IFC-to-RDF: adaptation, aggregation and enrichment

Pieter Pauwels, Davy Van Deursen

Ghent

28 March 2012
Outline

1. Research context: SMML collaboration
2. Towards interoperability with SMML
3. The IFC-to-RDF web service
4. Smart Virtual Environments
5. Semantic Building Performance Checking
6. Interoperability of 3D information
RESEARCH CONTEXT: SMML COLLABORATION
Multimedia Lab - SmartLab

VUBrussel  KULeuven  Ghent University


Multimedia Lab  SmartLab

Erik  Davy  Pieter  Ruben  Jos
SmartLab – Research topics

• Building Information Modelling and Interoperability in AEC:
  – Industry Foundation Classes (IFC)
  – Semantic web and rules (N3Logic)

• Visualisation applications in AEC:
  – gaming technology
  – VR/AR
  – rendering applications

• Calculation and simulation applications in AEC:
  – building performance checking
  – building code checking
EPW Building Checker
Multimedia Lab – research topics

- Video coding and compression
- Image/video processing and analysis
- Multimedia content adaptation
- Metadata technology
- Gaming technology
- Standardization in the domain of multimedia applications and systems
  - W3C, VCEG/JVT, MPEG, VQEG
NinSuna: metadata-driven media adaptation & delivery
TOWARDS INTEROPERABILITY WITH SMML
Energy Performance Simulation based on BIM/IFC

Current situation in construction industry
THE IFC-TO-RDF WEB SERVICE
Faculty of Engineering and Architecture

**Convert**

- html
- csv
- Excel

**Convert**

- GbXML
- IFC
  - Revit
  - ArchiCAD

**RDF**

- Ifc.rdf
- DesignerX.rdf
- Unity.rdf
- ...

---

- RDF diagrams and network visualizations
- Conversion process flowchart
- Software and file formats: Ifc.rdf, DesignerX.rdf, Unity.rdf, GbXML, IFC, html, csv, Excel
IFC-to-RDF

Mapping schema’s:
1. ifcXML to RDF/XML via XSLT transformation
   1. Light-weight: conversion can occur completely on an instance level
   2. Time-consuming and prone to errors: it takes time and concentration to build the XSLT file
   3. Limited expressiveness of RDF/XML format (no rule functionality)
2. Translate EXPRESS schema to OWL ontology and use the ontology to build RDF instance graphs
   1. Less prone to errors: once the mapping between EXPRESS elements and OWL elements is made, everything goes smoothly and correctly
   2. High expressiveness because of N3 syntax

RDF view on IFC

• IFC specification is written in EXPRESS

```ex
ENTITY IfcDoor
  SUBTYPE OF ( IfcBuildingElement);
  OverallHeight : OPTIONAL IfcPositiveLengthMeasure;
  OverallWidth : OPTIONAL IfcPositiveLengthMeasure;
END_ENTITY;
```

• Two tasks
  – automatic transformation of EXPRESS to OWL
  – automatic conversion of IFC to RDF instances
EXPRESS => OWL

• Not really new
  – related work on EXPRESS-to-OWL conversion
  – related work on IFC-to-OWL conversion

• We did it to get started
EXPRESS => OWL

• Entity -> owl:Class

ENTITY IfcRelationship
ABSTRACT SUPERTYPE OF (ONEOF (IfcRelDefines, IfcRelAssociates))
SUBTYPE OF (IfcRoot);
END_ENTITY;

ENTITY IfcRelDefines
SUBTYPE OF (IfcRelationship);
END_ENTITY;

ifc:IfcRelationship
    rdfs:subClassOf ifc:IfcRoot;
a owl:Class.

ifc:IfcRelDefines
    rdfs:subClassOf ifc:IfcRelationship;
    owl:disjointWith ifc:IfcRelAssociates;
a owl:Class.
EXPRESS => OWL

- Attribute -> owl:DatatypeProperty

```plaintext
TYPE IfcPlaneAngleMeasure = REAL;
END_TYPE;

ENTITY IfcLightDistributionData;
    MainPlaneAngle : IfcPlaneAngleMeasure;
END_ENTITY;

ifc:mainPlaneAngle
    rdfs:domain ifc:IfcLightDistributionData;
    rdfs:range xsd:double;
    a owl:DatatypeProperty.
```

EXPRESS => OWL

TYPE IfcRatioMeasure = REAL;
END_TYPE;

TYPE IfcPositiveRatioMeasure = IfcRatioMeasure;
WHERE
    WR1 : SELF > 0.0;
END_TYPE;

ENTITY IfcProductsOfCombustionProperties
    CO2Content : IfcPositiveRatioMeasure;
END_ENTITY;

ifc:cO2Content
    rdfs:domain ifc:IfcProductsOfCombustionProperties;
    rdfs:range xsd:double;
    a owl:DatatypeProperty.

{?x a ifc:IfcProductsOfCombustionProperties. ?x ifc:cO2Content
?y. ?y math:notGreaterThan 0.0.} => false.
EXPRESS => OWL

- SELECT for entities/types => owl:unionOf-based class in rdfs:range
- ENUM types => rdfs:subClass with owl:one of
- List types => rdf:List
EXPRESS => OWL


• Issues
  – property name conflicts
    • RDF => classes and properties independent
    • EXPRESS => properties are declared with classes
  – automatic generation of N3 rules

• TODOs
  – OPTIONAL keyword should be mapped to OWL cardinality restrictions
  – UNIQUE and DERIVE keywords are not considered for the moment
ISO-10303-21;
HEADER;
FILE_DESCRIPTION ("'IFC Engine Kernel version 1.11 beta.'), '2;1');
FILE_SCHEMA ("'IFC2X3'");
ENDSEC;
DATA;
#1 = IFCORGANIZATION($, 'Revit Architecture 2009', $, $, $);
#2 = IFCAPPLICATION(#1, '2009', 'Revit Architecture 2009', 'Revit');
#3 = IFCCARTESIANPOINT((0.,0.,0.));
...
#4796 = IFCAXIS2PLACEMENT3D(#3,$,$);
#4797 = IFCLOCALPLACEMENT(#4714,#4796);
#4798 = IFCDOOR('Z921',#33,$,#4797,#4792,'110146',2134.,914.9);
...
IFC instances => RDF

• Pretty straightforward from a conceptual point of view
  – Instance naming
    • name of the type + line number
      – #4796 = IFCAXIS2PLACEMENT3D(#3,?,?,);
        => ifcAxis2Placement3D_4796

• Not so straightforward from a practical point of view
  – memory issues (triples are directly saved in RDF store)
  – slow conversion progress (no optimized implementation)
@prefix : <http://multimedialab.elis.ugent.be/ontologies/ifc/instances#>.
@prefix ifc: <http://multimedialab.elis.ugent.be/ontologies/ifc/ontology#>.
@prefix list: <http://www.co-ode.org/ontologies/lists/2008/09/11/list.owl#>.
@prefix xsd: <http://www.w3.org/2001/XMLSchema#>.
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>.

:ifcOrganization_1
    ifc:name "Autodesk Revit Architecture 2010"^^xsd:normalizedString;
    ifc:theIfcId "1"^^xsd:long;
    rdf:type ifc:IfcOrganization.

:ifcApplication_2
    ifc:applicationDeveloper :ifcOrganization_1;
    ifc:version "2010"^^xsd:normalizedString;
    ifc:applicationFullName "Autodesk Revit Architecture 2010"^^xsd:normalizedString;
    ifc:applicationIdentifier "Revit"^^xsd:normalizedString;
    ifc:theIfcId "2"^^xsd:long;
    rdf:type ifc:IfcApplication.

:ifcCartesianPoint_4
    ifc:coordinates ( "0.0"^^xsd:double "0.0"^^xsd:double );
    ifc:theIfcId "4"^^xsd:long;
    rdf:type ifc:IfcCartesianPoint.

:ifcDirection_5
    ifc:directionRatios ( "1.0"^^xsd:double "0.0"^^xsd:double "0.0"^^xsd:double );
    ifc:theIfcId "5"^^xsd:long;
    rdf:type ifc:IfcDirection.

:ifcDirection_10
    ifc:directionRatios ( "0.0"^^xsd:double "0.0"^^xsd:double "-1.0"^^xsd:double );
    ifc:theIfcId "10"^^xsd:long;
    rdf:type ifc:IfcDirection.

:ifcDirection_11
    ifc:directionRatios ( "1.0"^^xsd:double "0.0"^^xsd:double );
    ifc:theIfcId "11"^^xsd:long;
    rdf:type ifc:IfcDirection.
Upload IFC information into IFC/RDF graph

http://ninsuna.elis.ugent.be/IfcRDFService
Query IFC/RDF graph

http://ninsuna.elis.ugent.be/SPARQLEndpoint
### Browse IFC/RDF graph


<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>ifc:buildingAddress</code></td>
<td><code>ifc-inst:IFC2X3_TC1/ifcPostalAddress_35</code></td>
</tr>
<tr>
<td><code>ifc:compositionType</code></td>
<td><code>ifc:ifcElementCompositionEnum_element</code></td>
</tr>
<tr>
<td><code>is</code></td>
<td><code>[1_anonymous_resource]</code></td>
</tr>
<tr>
<td><code>is rdf:first of</code></td>
<td><code>2uXRFQpW95K AoBFXcolOBT (xsd:string)</code></td>
</tr>
<tr>
<td><code>ifc:globalid</code></td>
<td><code>ifc-inst:IFC2X3_TC1/ifcLocalPlacement_25</code></td>
</tr>
<tr>
<td><code>ifc:objectPlacement</code></td>
<td><code>ifc-inst:IFC2X3_TC1/ifcOwnerHistory_33</code></td>
</tr>
<tr>
<td><code>is ifc:relatingObject of</code></td>
<td><code>ifc-inst:IFC2X3_TC1/ifcRelAggregates_268</code></td>
</tr>
<tr>
<td><code>ifc:thefld</code></td>
<td><code>36 (xsd:long)</code></td>
</tr>
<tr>
<td><code>rdf:type</code></td>
<td><code>ifc:IfcBuilding</code></td>
</tr>
</tbody>
</table>

This page shows information obtained from the SPARQL endpoint at http://ninsuna.elis.ugent.be/SPARQLEndpoint.  
[As N3](#) | [As RDF/XML](#)
Let's export some RDF!

Ready to publish the current project and explore its information online? Your project will be published as part of the Linked Open Data cloud, thereby enabling you to connect with all kinds of information over the web.

Publish my project!

Read more about IFC\RDF and the Linked Open Data project.
SMART VIRTUAL ENVIRONMENTS
visualisation in Unity game engine

Faculty of Engineering and Architecture

BIM model

- Export to FBX
- Export to IFC

IFC representation

- Import into Virtual world in Unity3D

FBX representation

- Import into Virtual world in Unity3D

Virtual world in Unity3D

- Real-time communication

IFC-to-RDF Service

- BIM model in LOD cloud

BIM model in LOD cloud

- References to LOD cloud

Enriched BIM model
Creation of the virtual world

Connecting FBX representation to the IFC/RDF graph (1)

Connecting FBX representation to the IFC/RDF graph (2)

```
SELECT distinct ?s WHERE {
}
```
Basic user interface
SEMANTIC BUILDING PERFORMANCE CHECKING
LOD cloud

Material Manager

Layer Manager

Enrich

Validate

Process

1. Triangulate
2. Check space
3. Inflate

MODEL
NBN.xml

MODEL
Energy.xml

MODEL
results.rdf

MODEL
Ifc.rdf

MODEL
material.rdf

rules.n3

MODEL
NBN.xml

MODEL
Energy.xml

Vis. Scripts

Acoustics Viewer

Acoustics Viewer

Energy Viewer

Energy Viewer
Description of ‘facts’

Description of ‘rules’

{ ?x ont:hasMaterial ?y.
  ?y ont:hasR63Hz ?z
} log:implies
{ ?x ont:hasAcousticProperty ?z
}
# Calculate R' [dB] for each Space surface that has a calculated Te,i value
{
  # Find relevant Space surface and their Te,i values
  ?SS a ifc:SpaceSurface.
  ?SS ifc:spaceBoundary [ifc:relatedBuildingElement [:acousticTau ?tau1]].
  (?SCOPE 1) e:findall

  # Calculate R' [dB] for each Space surface
  (-10 ?x) math:product ?R
}
=>
{?SS :acousticR ?R}. 
EYE reasoning engine
EYE command

eye --nope --quick-possible --quick-false
facts.n3 rules.n3 --query query.n3 > result.n3

#Processed by $Id: euler.yap 3098 2009-10-24 20:31:17Z josd$

@prefix : [...]

inst:RoomBoundary_1 NBNS014001:ComfortLevel “normaal”^^xsd:string.
inst:RoomBoundary_2 NBNS014001:ComfortLevel “verhoogd”^^xsd:string.
inst:RoomBoundary_3 NBNS014001:ComfortLevel “normaal”^^xsd:string.
inst:RoomBoundary_4 NBNS014001:ComfortLevel “normaal”^^xsd:string.

#ENDS 16 msec
#Trunk : 94/326 = 28.8343558282209 %
#Branch: 1/93 = 1.0752688172043 %

INTEROPERABILITY OF 3D INFORMATION
<table>
<thead>
<tr>
<th></th>
<th>DXF</th>
<th>DWG</th>
<th>FBX</th>
<th>OBJ</th>
<th>STL</th>
<th>DAE</th>
<th>VRML</th>
<th>X3D</th>
<th>L3D</th>
<th>3DS</th>
<th>STEP</th>
<th>IFC</th>
<th>GBXML</th>
<th>ACIS</th>
<th>PARASOLID</th>
<th>OPEN CASCADE</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Mesh geometry</strong></td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td></td>
</tr>
<tr>
<td>Face normals</td>
<td></td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td></td>
<td>Y</td>
<td>Y</td>
<td></td>
</tr>
<tr>
<td>Texture mapping vertices</td>
<td></td>
<td>Y</td>
<td>Y</td>
<td>N</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
<td></td>
<td>Y</td>
<td>N</td>
<td>Y</td>
<td>Y</td>
<td></td>
</tr>
<tr>
<td><strong>Freeform 3D</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NURBS curve</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
<td>N</td>
<td>Y</td>
<td>N</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
<td>Y</td>
<td>N</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td></td>
<td></td>
</tr>
<tr>
<td>NURBS surface</td>
<td>*</td>
<td>*</td>
<td>Y</td>
<td>N</td>
<td>Y</td>
<td>N</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Parameter space vertices</td>
<td>*</td>
<td>*</td>
<td>Y</td>
<td>N</td>
<td>Y</td>
<td>N</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Trimming loops / holes</td>
<td>*</td>
<td>*</td>
<td>Y</td>
<td>N</td>
<td>N</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td></td>
<td>Y</td>
<td>Y</td>
<td></td>
</tr>
<tr>
<td><strong>2D Primitives</strong></td>
<td></td>
<td></td>
<td>Y</td>
<td>N</td>
<td>N</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Point</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Arc2D</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
<td>N</td>
<td>Y</td>
<td>N</td>
<td></td>
<td>Y</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ArcClose2D</td>
<td></td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
<td>N</td>
<td>Y</td>
<td>N</td>
<td></td>
<td>Y</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Circle2D</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
<td>N</td>
<td>Y</td>
<td>N</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Disk2D</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>Y</td>
<td>N</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ellipse2D</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
<td>N</td>
<td>Y</td>
<td>N</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td></td>
<td>Y</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Polyline2D</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
<td>N</td>
<td>Y</td>
<td>N</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td></td>
<td>Y</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Polypoint2D</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>Y</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>Y</td>
<td>N</td>
<td>N</td>
<td></td>
<td>Y</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Rectangle2D</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>Y</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>Y</td>
<td>N</td>
<td>N</td>
<td></td>
<td>Y</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Triangle2D</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>Y</td>
<td>N</td>
<td>N</td>
<td></td>
<td>Y</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Hyperbola</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>Y</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>Y</td>
<td>N</td>
<td>N</td>
<td></td>
<td>Y</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Parabola</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td></td>
<td>Y</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>3D Primitives</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Box</td>
<td>*</td>
<td>*</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>Y</td>
<td></td>
<td>Y</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Cone</td>
<td>*</td>
<td>*</td>
<td>N</td>
<td>N</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>Y</td>
<td></td>
<td>Y</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Cylinder</td>
<td>*</td>
<td>*</td>
<td>N</td>
<td>N</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>Y</td>
<td></td>
<td>Y</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sphere</td>
<td>*</td>
<td>*</td>
<td>N</td>
<td>N</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>Y</td>
<td></td>
<td>Y</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Torus</td>
<td>*</td>
<td>*</td>
<td>N</td>
<td>N</td>
<td>Y</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>Y</td>
<td></td>
<td>Y</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Polyline3D</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>Y</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>Y</td>
<td></td>
<td>Y</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Helix</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>Y</td>
<td></td>
<td>Y</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Geometric features</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Basic feature transformati</td>
<td>*</td>
<td>*</td>
<td>N</td>
<td>N</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td></td>
<td>Y</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ons (scale, rotate, etc.)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Surface modelling (sweep,</td>
<td>*</td>
<td>*</td>
<td>N</td>
<td>N</td>
<td>Y</td>
<td>N</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td></td>
<td>Y</td>
<td></td>
<td></td>
</tr>
<tr>
<td>revolve, etc.)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Boolean operations</td>
<td>*</td>
<td>*</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
<td>Y</td>
<td></td>
<td>Y</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Possible enhancements through a semantic web approach
Three-dimensional information exchange over the semantic web for the domain of architecture, engineering, and construction.


RDF

```
@prefix inst: <http://smartlab.elis.ugent.be/aimontologies/inst#>.
@prefix geom: <http://smartlab.elis.ugent.be/aimontologies/geom#>.

inst:CircleX geom:hasRadius "20"^^xsd.double.
inst:CircleX geom:hasCenter inst:PointX.
inst:PointX geom:hasXCoord "0"^^xsd.double.
```

OWL

```
geom:Circle rdf:type owl:Class.
geom:hasCenter rdfs:domain geom:Property.
geom:hasCenter rdfs:range geom:Point.
geom:Point rdf:type owl:Class.
geom:hasXCoord rdfs:domain geom:Point.
geom:hasXCoord rdf:type owl:DataTypeProperty.
geom:hasXCoord rdfs:range xsd:double.
```

N3Logic

```
{ ?x rdf:type geom:Circle .
  ?x geom:hasRadius ?y .
}
log:implies
{ ?x geom:hasArea ?z }
```
How to integrate diverse information models, in particular those that describe the same information differently???