Interface for CityGML/IFC integration
Geometric and Topological issues

Ken Arroyo Ohori, Filip Biljecki, Abdoulaye Diakité, Thomas Krijnen, Hugo Ledoux, Jantien Stoter
Develop an interface between CityGML and IFC to prepare for a fundamental solution to bridge the gap between Geo and BIM.

1. open-source API to represent IFC + CityGML with the same data structure
2. recommendations for future integration
Several parties involved with different perspectives.
1. gap in semantic
2. gap in geometry/topology
3. gap in coordinate reference systems

unlike most other similar initiatives, geometry is our main concern
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 (1)

explicit representation

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

3D GIS

IFC
IFC vs 3D GIS

Our conversion methodology.

Figure Nagel et. al (2009)
Our conversion methodology

1. IFC
2. Semantic Mapping
3. Geometric Transformation
4. Geometric & Semantic Refinement
5. CityGML

Note: A CityGML LoD3 shouldn’t have “thick” walls.
Our conversion methodology

Input IFC → union of all solids → removal interior

Use cases
Use case 1

Import an IFC file and get a (valid) CityGML out of it, Vice-versa
We've got few freshly designed real models for test.

Municipalities import IFC models in their GIS.

Use case 2
We’ve got few freshly designed real models for test
First feedbacks...

Several Geometric/Topological issues...
Several Geometric/Topological issues...
Found several issues and came up with some solutions:

- Non-planar faces —> triangulation
- Non-closed objects —> Nef polyhedra w/ Boolean set unions
- Non-manifolds —> Boolean set difference to remove parts
- CGAL crashes —> Catch error, slightly modify object and retry
- Small gaps —> Minkowski sum to close
- Overlapping objects —> Boolean set unions
But some issues are more difficult to solve…

- Unsupported and problematic geometries will create large gaps
- Many geometries are labelled with generic classes
- Self-intersections can’t be imported directly
- Very large files can’t be processed with current workflow

Also about georeferencing…
But, hey, we’ve learned a lot!

We’re making specific recommendations
1. How to construct valid volumetric objects
2. How to avoid self-intersections
3. Where IfcSpaces should be used
4. Which Ifc classes should be used
5. How to correctly georeference

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, 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 IfcOpen-Shell and CGAL, 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...
IfcOpenShell

IfcOpenShell is an open source (LGPL) software library for working with the Industry Foundation Classes (IFC) file format. Currently supported IFC releases are IFC2x3 TC1 and IFC4 Add1.

For more information, see...
Our results
GeoBIM: Bridging the gap between Geo and BIM

- Summary
- Introduction
- Previous work
- The proposed solution
- Deliverables
- CityGML/IFC interface
- Agreed and supported plan for follow up
- The collaboration
- Planning
- Meetings
- Sponsors
- Team
Thank you for listening

Abdoulaye A. Diakité
a.a.diakite@tudelft.nl
TU Delft