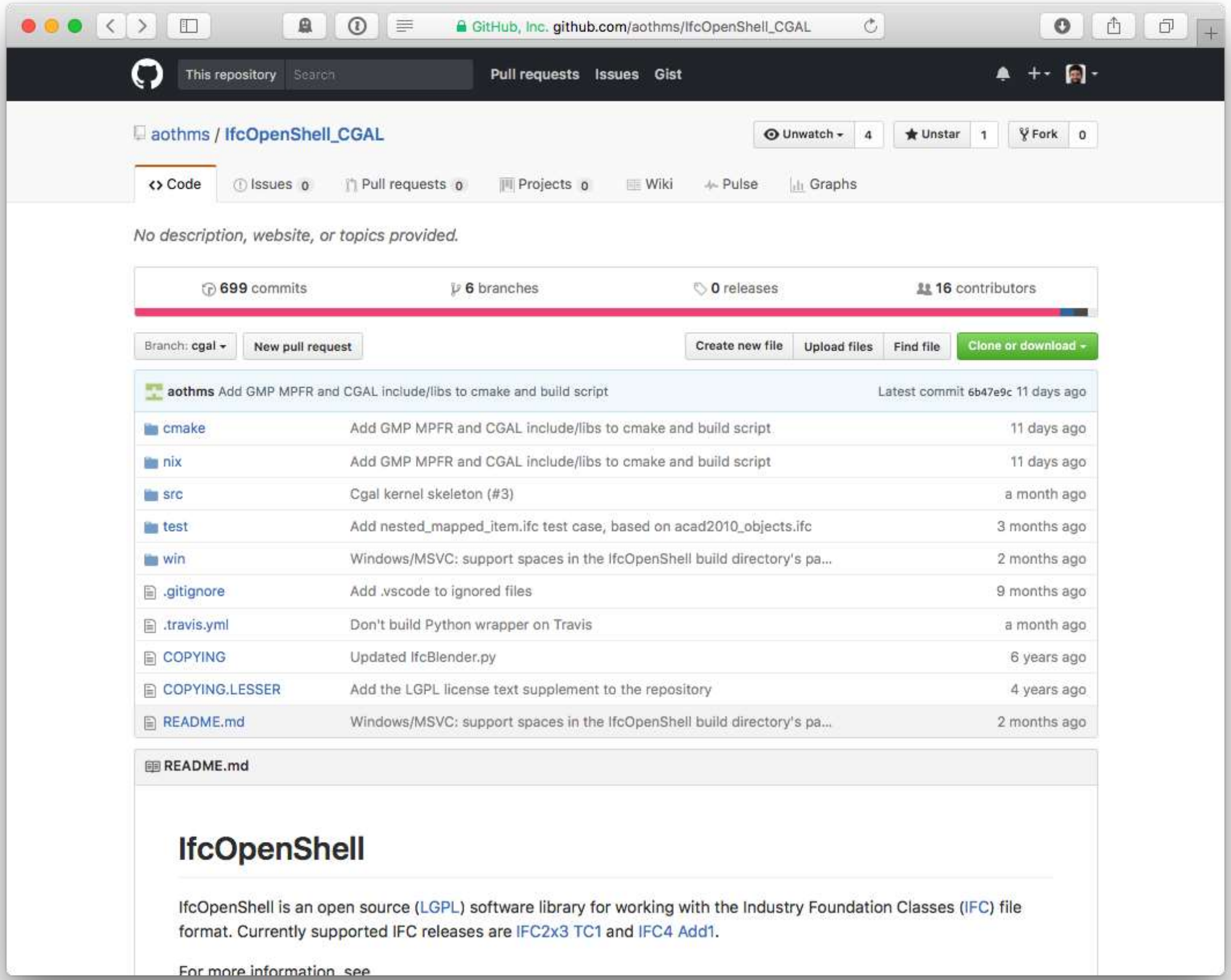
1 Introduction

In the GeoBIM project[1], we have been working on an interface between CityGML and IFC with the aim of bridging the Geo and BIM domains. This interface focuses on processing complex architectural IFC models in an automated fashion using IfcOpenShell[2] and CGAL[3], such as performing automated tests on them and converting them to CityGML.

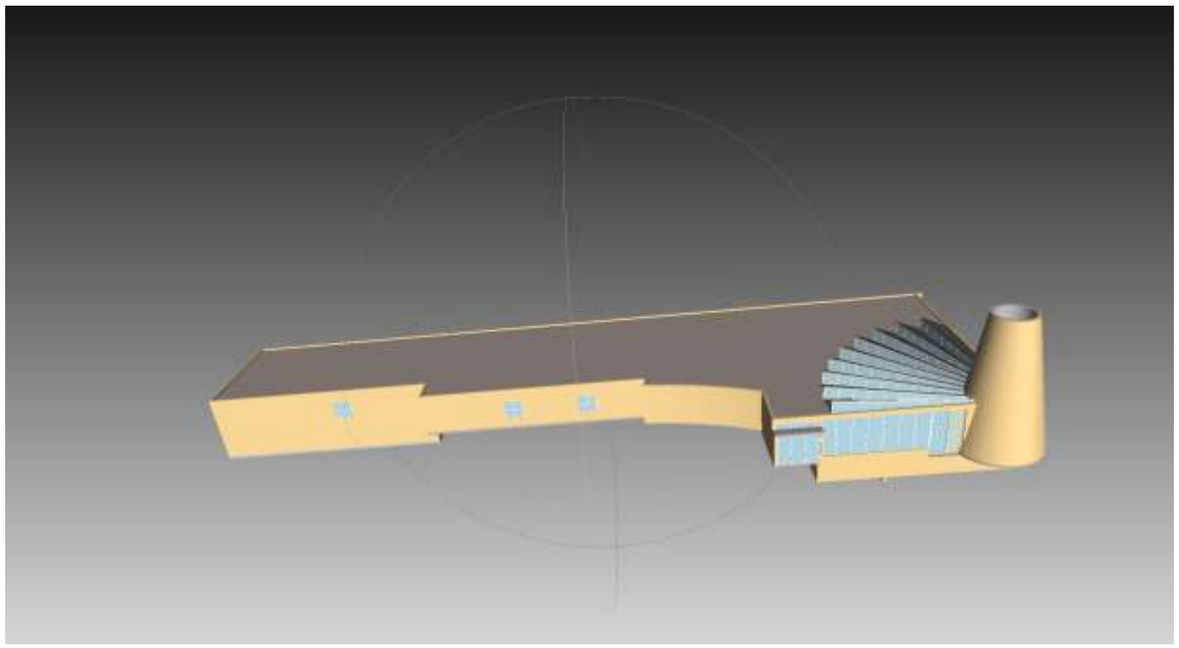
Many rules and recommendations regarding the proper use of IFC are already given in the IFC standard, implementation guidance, and external guidelines, among others. These range from fundamental aspects, such as how each IFC entity is defined and the possible values for each attribute, to common-sense practical rules, such as schemes for the consistent naming of objects. We make a brief summary of what is specified in each of these in Section 2.

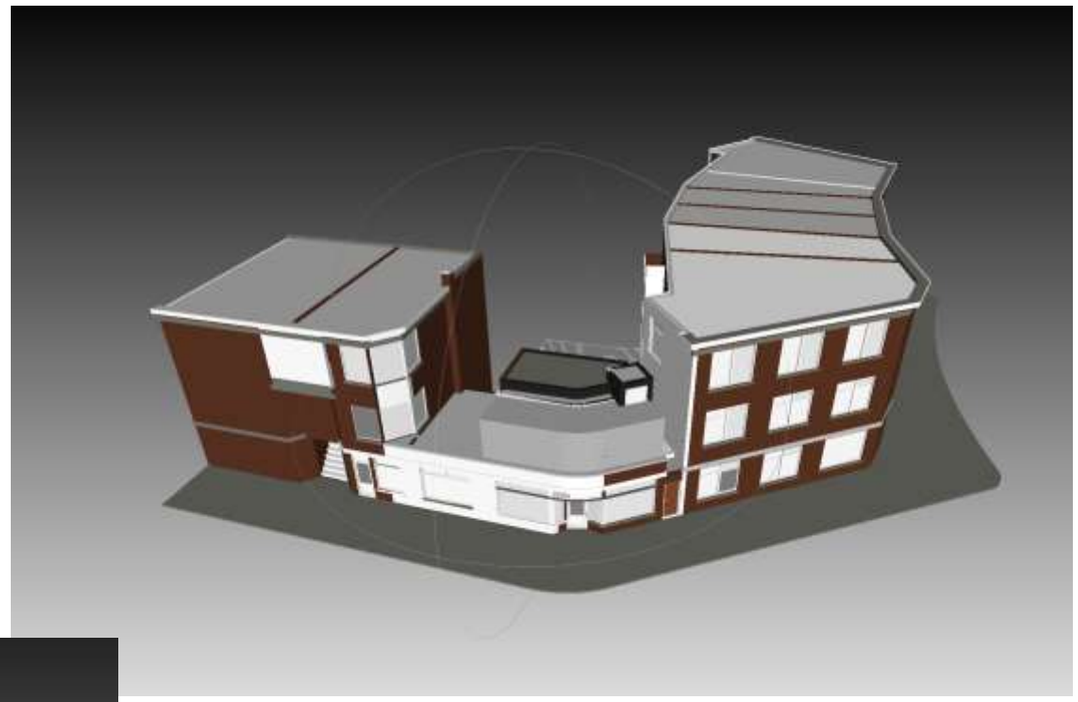
However, during the course of this project and based on previous experiences, we have

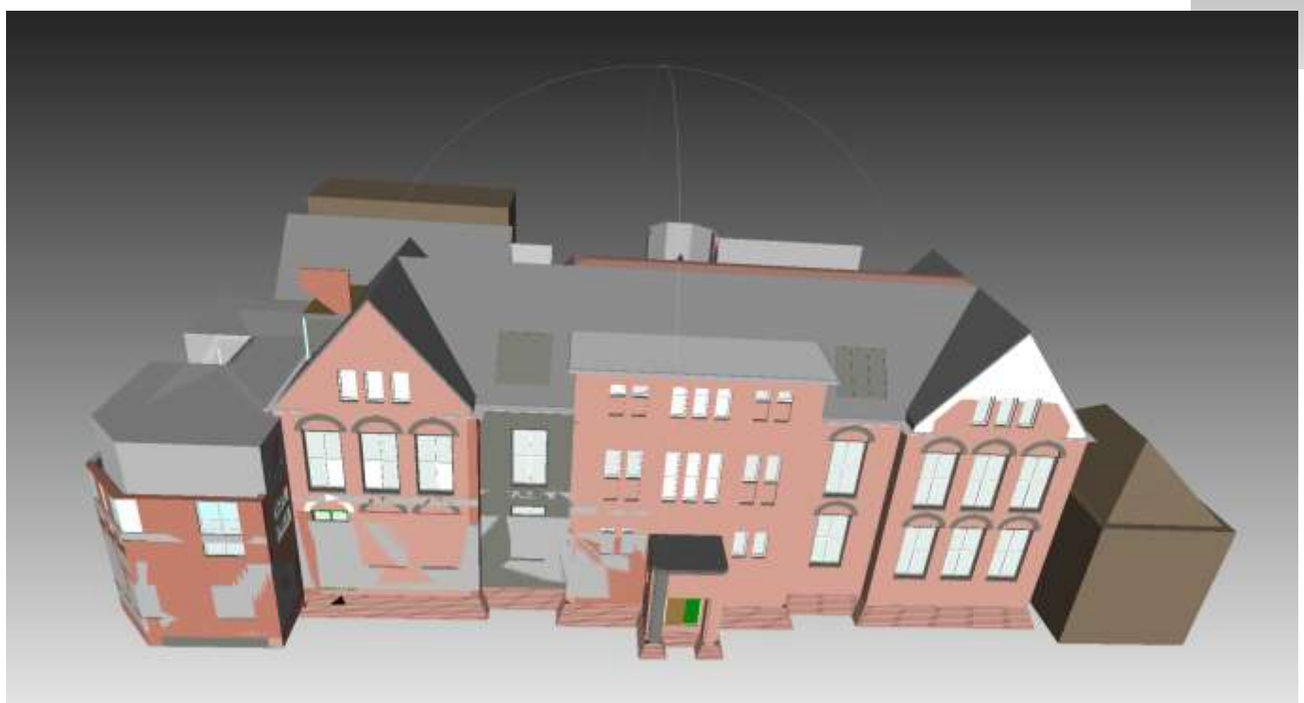
Code is open-source + results ain’t that bad



Our results

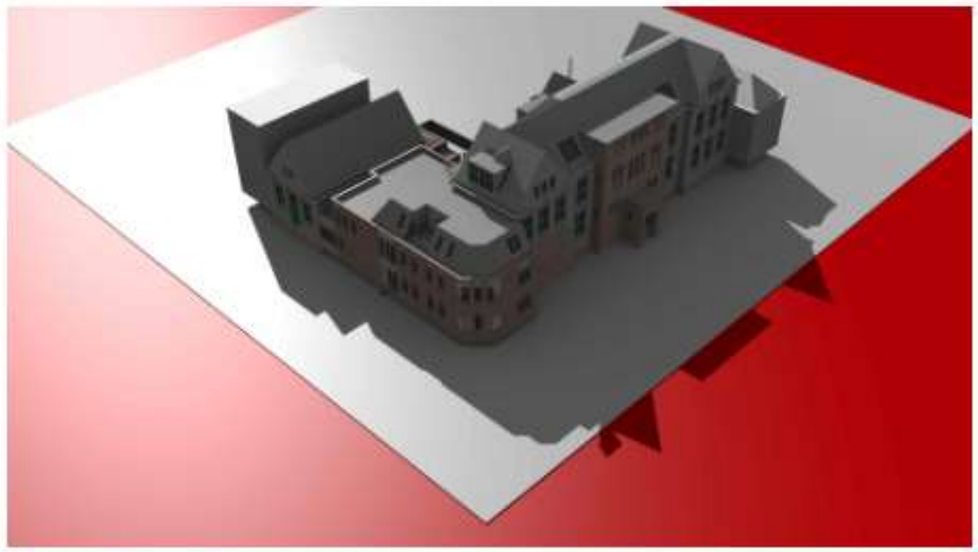


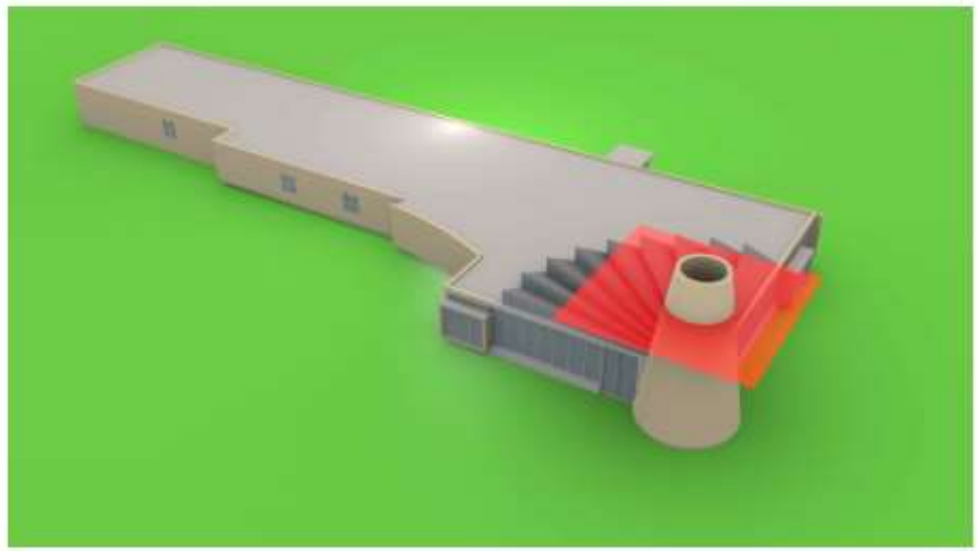




https://3d.bk.tudelft.nl/projects/geobim/

GeoBIM: Bridging the gap between Geo and BIM





Follow up

- 2 years Project of National Mapping Agencies from 10 countries:
  - Ireland
  - UK
  - Sweden
  - Denmark
  - Spain
  - Finland
  - Norway
  - Switzerland
  - France
  - Netherlands

- Kick off meeting on 29/30 nov 2017

- Aim:
  - Inventory of what GeoBIM is in the different countries:
    - Problems, issues, potential applications, initiatives
  - Identify 2-3 common issues to tackle, that can benefit from this international collaboration

Thank you!

With thanks to all partners of the project

j.e.stoter@tudelft.nl