LDAC 2015

3rd International Workshop on Linked Data in Architecture and Construction (LDAC)

Workshop Report

http://ldac-2015.bwk.tue.nl/
Technical University Eindhoven (TU/e)
Eindhoven, The Netherlands
July 15-17, 2015

TU/e

Abstract

The topic of the 2015 LDAC workshop is *"The use the Web of Data technologies for building information management in diverse practical use cases"*. The interest to this question arises from the expected benefits such as the ability to easily link building information models to and from external data sources, to open the models to new use cases and applications, to enable truly decentralised publication of models, to support loosely coupled interoperation through cross-model linking, and to utilise reasoning and other services developed in Semantic Web research. New use cases arise in various contexts. The 2015 LDAC event provides an opportunity to present and openly discuss these use cases.

Workshop Chair

Jakob Beetz
Department of the Built Environment, Eindhoven University of Technology, Eindhoven, The Netherlands

Organising Committee

Jakob Beetz
Department of the Built Environment, Eindhoven University of Technology, Eindhoven, The Netherlands
Pieter Pauwels
Department of Architecture and Urban Planning, Ghent University, Ghent, Belgium
Kris McGlinn
Department of Computer Science, Trinity College Dublin, Dublin, Ireland
María Poveda-Villalón
Department of Artificial Intelligence, Universidad Politécnica de Madrid, Madrid, Spain

Introduction to the LDAC2015 Workshop

This report summarizes the presentations and discussions held during the 3rd International Workshop on Linked Data in Architecture and Construction (LDAC), which took place at the Eindhoven University of Technology (TU/e) in Eindhoven, the Netherlands, from July 15 to July 17, 2015. The event was organised by Jakob Beetz the TU/e. The LDAC workshop typically gathers researchers with diverse ideas about ways in which linked data and semantic web technologies can enhance information exchange in the AEC domain.

Using Web of Data technologies for building information management has been in the focus of a number of research and development activities in recent years. Such efforts have resulted in the formation of a community manifest e.g. in the W3C Linked Building Data Community Group and a working group forming within the buildingSMART organization. The LDAC workshop series aims at bringing researchers, industry stakeholders, standardization bodies and other interested members of this growing community together. The aim of the LDAC workshop is to present current developments, coordinate efforts, gather stakeholders use cases and plan future activities.

The topic of the 2015 event was **"The use the Web of Data technologies for building information management in diverse practical use cases"**. The interest to this question arises from the expected benefits such as the ability to easily link building information models to and from external data sources, to open the models to new use cases and applications, to enable truly decentralised publication of models, to support loosely coupled interoperation through cross-model linking, and to utilise reasoning and other services developed in Semantic Web research. New use cases arise in various contexts. The 2015 LDAC event provided an opportunity to present and openly discuss these use cases.

The event followed to a large extent the program outline and set up that was used in the 2014 LDAC workshop in Espoo. Namely, in addition to a high number of plenary sessions, two parallel tracks took place: one track focused on industrial use cases and vocabularies for the Web of Building Data, and the other focused on technical advances and prospects. In addition, a number of hands-on sessions was hosted as well, on diverse topics.

The third edition of the LDAC workshop was cohosted with the 22nd International Workshop of the European Group for Intelligent Computing in Engineering (EG-ICE), which took place from 12 to 15 July, 2015. As the focus of EG-ICE is to promote research and applications of advanced informatics to all aspects of engineering, the purposes of the LDAC Workshop align well with those of the EG-ICE community and workshop. The primary goals of the EG-ICE group are to promote advanced informatics research across Europe by improving contact between researchers, fostering research collaboration and enhancing awareness of the latest developments. With this latter aim in mind, the group maintains active contact with similar groups in countries outside Europe and encourages membership by non-Europeans. The EG-ICE group has a further aim to increase the awareness of industry of advanced informatics as well as the economic benefits that can be gained by implementation.

Topics

- Use case presentations (sensor data, building performance checking, building regulation integration, etc.)
- ifcOWL, ifcRDF, bsDD
- Data Management for Energy Efficient Building Life Cycle Processes
- Linking BIM models to external data sources
- Cross-model linking (HVAC – MEP – ARCH – …)
- Distributed / decentralized BIM
- Loose linking vs. tight integration
- Querying and reasoning with building information
- Open world assumption vs. closed world assumption
- Preservation of Linked Data

Program

Wednesday, July 15th

09:30	EG-ICE Keynote Topic: Intelligence: SPACE | DESIGN. SYNTAX. SEMANTICS. COGNITION.	
10:30	Break	
11:00	EG-ICE session 7: Logic, graphs and data structures	
12:30	Closing EG-ICE	
13:00	Lunch	
14:30	**Jakob Beetz:** Opening LDAC	
15:00	Plenary Presentations	
		Nam Vu Huang, Seppo Törmä
		IFC-to-Linked Data Conversion: MultiLayering Approach
		Pieter Pauwels, María Poveda-Villalón
		Towards an industry-wide ifcOWL: choices and issues
16:00	Break	
16:30	Plenary Presentations	
		Peter Bonsma
		Geometry within OWL, real use of ifcOWL and CMO with Extensions for Proficient
		Hendro Wicaksono
		Rules integration in OWL BIM
		Walter Terkaj
		Ontology-based Factory Design & Management
18:30	Wrap-up	
20:00	LDAC Dinner.	

Thursday, July 16th

9:30	Plenary Presentations	
		Leif Granholm
		OGC activities regarding semantic web
		Daan Oostinga
		Use cases for linked data
		Michel Böhms
		Linked Data in VCON

Tarcisio Mendes de Farias, Ana Roxin, Christophe Nicolle
Semantically Adapting IFC Relations into OWL
Léon van Berlo, Pim van den Helm
IFC Model checking with N3 and reasoning Linked building Data

12:00	Lunch
13:15	Parallel discussion and work tracks **Use cases track** **Technical track**
18:30	Plenary Concluding Discussion
20:00	LDAC Dinner

Friday, July 17th

09:00	Get Together
09:30	LDAC Hands-on Sessions **Kris McGlinn** Converting Tabular Data to RDF using CSVW **Nick Kaklanis** Converting OWL to JSON-LD **Matthias Weise, Pieter Pauwels** Best practices for publishing and linking BIM data: scoping of IFC models (MVD and IfcDoc introduction)
12:00	Lunch
13:30	LDAC Hands-on Sessions **María Poveda-Villalón, Pieter Pauwels** Introduction to Ontology Engineering: Tutorial for ifcOWL
14:30	Wrap-up **Jakob Beetz** Closure and List of actions **Nam Vu Hoang** LDAC memories
16:00	Workshop end

Individual presentations

> Nam Vu Huang, Seppo Törmä – IFC-to-Linked Data Conversion: MultiLayering Approach

In this presentation, Nam Vu Hoang outlines that there is a need to convert IFC data into Linked Data. This would be useful to allow sharing distributed data, providing model-level and object-level access and linking ifcRDF, other RDF and non-RDF data. By doing so, an appropriate combination can be made of BIM technologies and linked data technologies. Linked data technologies are not the same as semantic web technologies, considering the differences outlined in Figure 1.

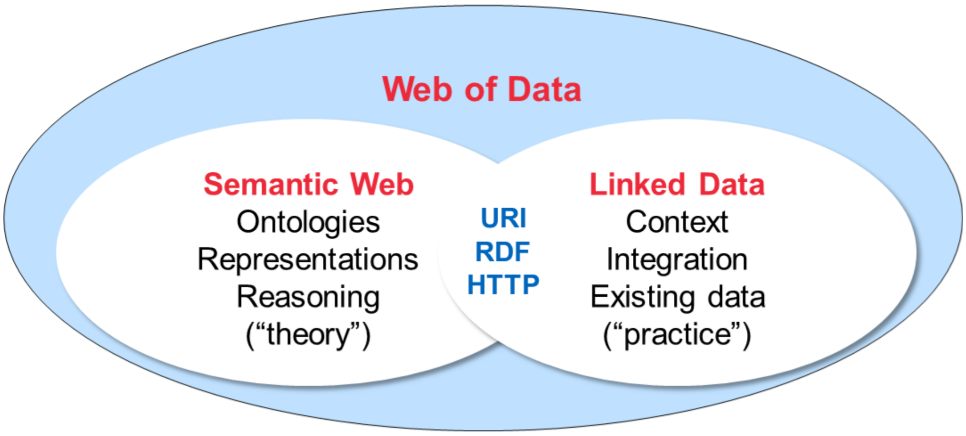

Figure 1: Semantic Web technologies versus Linked Data technologies.

In order to obtain a conversion from IFC to Linked Data, a double conversion procedure is needed, on the one hand from the IFC schema to an ifcOWL ontology, and on the other hand from an IFC-SPF file to an RDF graph. An ifcOWL ontology can be in one of the available OWL2 profiles: OWL2 DL, EL, QL, RL (see Figure 2). Each of these profiles has a different expressiveness and thus each take a different amount of time during inference.

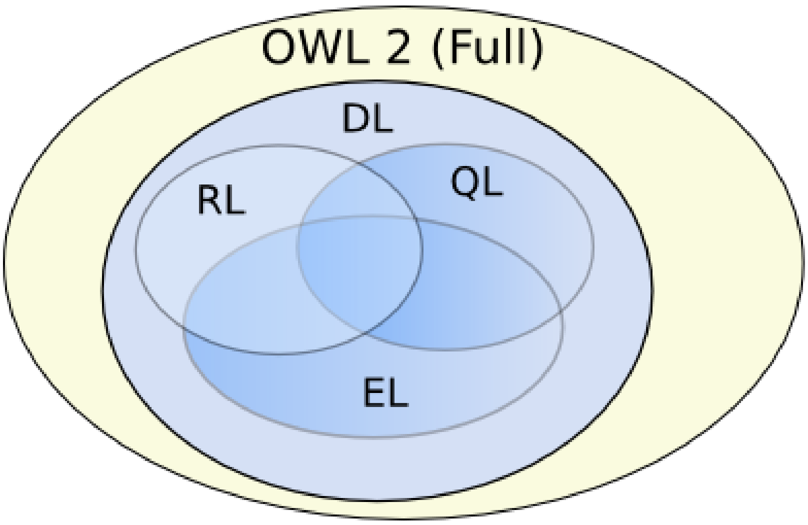

Figure 2: The different OWL2 profiles (see http://www.w3.org/TR/owl2-overview).

Considering the features of the different OWL2 profiles, there are contrasting opportunities available. On the one hand, one could aim at an ifcOWL ontology with as much type information as possible (high expressiveness). On the other hand, one could aim for an ifcOWL ontology that is in one of the less expressive profiles, but is more efficient at execution time. The ifcRDF graphs should follow the ifcOWL ontology. It is therefore proposed to build three ontologies with different levels of expressiveness that are connected (layered) with each other, as displayed in Figure 3. Figure 4 shows what is available in each of the three proposed ontologies.

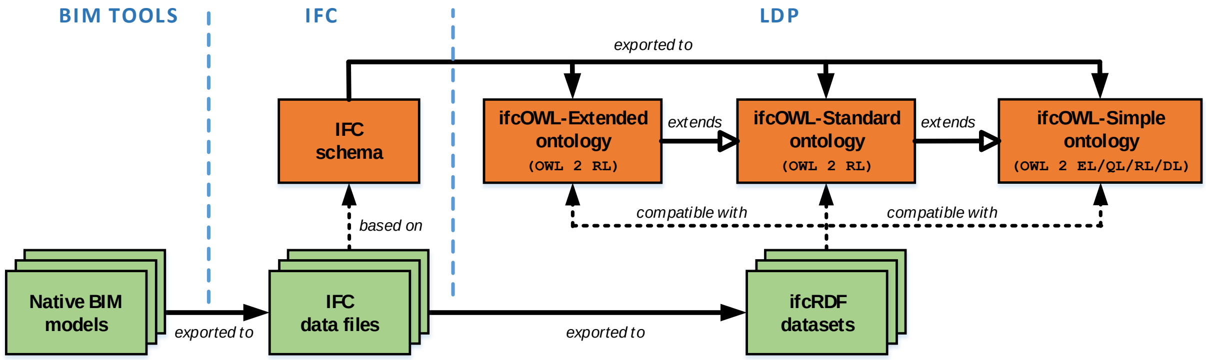

Figure 3: The proposed conversion procedure from IFC to Linked Data.

Layer	Compatible OWL profiles	Contents
ifcOWL-Simple	OWL 2 EL OWL 2 QL OWL 2 RL OWL 2 DL	Ontology metadata plus: - All basic constructs - All names of types - All inheritance relations of types - All names of properties
ifcOWL-Standard	OWL 2 RL OWL 2 DL	All above plus: - All value range constraints of properties - All keys of classes (entity types)
ifcOWL-Extended	OWL 2 DL	All above plus: - All cardinality constraints of properties

Figure 4: The content that is available in each of the three proposed ifcOWL ontologies (see http://drumbeat.cs.hut.fi/owl/).

The conversion procedure as displayed in Figure 3 is implemented in IFC2LD conversion software. This software is a free open-source JAVA program, which provides an API and a command line interface (CLI). The source code is available at https://github.com/Web-of-Building-Data/ifc2ld and the distributives are available at http://drumbeat.cs.hut.fi/ifc2ld. The three ifcOWL ontologies that are produced with the IFC2LD software are compared in Figure 5.

	ifcOWL-Simple (lightweight)	ifcOWL-Standard (normal)	ifcOWL-Extended (heavyweight)
OWL Profiles	**OWL 2 EL/QL/RL/DL**	**OWL 2 RL/DL**	**OWL 2 DL**
- **Ontology metadata**	+	+	+
- **Basic constructs**	+	+	+
- **Type definitions**	+	+	+
- **Type inheritance relations**	+	+	+
- **Property definitions**	+	+	+
- **Property value ranges**		+	+
- **Keys of entity classes**		+	+
- **Property cardinalities**			+
Time complexity	PTime	PTime (?)	N2ExpTime
File size	205 Kb	610 Kb	930 Kb
Triples / Classes / Properties	5200 / 1400 / 105	13200 / 2900 / 2000	22300 / 4200 / 2000
Global individuals	1315	1315	3

Figure 5: Comparison between the three ifcOWL ontologies.

There are a number of differences between this approach and the approach presented by Pieter Pauwels and Walter Terkaj. These differences are summarized in Figure 6 and 7.

		Pieter's & Walter's approach	Nam's & Seppo's approach
1	Number of ontologies for 1 IFC schema	1	3
2	Compatibility with OWL 2 EL/QL/RL	No	Yes
3	Domains and ranges are used	Yes	No (except for properties of basic types)
4	owl:allValuesFrom are used for properties	Yes	Yes (only Standard/Extended)
5	Simple datatypes	Replaced with XSD types	Wrap XSD types
6	Defined datatypes	Wrap XSD types	Subclasses of simple datatypes or other defined datatypes
7	Simple datatypes BOOLEAN, LOGICAL	Replaced with xsd:boolean	New enum types with values: TRUE, FALSE, and UNKNOWN
8	Simple datatype REAL	Replaced with xsd:double	User choice: xsd:decimal (default), xsd:double (only OWL 2 RL/DL), or owl:real (only OWL 2 EL/QL/DL)
9	Simple datatype NUMBER	= INTEGER	= REAL

Figure 6: First 9 main differences between the proposed conversion procedure and the conversion procedure as proposed by Pieter Pauwels and Walter Terkaj.

		Pieter's & Walter's approach	Nam's & Seppo's approach
10	Declarations of enum types	Using owl:oneOf	- Using rdf:type (Simple, Standard) - Using owl:oneOf (Extended)
11	Declarations of select types	Using owl:unionOf	- Using rdfs:subClassOf (Simple, Standard) - Using owl:unionOf (Extended)
12	Declarations of collection types	New pattern for LIST types	Common pattern for LIST, SET, ARRAY, BAG types
13	Naming individuals	All individuals are named using line numbers	a) IfcRoot derived objects are named with GUIDs b) Other objects are depended on user choice: - leave as blank nodes (only supported by OWL 2 RL/DL); - naming with special algorithm

Figure 7: Remaining 4 main differences between the proposed conversion procedure and the conversion procedure as proposed by Pieter Pauwels and Walter Terkaj.

Key aspects in the conversion procedure include:

- simple data types are "wrapped" behind classes (`expr:REAL a owl:Class ; rdfs:subClassOf expr:Defined .`)
- defined data types are "subclassed" (`ifc:IfcLengthMeasure a owl:Class ; rdfs:subClassOf expr:REAL .`)
- aggregation data types (LIST, SET, ARRAY, BAG) are all converted to custom constructions, inspired by the Ordered List Ontology (OLO)
- inverse properties are not defined in the ifcOWL ontology.
- domain and range restrictions are not defined for the properties, but appropriate domain and range restrictions are added as OWL class restrictions.
- object properties are not renamed, and they can have more than one domain and one range.

> Pieter Pauwels, María Poveda-Villalón - Towards an industry-wide ifcOWL: choices and issues

There have been many proposals in our past regarding the conversion of the IFC EXPRESS schema into an OWL ontology. By now, we start to feel the need for formalisation and standardisation among these diverse proposals, so that there is at least a version that can be adopted industry-wide, across the diverse use case scenarios. This is also at stake in the Linked Data Working Group at BuildingSMART, which was proposed and accepted in the BuildingSMART Technical Summit in London 2015. This presentation therefore focuses on the open issues that need to be resolved in order to obtain such an industry-wide ifcOWL ontology. We hereby limit to the ifcOWL ontology that was produced by Pieter Pauwels and Walter Terkaj (http://linkedbuildingdata.net/resources/20150219_IFC4_ADD1.owl) and the ifcOWL Extended ontology that was produced by Nam Vu Hoang and Seppo Törmä (http://drumbeat.cs.hut.fi/owl/).

The main features in the conversion procedure by Nam Vu Hoang and Seppo Törmä were outlined in the previous presentation. The main features in the conversion procedure by Pieter Pauwels and Walter Terkaj are outlined below:

- simple data types are “wrapped” behind classes (`expr:REAL a owl:Class .`)
- defined data types are “subclassed” (`ifc:IfcLengthMeasure a owl:Class ; rdfs:subClassOf expr:REAL .`)
- aggregation data types (LIST, SET, ARRAY, BAG) are all converted to custom constructions, inspired by the List ontology proposed by Drummond et al., but many alternative options are available as well
- inverse properties are defined in the ifcOWL ontology, except for a limited number of exceptions
- domain and range restrictions are defined for the properties and for the classes (OWL class restrictions)
- object properties are renamed into [PropertyName]_OF_[ClassName], in order to ensure that each object property has always one domain and one range.
- SELECT types are converted to unionOf declarations
- ENUMERATION types are converted to oneOf declarations

When considering only the ifcOWL extended ontology proposed by Nam Vu Hoang, there are many similarities between the two ontologies. In this case, the differences as displayed in Figure 8 and 9 apply. In most cases, there is agreement, except for the representation of domains and ranges (line 3 in Figure 8), the representation of the simple data types in line 7, 8 and 9 of Figure 8, and the representation of aggregation data types in line 12 of Figure 9.

		Pieter’s & Walter’s approach	Nam’s & Seppo’s approach
1	Number of ontologies for 1 IFC schema	1	1
2	Compatibility with OWL 2 EL/QL/RL	OWL DL	OWL DL
3	Domains and ranges are used	Can we follow the subproperty proposal?	
4	owl:allValuesFrom are used for properties	Yes	Yes (only Standard/Extended)
5	Simple datatypes	Wrap XSD types	Wrap XSD types
6	Defined datatypes	Subclasses of simple datatypes or other defined datatypes	Subclasses of simple datatypes or other defined datatypes
7	Simple datatypes BOOLEAN, LOGICAL	Replaced with xsd:boolean	New enum types with values: TRUE, FALSE, and UNKNOWN
8	Simple datatype REAL	Replaced with xsd:double	User choice: xsd:decimal (default), xsd:double (only OWL 2 RL/DL), or owl:real (only OWL 2 EL/QL/DL)
9	Simple datatype NUMBER	= INTEGER	= REAL

Figure 8: Attempted alignment (part 1) between our ifcOWL ontology and the extended ifcOWL ontology by Nam Vu Hoang.

		Pieter's & Walter's approach	Nam's & Seppo's approach
10	Declarations of enum types	Using owl:oneOf	- Using owl:oneOf (Extended)
11	Declarations of select types	Using owl:unionOf	- Using owl:unionOf (Extended)
12	Declarations of collection types	MANY OPTIONS HERE – LARGE IMPACT, as all geometric data (90% of IFC) is in lists ==> huge impact on triple count	

Beyond development of an extended ifcOWL

Figure 9: Attempted alignment (part 2) between our ifcOWL ontology and the extended ifcOWL ontology by Nam Vu Hoang.

If an alignment can be achieved between our ifcOWL ontology and the extended ifcOWL ontology by by Nam Vu Hoang and Seppo Törmä (http://drumbeat.cs.hut.fi/owl/), this would be a great achievement already for the LDAC workshop. In addition to the differences outlined in Figure 8 and 9, the following open issues are also undecided and need to be decided:

- Ontology name and provenance:
    - <http://www.buildingsmart-tech.org/ifcOWL>
- URI naming convention for object properties
    - [Attribute]_of_[Entity]
    - [Attribute]_of_[SUPERTYPE]
- URI naming convention for individual Enumerations
    - [VALUE]_of_[EnumType]

If these issues are resolved, we reach a situation as displayed in Figure 10, with one formalised conversion procedure and three identical ifcOWL ontologies. This conversion procedure and agreed ifcOWL ontology can then be proposed to the BuildingSMART alliance in October 2015.

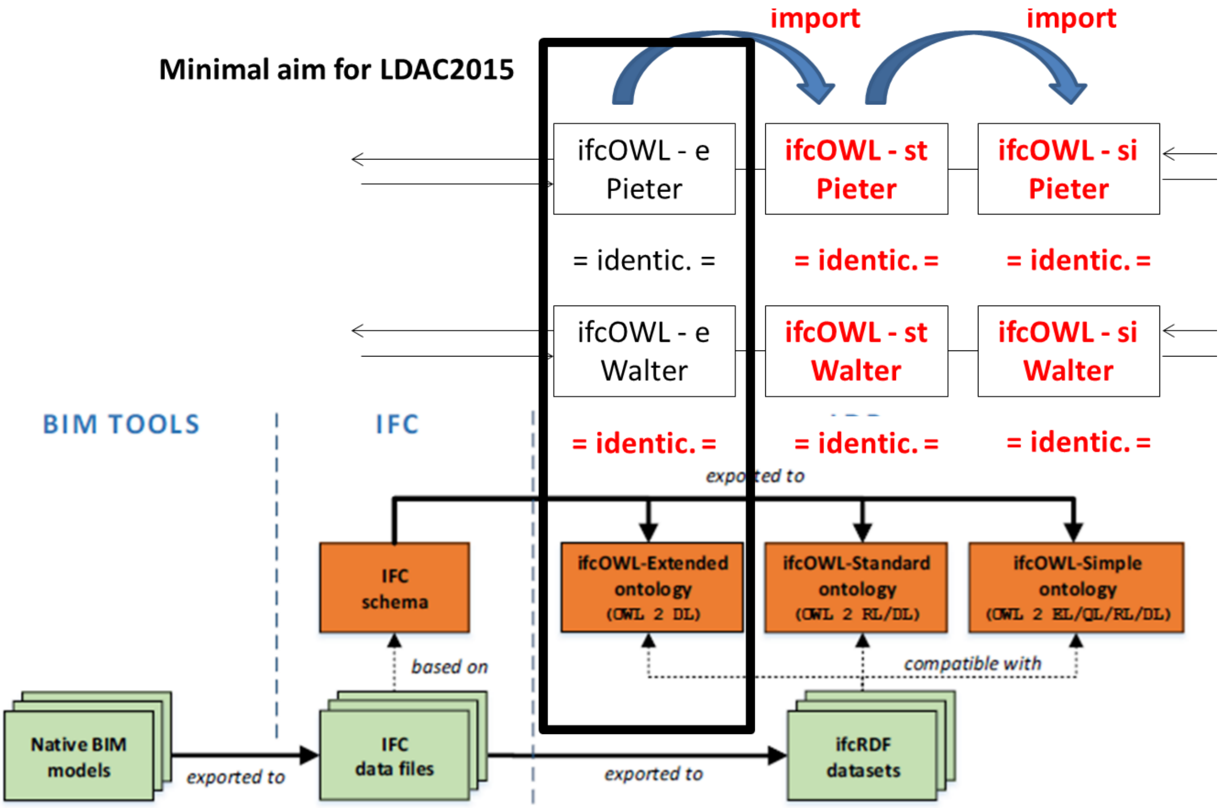

Figure 10: The minimal aim for LDAC2015 includes an alignment between the three ontologies that are currently available in the room.

> Peter Bonsma - Geometry within OWL, real use of ifcOWL and CMO with Extensions for Proficient

As part of the Proficient project (http://proficient-project.eu/Main.aspx?uri=43,14,15), an investigation was done of the semantic representation of geometry. It is hereby investigated to what extent geometry can be described not in a static 'flat' representation of points, solids and surfaces, but rather as a combination of the diverse parametric or procedural descriptions that allow to build the actual geometry. By encoding this information semantically, it becomes available in the data, rather than in the code. Hence, it can be adapted in a more flexible manner.

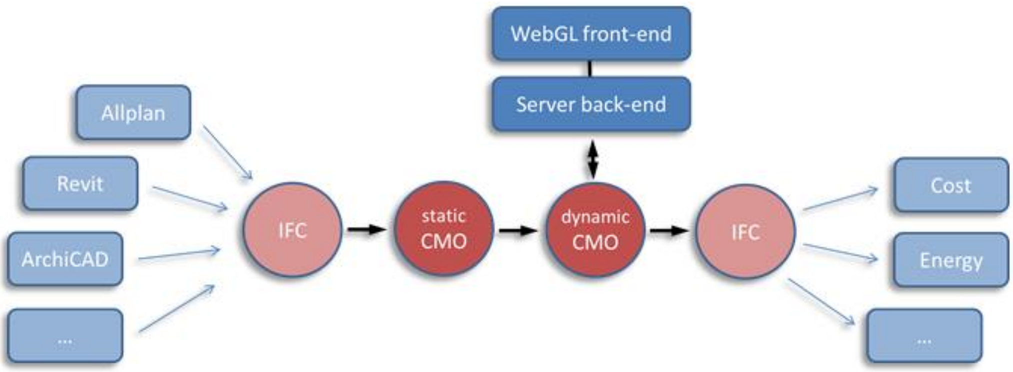

Figure 11: As a static IFC file is enriched with dynamic CMO instances, the constraints and the parameters of the geometry can be represented in a semantic fashion.

The results of this project are presented using an e-Marketplace use case. In this use case, IFC is used as a fixed model. On top of this model, additional CMO instances (ABox) are created. These CMO instances add ‘geometric constraints’ to the IFC file. This procedure is schematically presented in Figure 11. Adding the additional CMO instances (geometric constraints and parameters) requires interpretation by the end user. For this task, the end user is supplied with a kind of constraint-generation tool.

> Hendro Wicaksono - Rules integration in OWL BIM for holistic building energy management

As part of the FP7 project KnoholEM, a framework is developed for holistic building energy management in the operational phase of a building, hereby relying on semantic web technologies. The main purpose of the presented work is to extend BIM standards to allow reasoning capabilities, but also remain aligned with existing standards (IFC). For allowing the reasoning capabilities, SWRL rules and axioms are incorporated in the framework. An overview of the KnoholEM solution is provided in Figure 12.

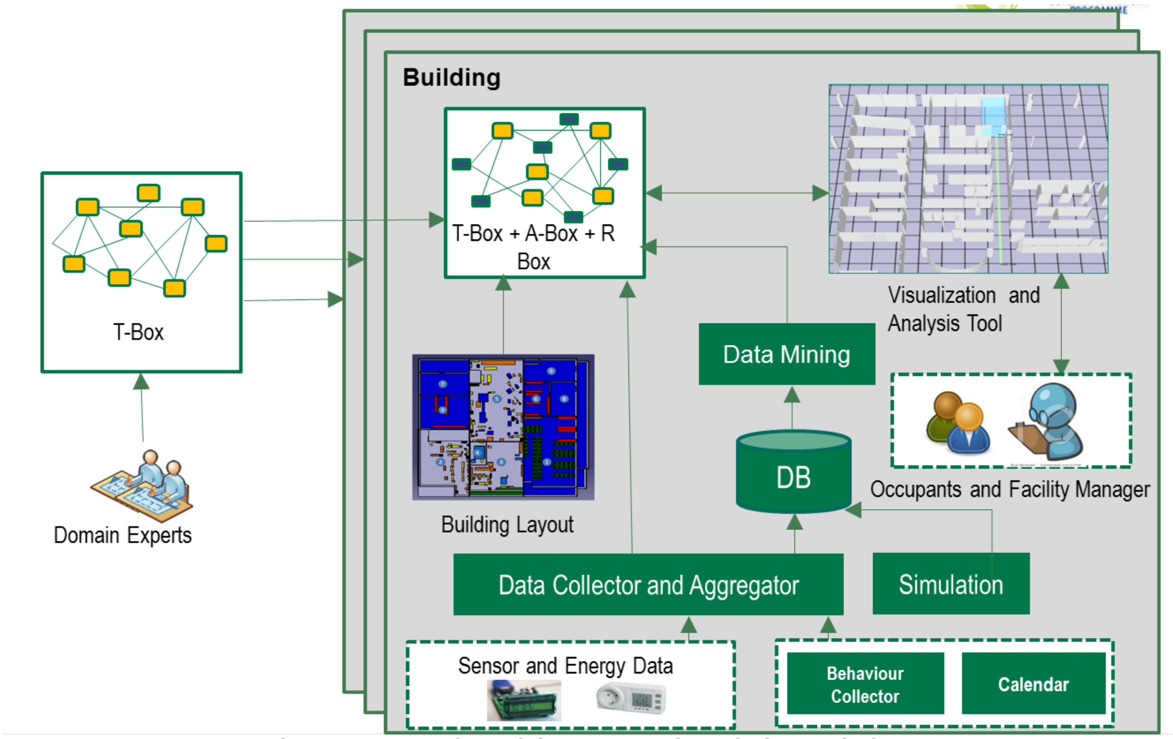

Figure 12: Overview of the proposed KnoholEM solution.

Central in the KnoholEM solution is an aligned TBox (ontologies), ABox (instances) and RBox (rules). This core assembles a representation of the building, relying on a building layout tool for the geometry (OntoCAD), which is enriched with sensor and energy data using data mining techniques. Based on this core, a visualization and analysis tool is developed that allows to better manage the operations in a building.

An overview of the ontology development process that was followed to build the core TBox + ABox + RBox is displayed in Figure 13. The six following steps are followed in this process:

1. main taxonomy by domain experts,

2. align to IFC, simple mapping through class annotation
3. ontology population with OntoCAD,
4. population by behaviour modeller,
5. enrichment with SWRL, data mining.
6. population by users with actors, goals, states.

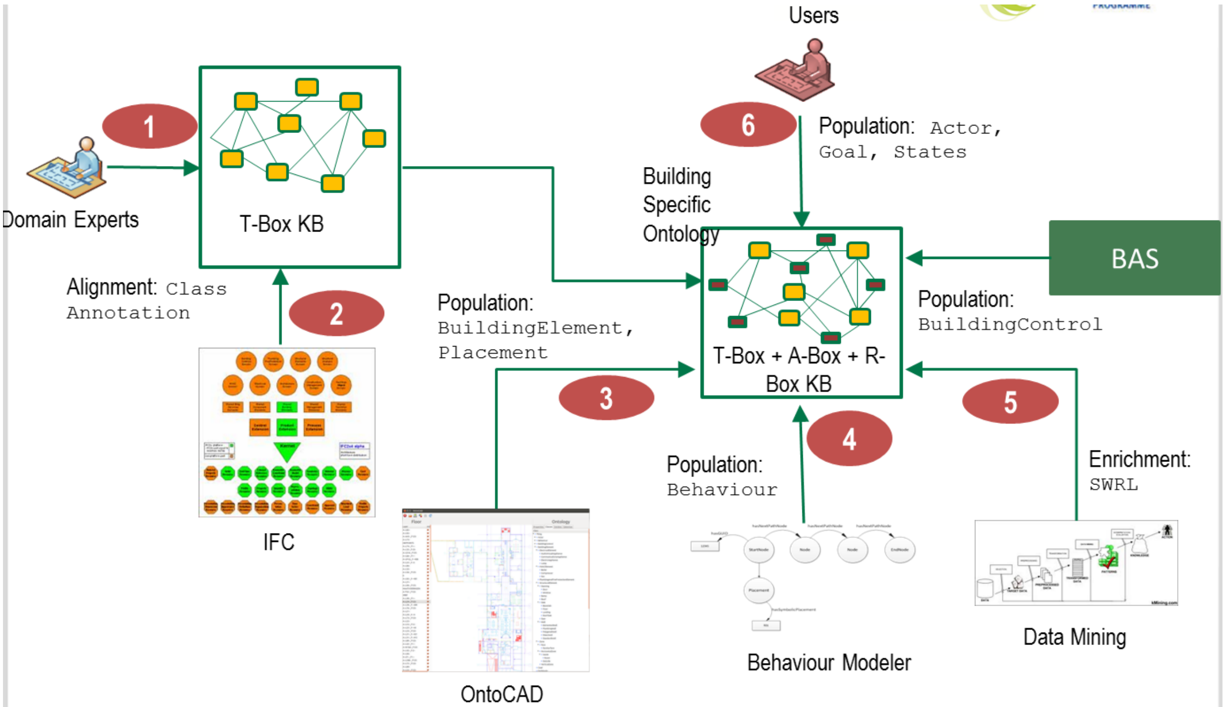

Figure 13: Ontology development process followed to obtain a building specific ontology (TBox + ABox + RBox).

The alignment of the generated knoholem.owl ontology with the IFC ontology currently occurs by means of annotations, not OWL primitives. After the alignment, the ontology is populated using the OntoCAD editor (see http://github.com/Victor-Haefner/ontocad/ and https://www.youtube.com/watch?v=uHoF5pKbxR8).

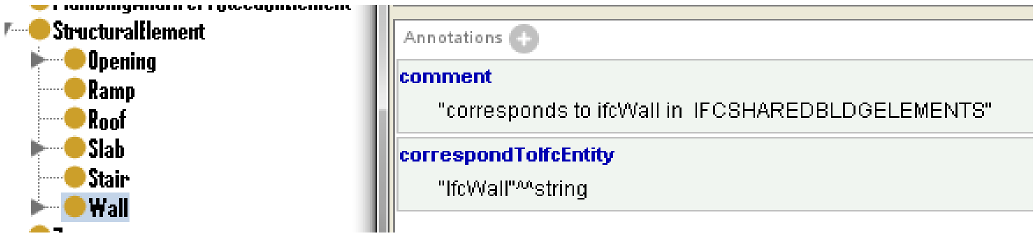

Figure 14: Alignment between the KnoholEM ontology and the ifcOWL ontology.

The RBox is built either in a model-driven (top down) approach (modelling all behaviour using a Business Process Model and Notation (BPMN), or in a data-driven (bottom up) approach (machine learning approach). The former is shown in Figure 13 with number 4; the latter is shown in Figure 14 with number 5. Using the data mining technique, it is possible to generate rules that indicate when certain sensor or metering values are anomalous when comparing to historical data. Such a rule is displayed in Figure 15.

```
IF LHT_DIG1_bri$hasAnalogValue$lux >1 AND
LHT_DIG2_hum$hasAnalogValue$percent>41.28 AND
LHT_DIG1_tem$hasAnalogValue$celsius>22.02 AND
LHT_DIG1_hum$hasAnalogValue$percent<=48.925 THEN
EM1$hasAnalogValue$kWh_disc=[0.98, 1.47]
```

$BrigtnessSensor$(LHT_DIG1_bri)$\wedge hasAnalogValue$(LHT_DIG1_bri, ?v1) $\wedge swrlb{:}greaterThan$(?$v$1, 1) $\wedge$
$HumiditySensor$(LHT_DIG2_hum)$\wedge hasAnalogValue$(LHT_DIG2_hum,?v2)$\wedge swrlb{:}greaterThan$(?$v$2,41.28) $\wedge$
$TemperatureSensor$(LHT_DIG1_tem)$\wedge hasAnalogValue$(LHT_DIG1_tem ,?v3)$\wedge swrlb{:}greaterThan$(?$v$3,22.02) $\wedge$
$HumiditySensor$(LHT_DIG1_hum)$\wedge hasAnalogValue$(LHT_DIG1_hum,?v4)$\wedge swrlb{:}greaterThanOrEqual$(?$v$4,48.925) $\wedge$
$EnergyMeter$(EM1)$\wedge hasAnalogValue$(EM1,?v5)$\wedge swrlb{:}greaterThan$(?$v$5,1.47)$\wedge BuildingElement(?be) \wedge isSensorOf$(EM1,$?be$)
$\rightarrow EnergyUsageAnomaly(?be)$

Figure 15: Rule obtained using data mining techniques, allowing to compare current metering values with historical records and indicating possible energy usage anomalies automatically.

Rules are similarly used in the operational phase of the building:

- to model **activities** based on the relation between building states and sensor values
- to detect **energy usage anomaly**. The rules are generated from historical metering data through data mining process
- to give **recommendation for controls** (e.g. setpoints). The rules are generated from simulation data.
- to model the **building states**
  - Simple state: on/off, occupied/unoccupied, etc.
  - Complex state: energy wasting

> Walter Terkaj - Ontology-based Factory Design & Management

Whereas ontologies can be used to model buildings, they can also be used to model factories, including the processes that have to be accommodated in such factories. In this presentation, a common model is presented for a factory, supporting different views according to different business processes in the life cycle of the factory (see Figure 16). The proposed solution (link specific models with the model of the factory as whole) hereby relies on an ontology-based approach (OWL). The result is a Virtual Factory Framework (VFF) as an *"an **integrated collaborative virtual environment** aimed at facilitating the sharing of resources, manufacturing information and knowledge, while supporting the design and management of all the factory entities, from a single product to networks of companies, along all the phases of the their lifecycles."*

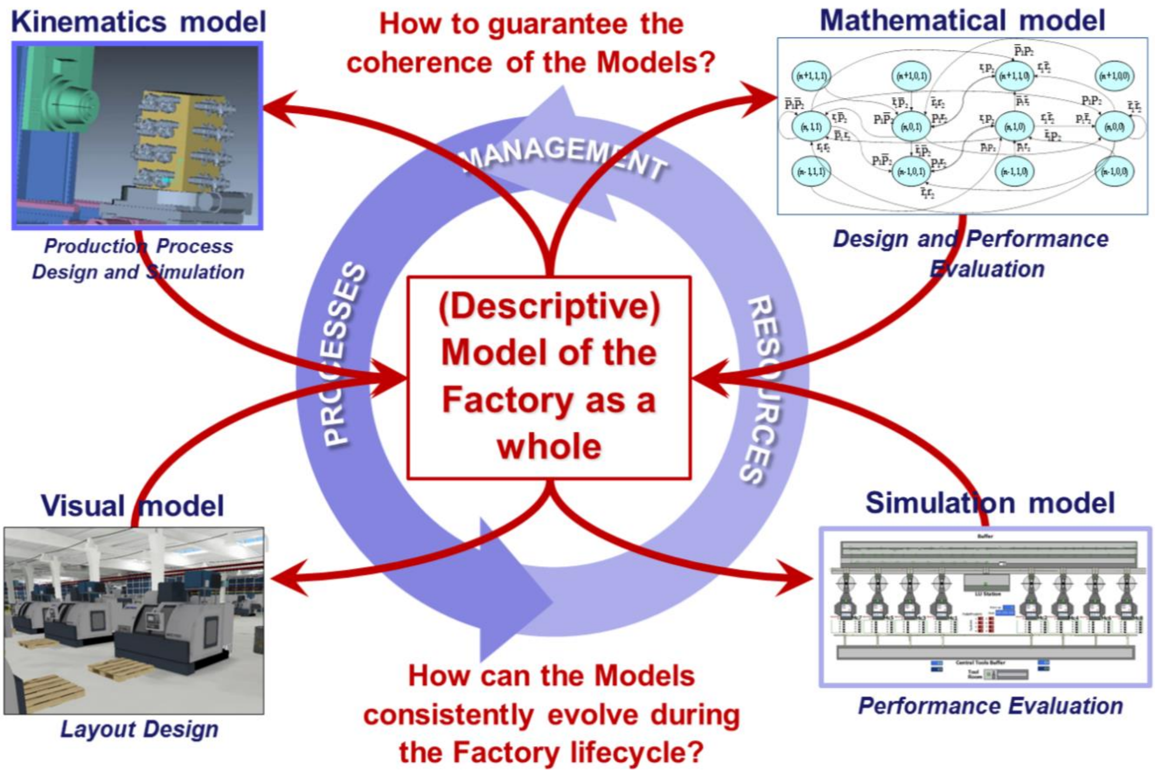

Figure 16: The model of the factory as a whole, which can serve as an input model for diverse simulations and evaluations.

At the core of the Virtual Factory Manager that implements the VFF is a Virtual Factory Data Model (VFDM), which is ontology-based and which aligns with existing technical standards, such as IFC and STEP-NC. The architecture of this VFDM is displayed in Figure 17, and includes the following ontologies:

- **StatisticsOntology**: basic concepts about probability distributions and descriptive statistics.
- **FSM**: basic concepts to model a finite state machine (http://people.cs.aau.dk/~dolog/fsm/).
- **FsmExtension**: extensions to fsm and integration with StatisticsOntology
- **IFC_ADD1**: ifcOWL automatically converted from IFC_ADD1.exp
- **ifcOWL_rules**: add class expressions to ifcOWL derived from WHERE rules in IFC_ADD1.exp
- **ifcOWL_extension**: integration of modules and general purpose extensions of ifcOWL
- **FactoryDomain**: specialization of ifcOWL with definitions related to products, processes, and systems
- **ISO14649-10**: fragment of the STEP-NC standard converted into an ontology
- **FixOnt**: fragment of the FixOnt ontology about modular fixtures (https://ontohub.org/repositories/fixture-design-ontologies/ontologies/9770/graphs)
- **DiscreteManufacturingDomain** module: represents the integration of four modules (ISO14649-10, FactoryDomain, FixOnt) and their further

specialization for the industrial domain of discrete manufacturing. Key point: integration of STEP-NC standard with IFC; integration of FixOnt ontology with the IFC standard; definition of the physical and logical structure of a pallet.

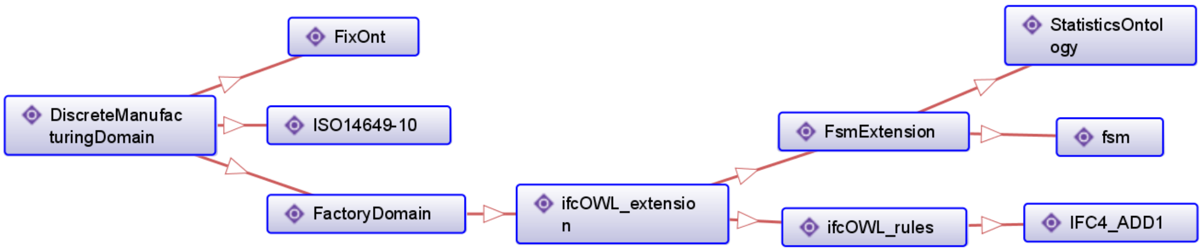

Figure 17: The Data Model architecture that underlies the VFDM.

This central data model is used by a number of digital factory tools, as displayed in Figure 18. This is implemented using the VfConnectorLib, which is a set of high-level C++ programming libraries that is based on the Redland RDF libraries and contains:

- **RdfStorageLib**: functionalities to manage, parse, create, modify and serialize complex factory projects consisting of individuals contained in one or more ontologies.
- **VfdmCppLib**: advanced extraction, generation and handling of owl individuals defined according to the VFDM. This library is automatically generated and maps the OWL classes and restrictions of the VFDM to C++ classes and methods (get, set, delete)
- **VirtualFactoryLib**: high-level functionalities to get and set complex and common data structures consisting in chains of individuals (e.g. definition of unit of measurement, object placement, shape representations, etc.).

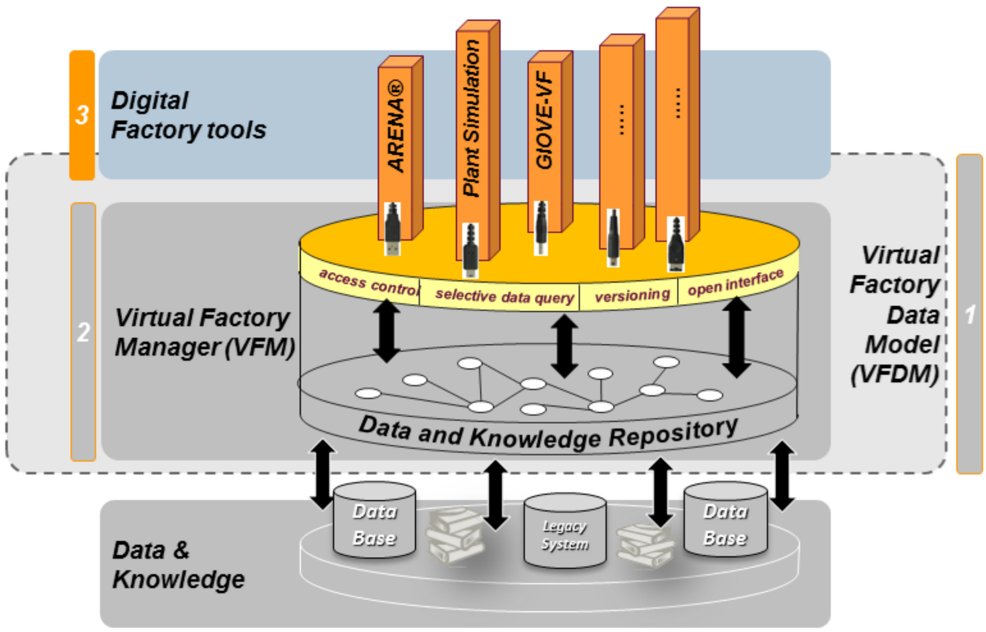

Figure 18: The VFF, with the digital factory tools relying on the same core virtual factory data model.

One example digital factory tool is GIOVE-VF, which is a virtual reality collaborative environment that supports factory layout design based on all the outlined ontologies.

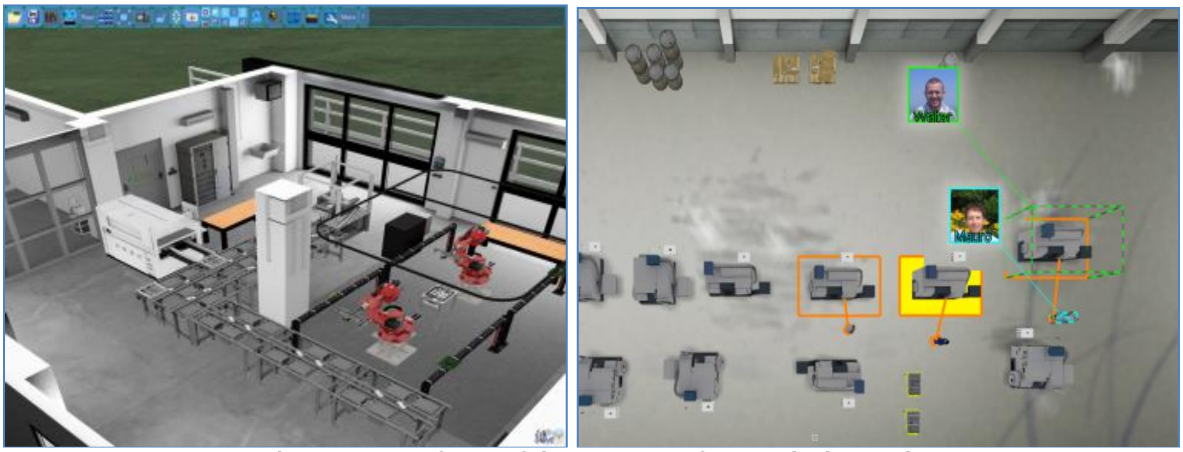

Figure 19: Interfaces of the GIOVE-VF factory design tool.

A relevant tool to manage the underlying ontologies and instances is the ONTO-GUI tool (Fig. 20), which provides a graphical user interface for managing the ontologies that contain factory libraries or projects. This tool allows to test and 'rapid prototype' ontologies (restrictions, what object properties are there, inconsistencies?), create instances following the ontology, and so forth.

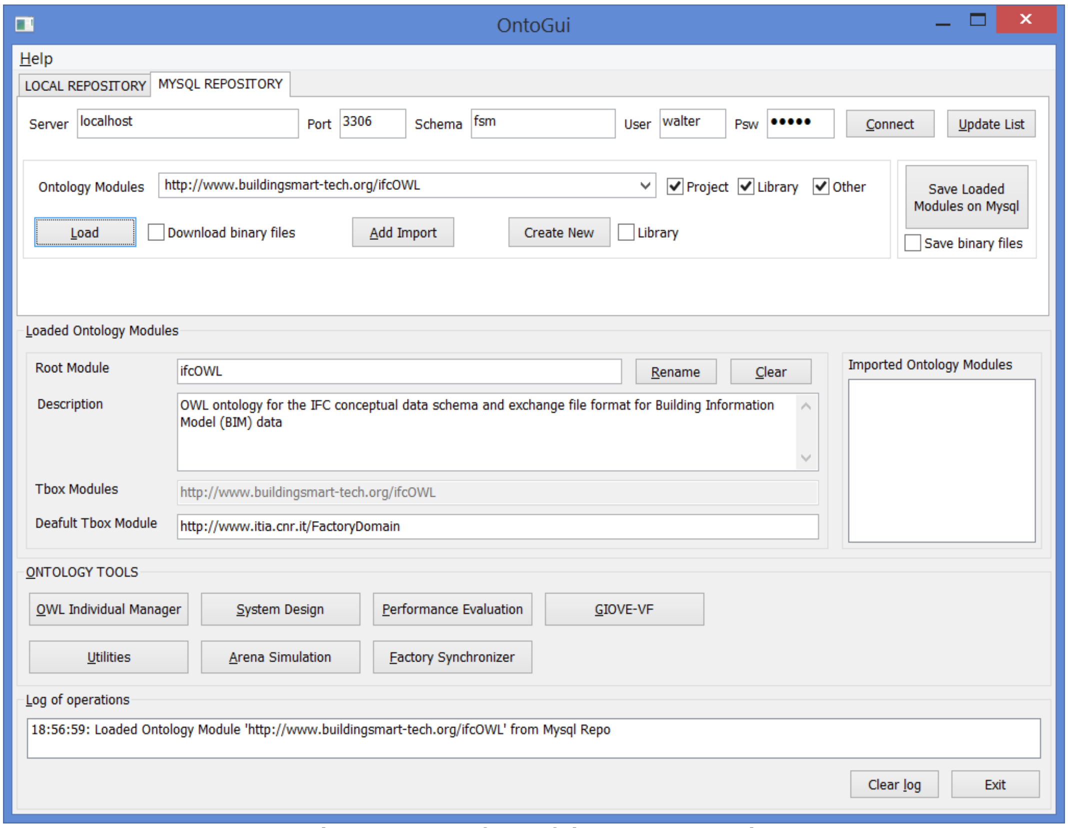

Figure 20: Interface of the OntoGUI tool.

> *Leif Granholm - OGC activities regarding semantic web*

- Presentation confidential

> *Daan Oostinga - Use cases for linked data*

- Presentation confidential

> *Michel Böhms - Linked Data in VCON*

As part of the VCON project (http://www.rws.nl/v-con/), the usage of linked data and semantic web technologies is tested for information management in the construction of roads (infrastructure). In such projects, there are often multiple datasets available that represent the same real-life objects. Each dataset hereby typically represents a certain context or view (e.g. realization, asset management, traffic management). The VCON project proposes to link these datasets into one common set of linked datasets (see Figure 21). One of these datasets is an ifcOWL dataset.

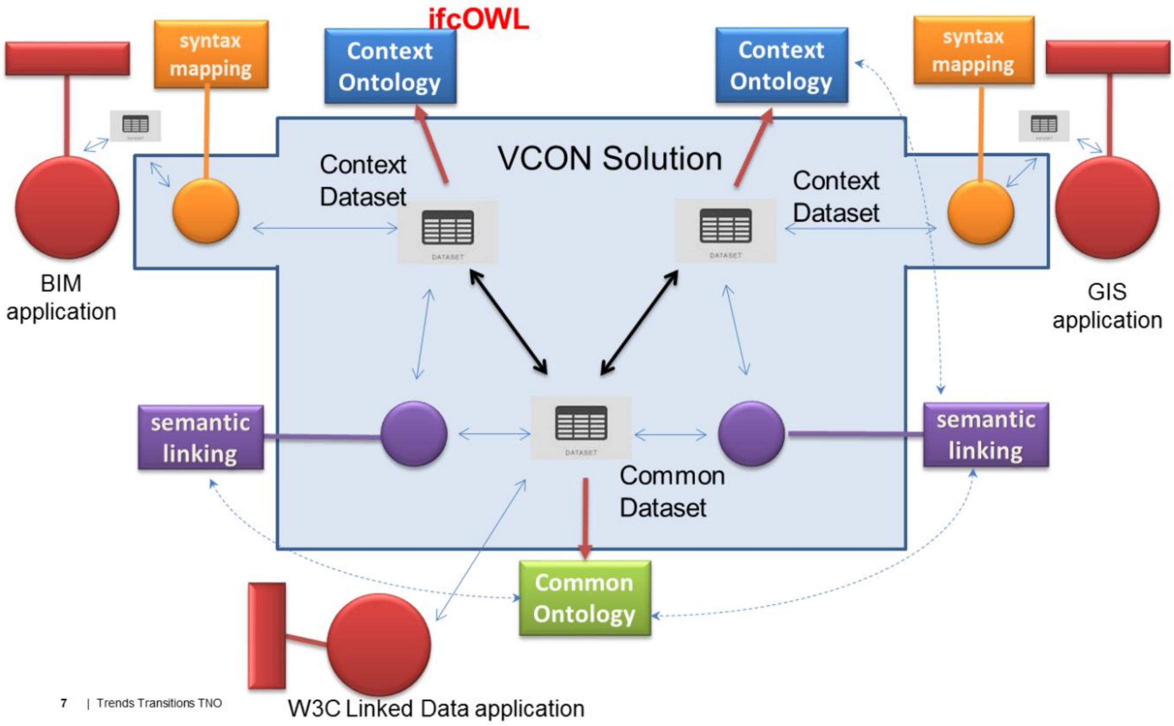

Figure 21: Schema displaying the way in which the VCON solution proposes to link a BIM application, a GIS application, and a W3C Linked Data application.

In this proposal, the main features of linked data technologies ("new school") are explored, instead of sticking to the tradition technologies ("old school"), as displayed in Figure 22. The main features of linked data technologies are:

- Evolution-based
- Short cycle times
- Flexible / dynamic
- Decentral solution
- Web-based

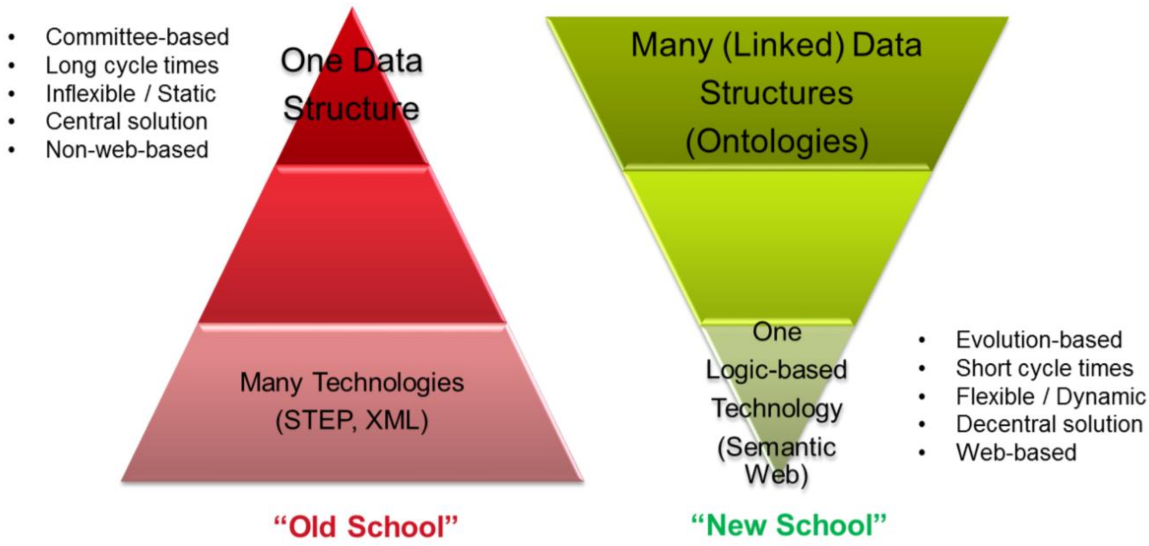

Figure 22: Opposition between the traditional techniques (left) and the new techniques (right).

The project hereby aims at building 5-star linked data, as displayed in Figure 23. In other words, data is represented using W3C open standards, such as RDF, RDFS, OWL2, SKOS, and this data is linked to other data. In realizing this scenario, the project delivers a modelling guide, which is based on CMO, and a linking guide.

Figure 23: 5-star linked data for the VCON project.

For linking concepts, four relevant 'dimensions' of linked data need to be taken into account:

1. Information can be represented using three key meta-concepts:
   - Classes (owl:Class)
   - Properties (owl:DatatypeProperty & owl:ObjectProperty)
   - Individuals (owl:NamedIndividual)
2. Restrictions can be group according to 12 Venn diagrams:
   - 5 for classes,
   - same 5 for properties,
   - 2 other for individuals
3. Class representations can be grouped in three semantic levels, depending on the restrictions that apply.
4. A distinction can be made between asserted and inferred data

As an example, Figure 24 shows the 5 Venn diagrams with basic linking situations for properties. In terms of the semantic levels that can be used to group class representations, the following three semantic levels can be considered:

- L1: Classes without restrictions
- L2: Classes with only ‘necessary’ restrictions
- L3: Classes with ‘necessary and sufficient’ restrictions

Semantically complete ontologies typically have only L3 classes, whereas more flexible ontologies typically have a mixture of 1/2/3-type classes. In practice, an ontology is typically a mixture with many L1 classes, some L2 classes and often no L3 classes, making them “semantically not that strong”.

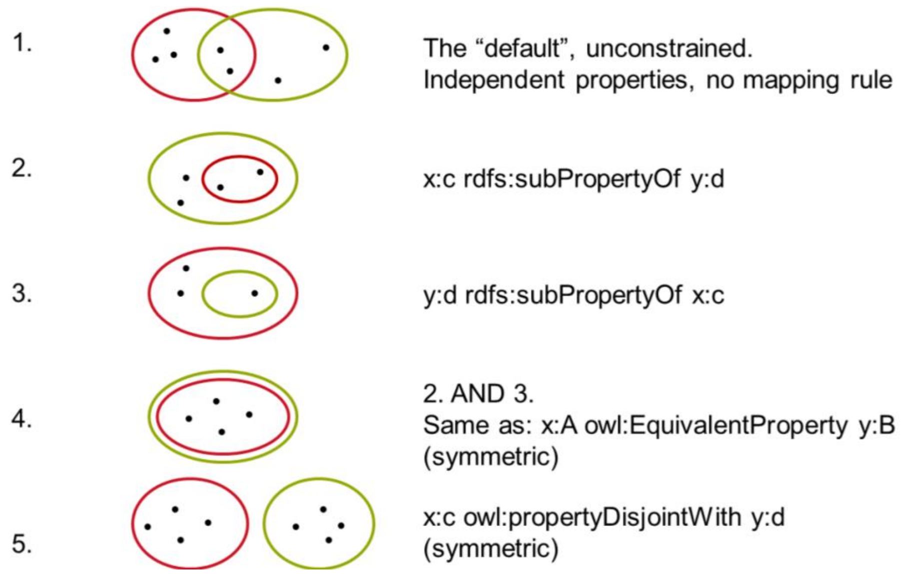

Figure 24: 5 Venn situations for properties.

If two ontologies or datasets are linked, the matrix that is displayed in Figure 25 results. This matrix displays 9 situations. Of these 9 situations, situation 1 is the situation in which the resulting ontology is most ‘underspecified’, which is considered to be (1) worst case in terms of expressiveness, but also (2) very practical for developing end user applications. Situation 9 is the situation in which the resulting ontology is most ‘overspecified’, which is considered to be (1) best case in terms of expressiveness, but also (2) very unlikely to happen.

x/y	L1	L2	L3
L1	Situation 1	Situation 2	Situation 5
L2	Situation 3	Situation 4	Situation 7
L3	Situation 6	Situation 8	Situation 9

Figure 25: Combining two datasets with different levels of semantics (L1, L2, L3).

> Tarcisio Mendes de Farias, Ana Roxin and Christophe Nicolle - IfcWoD, Semantically Adapting IFC Model Relations into OWL Properties

There are two main reasons why one would want to use Linked Data (LD) in the context of BIM. On the one hand, it enhances BIM interoperability, and on the other hand, it provides a dynamic information usage via RDF rather than the more static file-based information usage. A lot of the existing work has focused on making the Industry Foundation Classes (IFC) data model available for semantic web technologies, so that linked data principles can be applied. However, because of some of the core conceptual differences between EXPRESS and OWL (OWA vs. CWA, object-oriented modelling versus graph-based modelling), the constructs that are now becoming available in the ifcOWL ontology do not always make that much sense from a semantic web or linked data point of view, thereby hindering the targeted intuitive building information manipulation.

For example, one can wonder whether order is really important for justifying the use of OWL Lists in ifcOWL (Fig. 26). By using this construct, data access becomes non-intuitive and difficult. One might want to use direct, semantically more meaningful connections, as is displayed in Figure 27 for `IfcCartesianPoint`.

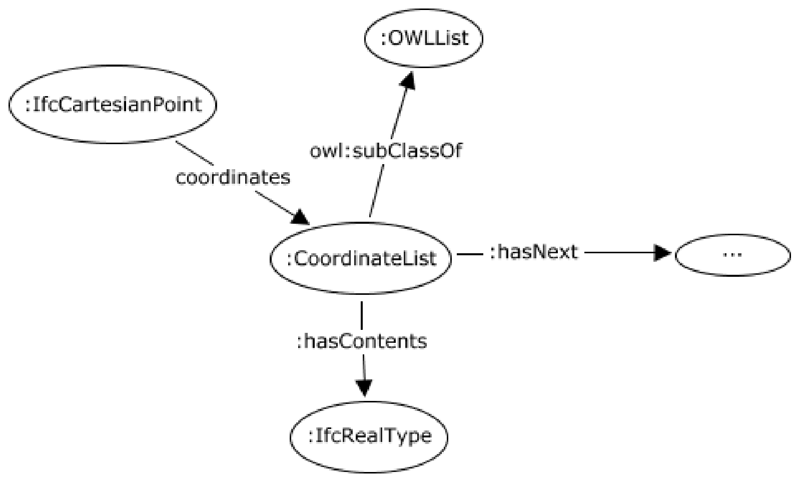

Figure 26: Using the OWL list for representing ordered lists in OWL.

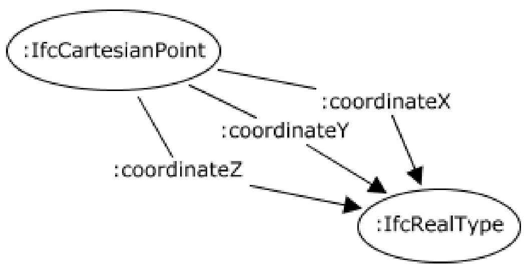

Figure 27: Alternatively using direct properties pointing to the values of the list.

As a second example, the `IfcRelationship` and its subtypes might be better translated in entirely different semantic graph structures. Instead of using an intermediate class instance (`IfcRel…`) that lies in the middle of an *n-to-n* relationship, it might be worthwhile to use direct properties pointing directly from *n* class instances to *n* other class instances. This could result in the sort of shortcut that is displayed in Figure 28 for the IFC attribute `OperatesOn`. These and other improvements (e.g. links between IfcBuildingElements and IFC Property Sets). should be considered in order to obtain better usable IFC instance graphs.

Figure 28: Replacing IfcRelationship instances with direct property relations between the respective class instances.

To address this situation, this research proposes to build IfcWoD, an IFC Web of Data Ontology. This IfcWoD is semi-automatically conceived and takes into account the IFC EXPRESS model, ifcOWL, and PSD (Property Set Defintions). The result is an IFC ontology that:

- Allows enhancing reasoning
- Provides easier query writing and relationship understanding
- Reduces query execution time

IfcWoD is not conceived as another ifcOWL version. Rather, it is conceived as an additional to ifcOWL which simplifies query writing and improves query response time for retrieving building data. Figure 29 gives an example of the improvements in terms of query writing and relationship understanding.

	Querying solely with IfcOWL terms		Querying with IfcWoD terms
Q1:	*SELECT ?wall WHERE {*	**Q1':**	*SELECT ?wall WHERE {*
	?wall rdf:type ifcowl:IfcWall;		*?wall rdf:type ifcowl:IfcWall;*
	ifcowl:IsDefinedBy_of_IfcObject ?rel.		***ifcwod**:isDefinedBy_IfcObject ?pSet.*
	?rel ifcowl:RelatingPropertyDefinition ?pSet.		*?pSet **pset_WallCommon**:isExternal ?val.*
	?pSet ifcowl:HasProperties_of_IfcPropertySet ?p.		*?val ifcowl:has_boolean "true"^^xsd:boolean.*
	?p rdf:type ifcowl:IfcPropertySingleValue;		*}*
	ifcowl:Name_of_IfcProperty ?name.		
	?name ifcowl:has_string "IsExternal"^^xsd:string.		
	?p ifcowl:NominalValue ?val.		
	?val ifcowl:has_boolean "true"^^xsd:boolean. }		

Figure 29: Query writing in ifcOWL only versus ifcOWL+ifcWoD.

> Léon van Berlo and Pim van den Helm - IFC model checking with N3: a proposal for a semantic rule checking language for BIM

IFC (BIM) model checking is popular (and still growing). In the current situation, there are however only limited and custom software implementations (Solibri, BIMSight, etc.). These implementations have a user friendly GUI, but rules are not exchangeable. In this project (https://github.com/openBIMstandards/modelcheckN3), we aim to create an exchangeable open standard (i.e. software independent) for rule checking and model checking. To achieve this aim, we investigate the usage of semantic web technologies, in combination with ifcOWL. Because of the logical basis of ifcOWL, new possibilities for semantic reasoning lie within reach, which could surpass the capabilities of currently available software implementations. The solution that we tested relies on Notation-3 (N3) and the Euler Euler Yet another proof Engine (EYE - http://eulersharp.sourceforge.net/). The following criteria are central to the measure the success of our test:

- As low as possible threshold for end users (architect and engineers): rules should be human readable / usable.
- As less as possible “schema”-dependent queries, because user driven requirements seldom map in a straightforward manner to an IFC schema.

In this presentation, the following simple question / query is considered: *“Check if all doors on the 3rd floor, with the colour ‘red’, are more than 900mm wide”*. Many of the elements in this query are ambiguous and can be interpreted in a number of ways. This is why a simple user interface is required, so that the end user can precisely model his query and make it unambiguous.

With N3, it is possible to create new graph-based queries which refer to ifcOWL triples and additional statements (Fig. 30). Combining such queries with sample building models in ifcOWL and the EYE inference engine, results in a simplified graph that is usable for ‘simple’ N3 queries. This lowers the threshold for domain experts to use the power of semantic web technologies for validating/checking of IFC models.

```
{
  ?Object a ifc:IfcDoor;
    ifc:HasColour "red".
}
=>
{?Object :found "red-door" }.
```

Figure 30: Example query in N3 rule syntax that shows which kind of queries should eventually be supported.

The main issue at the moment is that there is no final (standard) version of ifcOWL yet.

The Technical session

1. Discussion items

- URI design
    - How could we design URIs in a manner that promotes linking using the Linked Data approach? That is, that the linksets would remain (mostly) valid even when the new versions of the models are published. The goal is that a same object would always have a same URI and same URI would always mean the same object.
- Roundtrip or not
    - Do we need a reverse conversion (from RDF to IFC)?
    - What kinds of use cases are there for roundtripping?
- Model evolution
    - How to handle versioning with the Web of Data technologies/representations?
    - Use cases
- Dataset description
    - How to make datasets discoverable?
    - For instance, description about version, LOD, discipline, dataset relations. ...
    - What is the use of existing vocabularies? (e.g., VoID)
- Openness
    - What parts of data are open to who?
    - How to control the access?

2. Remarks on the scope and purpose (Nam Vu Hoang)

- General questions:
    1. What are main **purposes** of ifcOWL?
    2. Which **technologies & trends** should we take into account?
    3. Which **ifcOWL layers** should we focus on and keep in mind?
    4. What kind of **type info** should be included into each ifcOWL layer?
    5. What are **criteria for comparing** alternative solutions?

- Specific problems:
    1. What are roles of domain/range restrictions? Do we need them?
    2. How should Simple Datatypes be converted?
    3. How should Declared/Defined Datatypes be converted?
    4. How should Select Datatypes be converted?
    5. How should Enumeration Datatypes be converted?
    6. How should Entity Datatypes (without properties) be converted?
    7. How should Aggregated Datatypes be converted?
    8. How should Entity Datatypes (with properties) be converted?
    9. How should inverse attributes be converted?

10. How should entity key (unique) attributes be converted?
11. How should different kinds of individuals in ifcRDF be named?
12. How should ifcOWL layers be related with each other?

- Main criteria:
    1. Respect a predefined OWL profile: OWL2 DL
    2. Respect data (structure) consistency
    3. Have a decidable ontology - allow inference, tradeoff between expressivity and reasoning
    4. Respect the IFC specification / the EXPRESS schema

- The scope for the meeting:
    - We agree for this meeting that we look for 1 ifcOWL Extended ontology in OWL2 DL
    - It would be a good idea to 'modularise' this ifcOWL ontology, so that we can support smaller modules in other profiles (OWL2 EL / OWL2 RL / OWL2 QL).

3. Discussion items and agreements made

Issue #1: Domains and ranges to be included or not?

1. No domains/ranges are specified in the ontology
2. OWL 2 restrictions + d/r on simple datatypes
3. putting domain/range & property restrictions whenever we can

=> Decision: we vote for option 3. This was a tight vote. There is a great argument for easy names.

Issue #2: PropertyName - Consistent use of long names or only use of long names when required (inconsistent)?

=> Decision: consistent

Issue #3: PropertyName - Camelcase long names or Underscore long names?

=> Decision: underscore

Issue #4: PropertyName - First Class Name, then Property Name?

=> Decision: PropertyName_ClassName

Issue #5: PropertyName - Exclude "Ifc" from all names (classes, properties)?

=> Decision: Keep "Ifc" included

Issue #6: PropertyName - name_of_IfcRoot | name_IfcRoot?

=> Decision: name_IfcRoot

Issue #7: Use of "Grouping" SubProperty relations (Property "Name") or not?

=> Decision: no SubProperty relations

Issue #8: How to convert LISTs and ARRAYs?

1. OLO pattern (including index of LIST items)
2. Drummond pattern (excluding index of LIST items)

=> Decision: Drummond pattern (because it is also used by OntoSTEP)

Issue #9: How to convert SETs?

1. Follow the LIST pattern
2. Use non-functional object properties

=> Decision: Use non-functional object properties

Issue #10: Inverse attributes?

1. included in conversion when possible
2. not included in conversion

=> Decision: included in conversion when possible

Note: what cannot be included in the conversion?

- An attribute has two or more INVERSE attributes. This is, for example, the case of attribute RelatedDefinitions of entity IfcRelDeclares. This attribute has two inverse attributes: HasContext of entity IfcObjectDefinition and HasContext of entity IfcPropertyDefinition. If all these INVERSE attributes were converted to object properties in ifcOWL, then a reasoning engine would infer that the two HasContext object properties are equivalent. Moreover, other inferences would lead to say that some classes are equivalent to owl:Nothing.
- A regular attribute or its INVERSE attribute has a LIST or an ARRAY as its range. Given the particular conversion pattern needed for ordered lists, if the INVERSE attributes were converted to object properties, then there would be a mismatch between the range of an object property and the domain of its inverse. Therefore, a reasoning engine would infer that the range of the object property is equal to the intersection of two disjoint classes. An example of this case is represented by attribute Addresses of entity IfcPerson and attribute OfPerson of entity IfcAddress.

Issue #11: NUMBER simple datatype is considered as an INTEGER or as a REAL?

=> Decision: REAL

Issue #12: REAL simple datatype is considered as an xsd:double, xsd:decimal or xsd:real?

=> Decision: xsd:double

Issue #13: LOGICAL simple datatype is considered as an xsd:boolean, or as an enumeration of TRUE, FALSE, and UNKNOWN?

=> Decision: enumeration of TRUE, FALSE, and UNKNOWN

Issue #14: BOOLEAN simple datatype is considered as an xsd:boolean, or as an enumeration of TRUE and FALSE?

=> Decision: enumeration of TRUE and FALSE (because we can then reuse the same concepts that are available in the LOGICAL datatype.

Issue #15: How to declare ENUM datatypes?

1. using owl:oneOf
2. using rdf:type only (when owl:oneOf is used, a reasoner infers the "rdf:type" relation)

=> Decision: using rdf:type only

Issue #16: How to declare SELECT datatypes?

1. using owl:unionOf
2. using rdfs:subClassOf only (when owl:unionOf is used, a reasoner infers the "rdfs:subClassOf" relation)

=> Decision: using rdfs:subClassOf only

Issue #17: What namespace structure do we use?

1. http://www.buildingsmart-tech.org/ifcowl/IFC4
2. http://www.buildingsmart-tech.org/def/IFC4
3. Other

=> Decision: http://www.buildingsmart-tech.org/ifcowl/IFC4

For the other ontologies, this results in:
- http://www.buildingsmart-tech.org/ifcowl/IFC4
- http://www.buildingsmart-tech.org/ifcowl/IFC4_ADD1
- http://www.buildingsmart-tech.org/ifcowl/IFC2x3
- http://www.buildingsmart-tech.org/ifcowl/IFC2x3_TC1
- http://www.buildingsmart-tech.org/ifcXML/IFC4

Issue #18: Naming individuals / instances?

=> Decision: following existing best practices.

Note, these are best practices:

- Data on the Web Best Practices: http://www.w3.org/TR/2015/WD-dwbp-20150625/
- Data Identification: http://www.w3.org/TR/2015/WD-dwbp-20150625/#DataIdentification
- Content Negotiation - http://www.w3.org/Protocols/HTTP/Negotiation
- Use unique namespaces (eventually including line number)
- Identify key uses cases and for each use case define key recommendations (e.g. BIM, versioning, etc.)
- http://www.w3.org/TR/cooluris/
- http://www.w3.org/2001/tag/doc/URNsAndRegistries-50.html
- http://www.w3.org/Provider/Style/URI
- http://www.w3.org/TR/2003/NOTE-chips-20030128/
- http://www.w3.org/2001/tag/doc/URNsAndRegistries-50.html

Issue #19: EnumName - what naming to use?

1. follow the same approach as for object properties
2. adopt simple naming strategy

=> Decision: follow the same approach as for object properties

Issue #20: Which license to use?

=> Decision: CC-BY if possible, depending on the BuildingSMART license

The Tutorial Sessions

> Kris McGlinn - Converting Tabular Data to RDF using CSVW

Lots of structured data on the web is currently in formats which provide little description of the actual data (csv, xml, json). Where it is provided, the descriptions themselves are often only meant for the next developer when implementing a parser for that data. CSVW is a simple way of providing semantic meaning for CSV values in a JSON document1. CSVW allows for the conversion of data into a single format (RDF), meaning that it can be easily interlinked with other relevant data, and also developers no longer must write multiple types of structured queries as SPARQL can be used for all.

In this tutorial you will learn to convert tabular data into RDF and query that data to meet a specific use case in the Energy Efficient Building domain. You will look at some sample data sets (simple csv files) and you will write a CSVW file adding additional semantics to your tabular data. Next you will convert your CSV data into RDF. You will then link that data to another open linked data source. Finally, you will conduct some simple visualisations of the data.

The tutorial takes the form of 5 steps. Step 1 is to analyse your data sets and uncover what semantics you wish to include to describe your tabular data. Step 2 is to create a CSVW description of your tabular data (see Figure 31). Step 3 is to parse the data, output an RDF file and examine the RDF file. Step 4 is to write some SPARQL queries to access the data. Step 5 is to examine an application which links your tabular data to another data set.

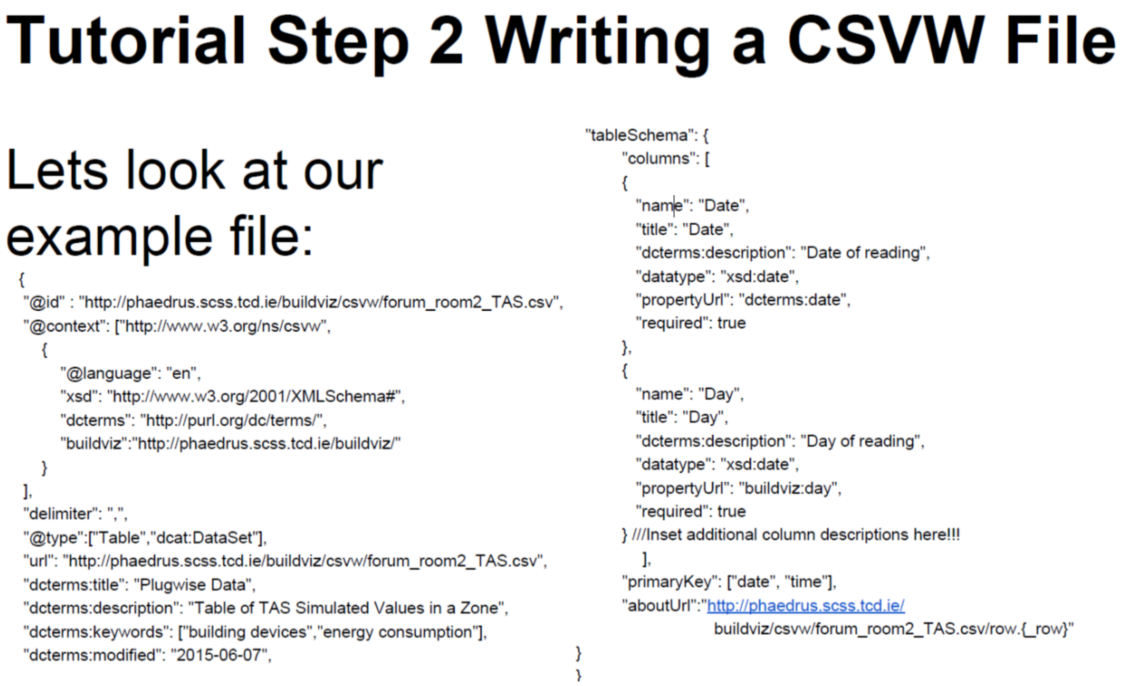

Figure 31: Example CSVW file that interprets CSV data and outputs the data in an RDF graph.

> Nick Kaklanis - Converting OWL to JSON-LD

Ontologies have been successfully applied as a semantic enabler of communication between both users and applications in fragmented, heterogeneous multinational business environments. In this tutorial we discuss the underlying principles, their current implementation status, and most importantly, their applicability to problems in the building information modelling domain. Special focus is given on the development of an ontology for the building and construction sector based on the industry foundation classes.

In this tutorial you will learn how ontologies can be used for the meta-modelling of building and construction process. The most interesting research efforts towards this direction will also be analysed.

The following resources are used throughout this tutorial:

- Ifcowl: A case of transforming express schemas into ontologies: http://dl.acm.org/citation.cfm?id=1520339
- Linked Building Data: http://linkedbuildingdata.net/
- IFC4_ADD1.owl: http://linkedbuildingdata.net/resources/IFC4_ADD1.owl
- ifcOWL developments: https://www.w3.org/community/lbd/wiki/images/9/9a/IfcOWL_intro.pptx

The second part of this tutorial will connect with the tutorial by Kris McGlinn, in the sense that building and construction metadata in the OWL format will be converted into a JSON-LD format, and vice versa. This is specifically applied to the IFC4_ADD1.owl ontology referenced above.

> Matthias Weise, Pieter Pauwels - Selection of IFC subsets using ifcOWL and rewrite rules

The use of Linked Data in Architecture and Construction is still faced with a set of challenging questions. Besides identification of business cases the development of best practices is urgently needed to get started with publication of building data. This presentation is focusing on the step of extracting relevant data from an existing BIM data set. We will clarify the use of existing technologies and, as a possible alternative, will examine how similar results can be achieved based on semantic web technologies. Both approaches (see Figure 32) will be compared and evaluated.

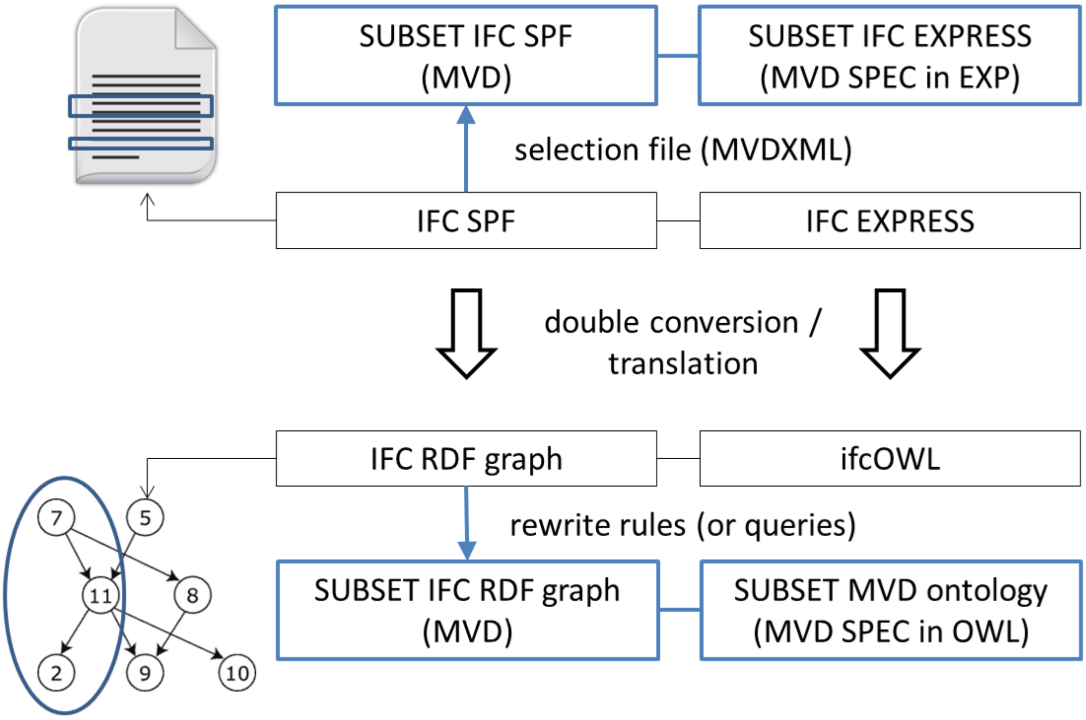

Figure 32: Overview of the scope for this tutorial presentation, showing the regular EXPRESS-based subset selection via MVDs (top) and the alternative OWL-based subset selection via rules (bottom).

IFC is an international open standard for sharing building data and is being a main driver of the BIM approach. The aim of BIM is to collect all relevant data of a building in a consistent and redundant free manner. However, working with IFC typically means to use a subset of all BIM data. For instance, when an architect communicates information from his Building Information Model (BIM) to an HVAC system designer, then this HVAC system designer typically does not require the full BIM model, but only parts of the data. Thus, IFC data exchange is scoped according to the purpose of its use. Scoping of BIM data is not only interesting to reduce the file size, it is also used to control the data flow. Similarly, when publishing a BIM model in the web, such a technique will become useful to avoid privacy and security issues, to improve performance or to anonymize the BIM model.

The objective of scoping BIM and IFC has been a key objective for standardization efforts by buildingSMART that have been put in the concepts of Information Delivery Manuals (IDMs) and Model View Definitions (MVDs). With mvdXML a formal schema exists that enables to specify required IFC subsets and therefore can be used for scoping of BIM data. Also, a tool called IFC Documentation Generator is provided by buildingSMART free of charge, which supports mvdXML developments. All this is a sound foundation that can be used in a data publication scenario.

Alternatively, there is an option that semantic web technologies can improve this process. Two kinds of improvements might be available. First, semantic web technologies might allow to define MVDs in a much more intuitive and faster manner, both at schema level (TBox) and at instance level (ABox), mainly because of the standard functionalities provided by these technologies, such as the SPARQL query language and inference support. Second, using the logical basis of semantic web technologies (Description Logics – DL), it might be possible to rapidly and trustworthy check the validity of an MVD model. If a generated MVD subset is compliant with the

MVD ontology defined in the Web Ontology Language (OWL), which can be checked automatically, more reliable MVD subsets might be within reach.

The presentation first introduces the existing buildingSMART technologies for IFC scoping. As an example we will use the data requirements that have been defined for the Indoor Navigation use case. A main aspect is to explain the structure of mvdXML and the idea of having functional units that can be reused in different application contexts. We will show how such mvdXML specification can be used to create a valid IFC subset containing required data only. Finally, this subset data can be mapped to ifcOWL RDF for data publication purposes.

The same will be done with help of semantic web technology. In that scenario, the entire IFC file will be converted into an ifcOWL RDF graph first. According to the mvdXML scope specification a number of transformation rules is then generated and applied that reformulate the required concepts from the entire RDF graph and generate a distinctly scoped graph. The usage of SPARQL queries is also explored as an alternative.

We will discuss our experiences and results about the two approaches during a joint presentation at the LDAC workshop.

Resources:

- Download ifcDoc tool (for Windows only): http://www.buildingsmart-tech.org/specifications/specification-tools/ifcdoc-tool/ifcdoc-download-page
- Download IFC4 baseline definition: http://www.buildingsmart-tech.org/specifications/specification-tools/ifcdoc-tool/ifcdoc-baselines
- IFC example files (NIBS clinic): http://www.nibs.org/?page=bsa_commonbimfiles
- IFC viewer: http://www.iai.fzk.de/www-extern/index.php?id=1138&L=1
- MVDXML model checker by Chi Zhang et al.
    - http://www.itcon.org/cgi-bin/works/Show?2015_2
    - https://github.com/opensourceBIM/mvdXMLChecker

> María Poveda-Villalón, Pieter Pauwels - Introduction to Ontological Engineering: Tutorial for ifcOWL

As semantic web technologies and linked data techniques are of increasing importance for the do-main of architecture, engineering and construction (AEC), also the ontologies that are used by these technologies gain in importance and relevance. An ontology is generally defined as *“a formal, explicit specification of a shared conceptualization”*, after Studer et al.. Because not only people, but also technologies rely on these ontologies, it is important that they are well conceived and well developed. In this tutorial session, we will therefore give an overview of the main principles behind ontological engineering. We will hereby rely on the principles that are developed as part of the NeOn methodology for ontology engineering. A schema of this methodology is provided in Figure 33.

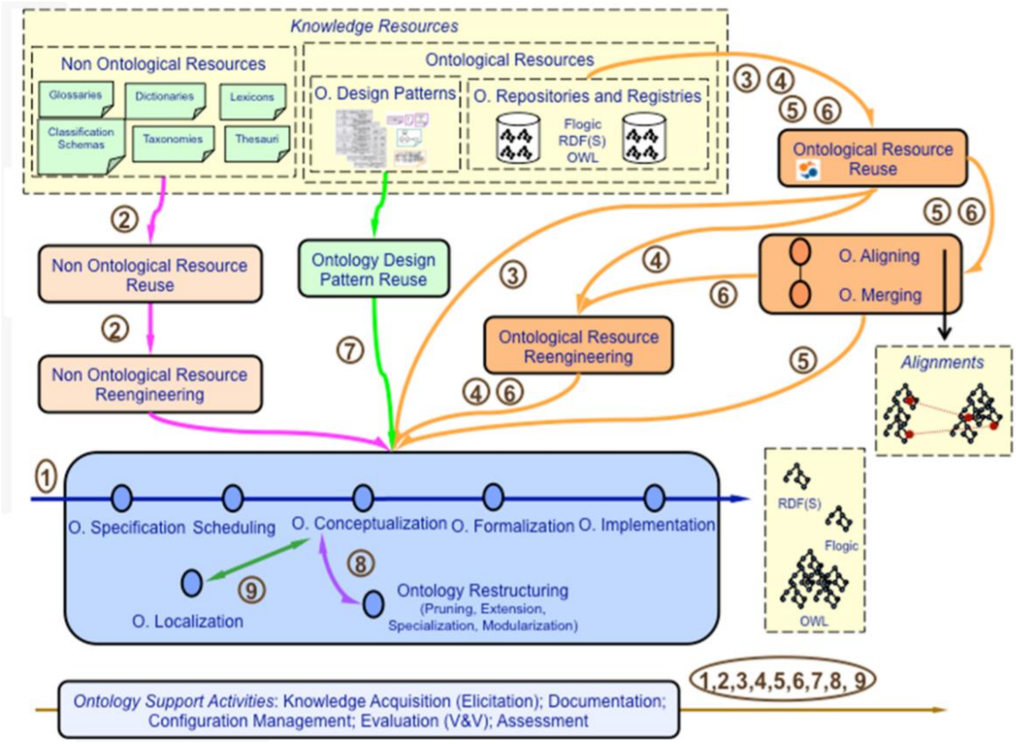

Figure 33: NeOn ontology development scenarios (original image by Mari Carmen Suárez Figueroa).

Our tutorial session will go through all steps in the NeOn methodology. We will hereby look into the following phases:

1. Requirements definition
2. Terms extraction
3. Ontology conceptualization
   a. Initial model drafting
   b. Detailed model drafting
4. Ontology implementation
   a. Non ontological resource transformation
5. Ontology evaluation

In presenting these phases, we will use the ifcOWL ontology as a reference example and point out where it is useful to make which kind of decisions. This is of particular relevance in the ontology implementation phase. For example, we indicate the relevance of using smaller ontologies (ifcOWL core, ifcOWL standard), as proposed by Nam Vu Hoang, and importing these as subsets in more expressive ontologies (i.e. ifcOWL Extended); we outline a number of guidelines and recommendations regarding the creation of persistent URIs; we summarize best practices regarding the inclusion of metadata and license information; and so forth. This on-topic and tutorial-like overview should improve the ontological engineering skills of the audience present at this LDAC presentation.

Action items

> Proposal of recommended ifcOWL version in the BuildingSMART Technical Summit in Singapore (mid-October)

The development and maintenance of a recommended ifcOWL ontology will be in hands of / managed by a standardisation body the Technical Room of the buildingSMART organisation.

> Follow-up actions and events

- Capture remaining points from ballots on ifcOWL specification
- Help Nam to implement configurations for the drum tools, Pieter and Walter are continuing their implementation so we can also validate/cross-check. Nam's tool will be the reference.
- Maria lends support with validation suite http://oops.linkeddata.es/
- Maria, Nam, Pieter, Seppo, Jakob (and Walter?) chip in documenting the ontology document to be published on buildingSMART
- publish on server including content negotiation. (Pieter, TU/e happy to help and/or host)
    - Set up pubby or similar
    - get the subdomain / redirect from buildingSMART
    - make the official URI resolvable
- Export labels and description for each IFC element in some tabular format from http://www.buildingsmart-tech.org/ifc/IFC4/final/html/index.htm
    - Jakob has an old html doc scraper and XML file of 2x3_TC1
    - We might use the ifcDOC format to harvest (Chi Zhang, Thomas Krijnen?)
- Deadlines:
    - Internal deadline: Mid-September.
    - Official Deadline: buildingSMART meeting in Singapore (mid october)
- Kris updates the Use case wiki. Everyone chips in.
- Look into the license that should be used for the ontology!
    - Suggestion: http://creativecommons.org/licenses/by/4.0/
    - or http://creativecommons.org/licenses/by-nc-sa/2.0/
- http://purl.org/dc/elements/1.1/source → for pointing to the original IFC element.

LDAC 2016

In 2016, the 4th LDAC workshop will be hosted by the Ontology Engineering Group (OEG) at Universidad Politécnica de Madrid (UPM) in Madrid, Spain. More information will follow.

Acknowledgements
All picture credits go to Nam Vu Hoang. A sincere thank you goes to all participants of the LDAC2015 Workshop on Linked Data in Architecture and Construction:

Ana Roxin, University of Burgundy
Chi Zhang, TU Eindhoven
Daan Oostinga, Semmtech
Hendro Wicaksono, Karlsruhe Institute of Technology
Francisco Regateiro, Instituto Superior Técnico, Universidade de Lisboa
Jakob Beetz, TU Eindhoven
Jyrki Oraskari, Aalto University

Kris McGlinn, Trinity College Dublin
Leif Granholm, Trimble/Tekla
Léon van Berlo, TNO
María Poveda-Villalón, Universidad Politécnica de Madrid
Mathias Kadolsky, TU Dresden
Matthew Horrigan, University College Dublin
Matthias Weise, AEC3
Michel Böhms, TNO
Nam Vu Hoang, Aalto University
Nick Kaklanis, CERTH-ITI
Nick Purshouse, Integrated Environmental Solutions
Odilo Schoch, ETH Zürich
Peter Bonsma, RDF Ltd.
Pieter Pauwels, Ghent University
Pim van den Helm, TNO
Robert Amor, University of Auckland
Seppo Törmä, Aalto University
Thomas Krijnen, TU/e
Tarcisio Mendes de Farias, Univ. of Burgundy
Tjeerd Dierckxsens, ProRail
Walter Terkaj, ITIA-CNR
Willie Lawton, Tyndall National Institute

Final thanks go to TU/e for hosting this event in their university in Eindhoven, The Netherlands. Special thanks goes to Jakob Beetz for organizing and hosting this workshop.