

IfcOWL: A case of transforming EXPRESS schemas into ontologies

JAKOB BEETZ, JOS VAN LEEUWEN, AND BAUKE DE VRIES

Department of Architecture, Building and Planning, Eindhoven University of Technology, Eindhoven, The Netherlands

(RECEIVED September 20, 2007; ACCEPTED July 13, 2008)

Abstract

Ontologies have been successfully applied as a semantic enabler of communication between both users and applications in fragmented, heterogeneous multinational business environments. In this paper we discuss the underlying principles, their current implementation status, and most importantly, their applicability to problems in the building information modeling domain. We introduce the development of an ontology for the building and construction sector based on the industry foundation classes. We discuss several approaches of lifting modeling information that is based on the express family of languages for data modeling onto a logically rigid and semantically enhanced ontological level encoded in the W3C Ontology Web Language. We exemplify the added value of such formal notation of building models by providing several examples where generic query and reasoning algorithms can be applied to problems that otherwise have to be manually hard-wired into applications for processing building information. Furthermore, we show how the underlying resource description framework and the set of technologies evolving around it can be tailored to the need of distributed collaborative work in the building and construction industry.

Keywords: Building Information Modeling; Knowledge Representation; Model-Driven Architectures; Ontologies

1. INTRODUCTION

Today, many excellent information communication technology (ICT) tools exist to support engineering tasks within the various domains involved in the building and construction industry. Each of these software tools has internal models of particular domains of interest. To orchestrate the collaboration of the various domains involved over the whole life cycle of a building, such as architecture, civil engineering, heating–ventilating–air conditioning, and facility management, these internal models need common information interfaces to facilitate complex business process chains.

One of the most urgent issues to be addressed by the research and industry community that aims at the establishment of common standards for the exchange of building information is the complexity that is inherent in common information exchange models including a large number of different sub-domains. The standardization efforts of product data exchange formats that have a long tradition in research have led to the creation of the industry foundation classes (IFC) led by the International BuildingSmart consortium¹ in recent

years. Many of the market-leading software vendors have stepped into the process of implementing this exchange format in their packages, enabling the exchange of information in heterogeneous project teams. However, many problems remain to be addressed to make this process efficient. Not only have various aspects related to the general nature of information modeling, storage, and processing are still to be solved, the introduction of an exchange model that is constantly evolving in size and complexity has also created some new problems. Especially small software vendors who have specialized in niche domains and markets as well as research and other public institutions face severe problems of adopting and maintaining interfaces between their internal computational models and the IFCs. Several efforts within the IFC development community, such as the introduction of the STEP Part 28 XML serializations, for instance data (ISO10303-28:2007, 2007), the initiatives for common partial model view definitions (Hietanen, 2006), commonly agreed property sets, and the advent of mechanisms for reference of external information from heterogeneous information sources (Wix et al., 2005), try to overcome some of these problems. At the same time, some of these new features impose the risk of an even further growing diversification of instance models, for example, by sanctioning the use of weakly typed information encoding.

Reprint requests to: Jakob Beetz, Department of Architecture, Building and Planning, Eindhoven University of Technology, Den Dolech 2, VRT 9.09, 5600 MB, Eindhoven, The Netherlands. E-mail: j.beetz@tue.nl

¹ Formerly known as International Alliance for Interoperability (IAI).

The Semantic Web initiative was prompted by the problems related to the heterogeneous data formats in distributed collaboration setups (even though XML was embraced as “the end of interoperability problems” in the beginning), a lacking standard of exchange policies, and the high level of manual human engineering requirements resulted from it. Although many varying views regarding the nature and use of the Semantic Web are in existence (“information annotation,” “classification,” “web of services,” “one giant database,” “information indexing,” etc.), a set of goals is common to all of them: to enhance the machine readability and interpretability of distributed and possibly incomplete information, and to standardize the way this information is exchanged among pieces of software. To achieve this, several new methods and technologies are being actively developed, existing ones adapted, and almost forgotten ones resurrected (Lassila & McGuinness, 2001; Corcho et al., 2003).

In this paper we propose methods for the adoption of these underlying methods and technologies that currently evolve in the research fields connected to the Semantic Web initiative to the domain of product data exchange.

The remainder of this paper is structured as follows: after a brief look at the current state of the art of both ontology engineering and product data management (PDM) in the second section we demonstrate how a complex EXPRESS schema can be semiautomatically lifted onto an ontological level, using the IFC schema as an example. In the fourth section we then illustrate how the resulting ontology and instantiations of its concepts and roles can be split along semantically meaningful borders to improve queries, reasoning tasks, and reuse in different contexts. As a concrete illustration of the added value of using IFCs as an Web Ontology Language (OWL) ontology, we address the common problem of partial model generation by using generic Resource Description Framework (RDF) query languages and ontology reasoning engines in the fifth section. We conclude by discussing the proposed solutions and giving an outlook to future directions of our work.

2. PDM AND ONTOLOGIES

PDM, and more specifically, building information modeling (BIM) have played a central role in research and development over the last decades. From early efforts such as GARM (Gielingh, 1988), RATAS (Björk, 1994), and COMBINE (Augenbroe, 1994) onward, the modeling of complex information entity relationship models (ERM; Chen, 1976) and definitions for their processing has been done using a range of languages and methods. These include NIAM (Nijssen & Halpin, 1989), IDEF1X (IDEF1X, 1993), and EXPRESS (Schenk & Wilson, 1994) of which the EXPRESS family of languages and modeling methods is the most popular in engineering contexts to date, and has been standardized as ISO 10303 “Industrial automation systems and integration—Product data representation and exchange” (STEP). There are several aspects, however, that limit the expressive power

of the languages incorporated in the STEP technologies as definition languages for engineering ontologies:

- *Lack of formal rigidness:* The means of modeling the IFCs—EXPRESS—is not based on a mathematically rigid theory such as used by OWL and other Description Logic (DL)-based ontology definition languages. To profit from some of the existing “intelligent” algorithms and technologies, however, the existence of a logic-based provable set of axioms and theorems is necessary.
- *Limited reuse and interoperability:* In the broader picture of a domain-independent Semantic Web, the encoding of the metamodel and its concretizations using the technological means of the STEP initiative is a serious limitation on the interoperability: outside of the few engineering domains that use EXPRESS, the popularity among developers, the use of this particular family of modeling languages and the existence of (affordable or free) tools is very limited. The incorporation of external ontology resources as well as the reuse of engineering ontologies is hence inhibited.
- *Lack of built-in distribution:* One of the enablers for the Semantic Web vision of interwoven small, modular, and reusable ontologies is the support of the defining formats to be distributed across networks. Although there do exist some mechanisms within the STEP world that grant support of multiple schemas and instances to be distributed, mapped, and merged among different resources, some severe structural limitations such as the file-based indexing and attribute scoping local to entity definitions constitute obstacles for easy distribution.

Although much of the ongoing research in the BIM area is based on the closed-world STEP ERM approaches of information modeling and exchange, knowledge representation (Sowa, 2000) methods have been identified as a key area of future research and are being applied in a number of past and ongoing projects. Among them are the Inteli-Grid (Dolenc et al., 2007) project, where ontologies are used for the management of virtual organizations in a grid infrastructure, and the SWOP project (Böhms et al., 2006), that aims at the creation of an engineering platform for semantically rich parametric object libraries. In the e-COGNOS project (Lima et al., 2005) a platform for the semantic integration of various knowledge resources was created that was grounded upon the conceptual definitions of the IFCs.

The underpinnings of ontology engineering for the creation and use of ontologies in environments of disparate information sources is a synthesis of several research fields including frame based-systems (Minsky, 1975), semantic networks (Woods, 1975), and description logics (Baader et al., 2003). Based on these, the work done by numerous researchers in the context of the Semantic Web initiative has led to the creation of the OWL, which is the successor of the joint research activities leading to DAML + OIL (McGuinness,

2002) and based on the RDF (Lassila & Swick, 1999) and a schema vocabulary (RDFS) addition (Brickley & Guha, 2004). Of the three language flavors of OWL (Lite, DL, and Full), the one relevant for the work presented here is OWL DL, which is based on the logic *SHOIN(D)* (Horrocks et al., 2003).

In many regards, the standard definition of an ontology by Gruber (1993) as a “formal specification of a shared conceptualization” can be regarded applicable for the existing IFC model designed in EXPRESS: although its original, intended use is providing a data model for the exchange of information between different applications, key definitions on an epistemological level have been agreed upon during its design that makes it a suitable basis for the construction of an ontology. It reflects the agreement of many industry stakeholders from various domains in the building and construction industry on the definition of concepts that reoccur in most building projects, which go beyond the agreement on data types and entity names into agreeing on a philosophical mindset. For example, users can agree upon the definition of a concept “building” as some decomposable spatial structure that among other things has a defined level above the sea level and a postal address.

Following the notion of “genericity” by Spyns et al. (2002), the general nature of the concepts defined in the IFCs already exhibit some basic requirements for the design of an ontology beyond a mere single purpose data model in that they are intended as a reusable, reliable (Uschold & King, 1995), shareable, portable, and interoperable formal specification of a universe of discourse (UoD), in this case, the building and construction domain. However, to profit from the advantages of a knowledge representation system based on a rigid logic, models and their populations have to be transformed into a set of axioms forming theorems about a UoD that then can be proven to hold.

3. LIFTING EXPRESS SCHEMAS ONTO AN ONTOLOGIC LEVEL

In this section we introduce a way of lifting an EXPRESS schema onto an ontological level by transforming the language constructs from the schema into a terminology box (TBox) definition using OWL. Although the main objects of investigation to test the validity of our approach were taken from several versions of the IFCs resulting in an “IfcOWL” ontology, we have successfully applied the transformation to other schemas such as the ISO 12006-3. The aim in our efforts was to find constructions that only use the constructs of the *SHIQ(D)* logic, because this fragment of first-order logic has a good balance between decidability and efficiency of existing reasoning algorithms on the one hand and expressivity on the other. Apart from the general *ALC* spectrum (attribute language with complements) only inverse roles (for inverse attributes in the schema), role hierarchies (for subproperty definitions), and qualifying number restrictions

(for n-ary relations and collection ranges) are used. Using this model schema definition, the concepts and roles (“Every ‘Building’ has-a ‘Level above the sea’”), defined in the TBox, statements about concrete occurrences (instances) can be defined in an ABox (assertion box) (“‘VillaSavoy’ is-an-instance-of ‘Building’”). This population of an ABox is automated by converting Part 21 files using a prototype we have developed.

The remainder of this section is organized along the familiar modeling constructs of EXPRESS. For each language feature that is used in the modeling of the IFCs an according transformation in OWL is given. The prototypical transformer that was implemented is based on the EXPRESS grammar and parser generator based on ANTLR (Parr & Quong, 1995) provided by the open source project OExpress (2001) started by Stephane Lardet and Joshua Lubel.

3.1. ENTITY

One of the first steps to undertake in ontology construction is to create an inventory of concepts that will be used to describe a particular UoD (dictionary). In a second step, these concepts are classified along a specialization–generalization axis using “is-a” relationships, thereby creating a taxonomy. In the concrete case of lifting the EXPRESS schema of the IFCs onto an ontological level, the mapping between class hierarchy compositions in EXPRESS and the subsumption operator in the DL underlying OWL is rather straightforward: For each ENTITY definition in the schema a corresponding owl:Class is created as a concept in the TBox of the ontology. SUBTYPE OF and SUPERTYPE OF relations are transformed into rdfs:subClassOf relations that function as a hyponym–hypernym subsumption operator in OWL: the concept “door” in the IFCs is subsumed by the concept “building element,” which in turn, is subsumed by the concept “product.” There are, however, certain aspects of the composition mechanisms available in EXPRESS that are hardly transformable into OWL: although a SUPERTYPE OF construct prepended by the ABSTRACT keyword prevents the instantiation of a particular class, no direct mechanism in OWL exists that, for example, prevents the assertion of some resource to be a mere “IfcObject.”² Although the modeling of multiple inheritance is outside the scope of the IFC definition by design, and has thus not been addressed in the scope of our current research, this construct can be reproduced in OWL using set operators such as owl:unionOf or owl:intersectionOf on a number of super classes in the subsumption definition of an owl:Class. As an example, consider this abbreviated ENTITY hierarchy from the IFCs translated into OWL:

² Although creating a value partition by making A subClassOf (unionOf [B, C]) and disjoint(B C) has a similar effect. This means that any instance of A must be either a B or a C.

```

ENTITY IfcElement
  ABSTRACT SUPERTYPE OF      (ONEOF(IfcBuildingElement,
                                   IfcFurnishingElement, [...]))
END_ENTITY;

ENTITY IfcBuildingElement
  ABSTRACT SUPERTYPE OF      (ONEOF(IfcDoor, IfcWall, IfcSlab, [...]))
  SUBTYPE OF (IfcElement) ;
END_ENTITY;

ENTITY IfcDoor
  SUBTYPE OF (IfcBuildingElement) ;
END_ENTITY;

```

which translates to OWL (given in N3 for improved readability; Berners-Lee, 1998).

```

:IfcElement
  a      owl:Class ;
  rdfs:subClassOf owl:Thing .

:IfcBuildingElement
  a      owl:Class ;
  rdfs:subClassOf :IfcElement ;
  owl:disjointWith :IfcFurnishingElement

:IfcDoor
  a      owl:Class ;
  rdfs:subClassOf :IfcBuildingElement ;
  owl:disjointWith :IfcWall , :IfcWindow

```

3.2. SELECT TYPES

A second application of OWL set operators to transform the IFCs into an OWL ontology is used for the EXPRESS SELECT types: similar to abstract super classes, they allow the typing of attributes of ENTITY classes to be a cascading (SELECTs may contain other SELECTs) and variable (but immutable) choice of simple or complex type definitions. In the case of ENTITY definitions as choices of SELECT types a simple addition of the concept definition references to the union domain range of owl:ObjectProperty (or the

of simple types as choices of SELECT types can only be addressed by treating simple types as classes.³

3.3. Types

For the transformation of simple data types such as INTEGER, REAL, STRING, and so forth, and their instances as attribute values, the simple approach of creating owl:DatatypeProperty with a range of respective XML schema data types leads to several problems: multiple anonymous identical values in (unordered) collections cannot be instantiated because they contradict the uniqueness requirement of set members. Furthermore, the SELECT construct allows the use of both simple types and ENTITY definitions. Using the transformation described earlier the latter makes the inclusion of XML Schema types in the union range of owl:ObjectProperty necessary, which is disallowed by the specification.

For these reasons we propose the creation of wrapper classes, similar to languages like Java, for the simple data types in EXPRESS and to subclass defined types from these wrappers. Each of the wrapper classes has a single owl:DatatypeProperty with a range of the according XML Schema type. This creates the additional advantage of having a consistent transformation of *all* attributes into owl:ObjectProperty. We argue that the overhead created by having to instantiate a class and a property is outweighed by the advantages of the extended usability for the issues mentioned above and the general tendency of reasoning implementations to have a much broader support for owl:ObjectProperty.

To illustrate this, let us consider the following abbreviated definition of a building in the IFCs:

```

TYPE IfcLengthMeasure = REAL;
END_TYPE;

ENTITY IfcBuilding
  SUBTYPE OF (IfcSpatialStructureElement) ;
  ElevationOfRefHeight : OPTIONAL IfcLengthMeasure;
END_ENTITY;

```

definition of separate umbrella classes) fulfills the same purpose. However, the transformation of the (rare) occurrences

³ See discussion of simple type transformations later in Section 3.3.

Using the proposed transformation we write this as OWL given in N3 (Berners-Lee, 1998) for improved readability

```

:IfcRealType
  a owl:Class .
:has_real_value
  a owl:DatatypeProperty;
  rdfs:domain :IfcRealType;
  rdfs:range <http://www.w3.org/2001/XMLSchema#double> .
:IfcLengthMeasure
  a owl:Class;
  rdfs:subClassOf :IfcRealType .
:IfcBuilding
  a owl:Class;
  rdfs:subClassOf :IfcSpatialStructureElement;
  rdfs:subClassOf
    [ a owl:Restriction ;
      owl:onProperty :ElevationOfRefHeight ;
      owl:allValuesFrom :IfcLengthMeasure
    ].
:ElevationOfRefHeight
  a owl:ObjectProperty
  rdfs:domain :IfcBuilding
  rdfs:range :IfcLengthMeasure .

```

This short example also illustrates the mechanism to disambiguate attributes of the same name with different types as it occurs in models like the IFC. In EXPRESS, the scope of attributes is local; in RDF/OWL, each property is global. To be able to have the same global attribute name with different types, these types are added to the range of the property. In the concrete use of such a property, for example, the `IfcBuilding` example above, this range is then restricted to the type valid in this context.

3.4. Attributes

For the transformation of EXPRESS attributes into corresponding roles that connect conceptual classes in DL (called properties in OWL) an important difference in modeling approaches is the scoping of both in their languages: although EXPRESS attribute names are local to their ENTITIES, the OWL properties are global as a part of the open world assumption. To avoid potential naming conflicts in large schemas and ontologies such as the IFCs several approaches are possible: attributes with the same name but different types could be transformed into `owl:ObjectProperty`s with both types included in their ranges and both classes in their possible domains. Using role restrictions such as the universal quantification `owl:someValuesFrom` the proper use in the context of the attribute can then be limited to its proper meaning. To avoid typing conflicts when trying to filter both simple typed attributes and entity references, we propose the simple type class wrapper approach described earlier as a solution.

A second solution to scoping issues we have successfully applied in our transformer is the reduction of possible naming

conflicts by compartmenting the model along semantic borders into separate XML namespaces and/or physically differ-

ent resources. For a broader discussion about this approach see Section 4 in this paper.

3.5. Enumerations

To limit the range of properties to some concrete individuals or simple data type values, OWL offers the `owl:oneOf` construction. For the many cases where the nature of some specific concept is further specified by an enumeration or concrete value in the IFC model or other STEP-based models, like setting the general construction type of a roof to one of “FLAT_ROOF,” “SHED_ROOF,” “GABLE_ROOF,” and so forth, via the “`IfcRoofTypeEnum`,” the range of the corresponding attribute is set to one of the string values that represents it. In the XML/RDF schema encoding of OWL models, these string values are members of an `owl:oneOf`,⁴ using `rdf:first` and `rdf:rest` constructs to order the values. The reason for this is that no other in-build RDF collection type has the possibility to finalize a collection such that a completely defined set is created that ensures the restriction of the possible ranges of values.

3.6. Collections and n-ary relations

One of the important requirements to ontology modeling in engineering domains is the ability to represent (ordered)

⁴ The `owl:oneOf` construct is an `rdf:Property` assigned to an anonymous `owl:DataRange` with an `rdf:List` as its range. Although the use of `rdf:List` is generally disallowed in OWL DL, this special construct for enumerated `owl:-Datatype` values is an exception and does not make it OWL Full (Bechhofer et al., 2004).

collections of concept instances such as wall material layers, sequences of tasks or simple things like coordinates. Because of the nature of DL-based ontologies, however, concepts and their instances are treated as sets and their members, and little or no in-built mechanisms are offered for ordered structures. Possible solutions to this issue that we have successfully applied to the ifcOWL case include the construction of ordered collection-like structures on an upper ontology level using OWL-DL constructs and the application of lower level RDF constructs without stepping outside of the DL complexity box into OWL-Full.

Inspired by the work of Drummond et al. (2006), constructing ordered collections using only the DL language profile of OWL can be achieved by creating an owl:Class with two object properties to hold instance references to the actual value and the next list class member. By applying a universal quantification that restricts the value members to a given concept a typed list can be created that ensures the construction of valid lists. A clear drawback of this approach is that at present the support of generic reasoning implementation that supports the recursive traversal of such list for the collection of all members is very limited. Parts of this problem can be resolved, however, by adding a third transitive role on list entries that allows inference engines, which support transitive reasoning to check the existence of members without explicit recursion. Extending the suggestions made by Drummond et al. (2006), this OWL DL list mechanism can be enabled to guard the size of some collection types (lists and sets⁵) using cardinality restrictions. In addition to role memberships in OWL DL lists, the slot fillers representing the actual values, in addition, are directly assigned to an owl:ObjectProperty that is then restricted with owl:maxCardinality, owl:minCardinality, or owl:cardinality in the owl:Classes of its domain.

A second option that does not allow reasoning⁶ but enables more flexible access by RDF toolkits and query engines is the augmentation of set members by RDF lists. Here the standard encoding of RDF lists is used to keep the order for members of a set in a collection. A workaround to prevent the “pollution” of an OWL-DL ontology with RDF constructs is the separation of contents and ordering elements. Here, the idea is to keep RDF-List elements in a separate namespace or physical file that is not visible to generic non-RDF aware reasoning engines but can be loaded on demand by lower level processors such as query engines. In the experiments we have conducted we successfully applied a file-naming convention by appending .list extensions to the regular contents that holds the ordering information.

⁵ The enforcement of uniqueness of collection elements using Drummond et al.’s (2006) OWL DL lists (as modeled with the UNIQUE keyword in EXPRESS) as well as BAGs, which allow duplicate values in unordered collections by reasoners, has not been addressed in our current work. Both problems are nontrivial, and cannot be addressed within OWL DL to the best of our knowledge.

⁶ Because it involves language constructs from the lower RDF levels that are not part of the strict OWL vocabulary.

4. PARTITIONING

One of the most urgent obstacles to overcome to come to a wide use of vendor-independent, interoperable building information models is the reduction of the complexity and size of the current IFCs. Not only has the extent of its current versions (2x3(g)/2x4) reached a level of complexity that requires the devotion of a considerable amount of time before any interfacing of custom of the shelf (COTS) solutions can be done, but also the prospective future extension for domains like bridges and roads and the growing number of standardized and nonstandardized extensions using the property set mechanism will further increase the demand for partial model handling in future. This issue has been recognized and addressed by a number of researchers and practitioners: Adachi (2003) has proposed a Partial Model Query Language, Weise et al. (2003) have laid out a Generalized Model Subset Definition Schema. Wix et al. (2005) are working on the compilation of an Information Delivery Manual (IDM) for the exchange of subsets of the overall model, Lee and Sacks (2006) have applied Georgia Tech Process to Product Modeling (GTPPM) to this problem for the composition of views, and in several nonengineering fields work has been done aiming at the segmentation of very large ontologies (Seidenberg & Rector, 2006). The approach of both IDM and GTPPM is the replication of schema parts that are then used as independent. In the case of IDM, the reintegration into the original schema and hence round tripping is achieved more easily because the replication includes the types of the original schema, whereas the GTPPM approach semantically “flattens” out all the information by shifting attributes from super classes to the child classes and (currently) converting all types to simple string types.

The approach we propose to address parts of this problem is the compartmentalization of both TBox and ABox along configurable semantic borders given in a mapping file or evaluated per individual use as a result of graph extraction queries that can be assembled by the assistance of a tool we have developed for this purpose.

The compartmentalization of the large IFC model into smaller, thematically grouped subparts is achieved by using the inherent distribution facilities of the lower levels of OWL, namely, XML and RDF: because every resource that is part of an RDF triple may have different locations the sets of classes and properties used to compose the model as outlined in the earlier sections may be distributed among arbitrary resources. As a proof of concept we have applied the architectural structure of the model defined by the IAI: several layers, domain models and basic resources such as geometric resources and material resources, have been encapsulated into different namespaces. A schematic overview of the resulting model is illustrated in Figure 1. Referencing other parts of the models can be achieved on two different levels of semantic expressiveness: on a low level nodes and edges of external RDF graphs may be referenced from an ontology. As long as the processing that needs to be done stays on the RDF lan-

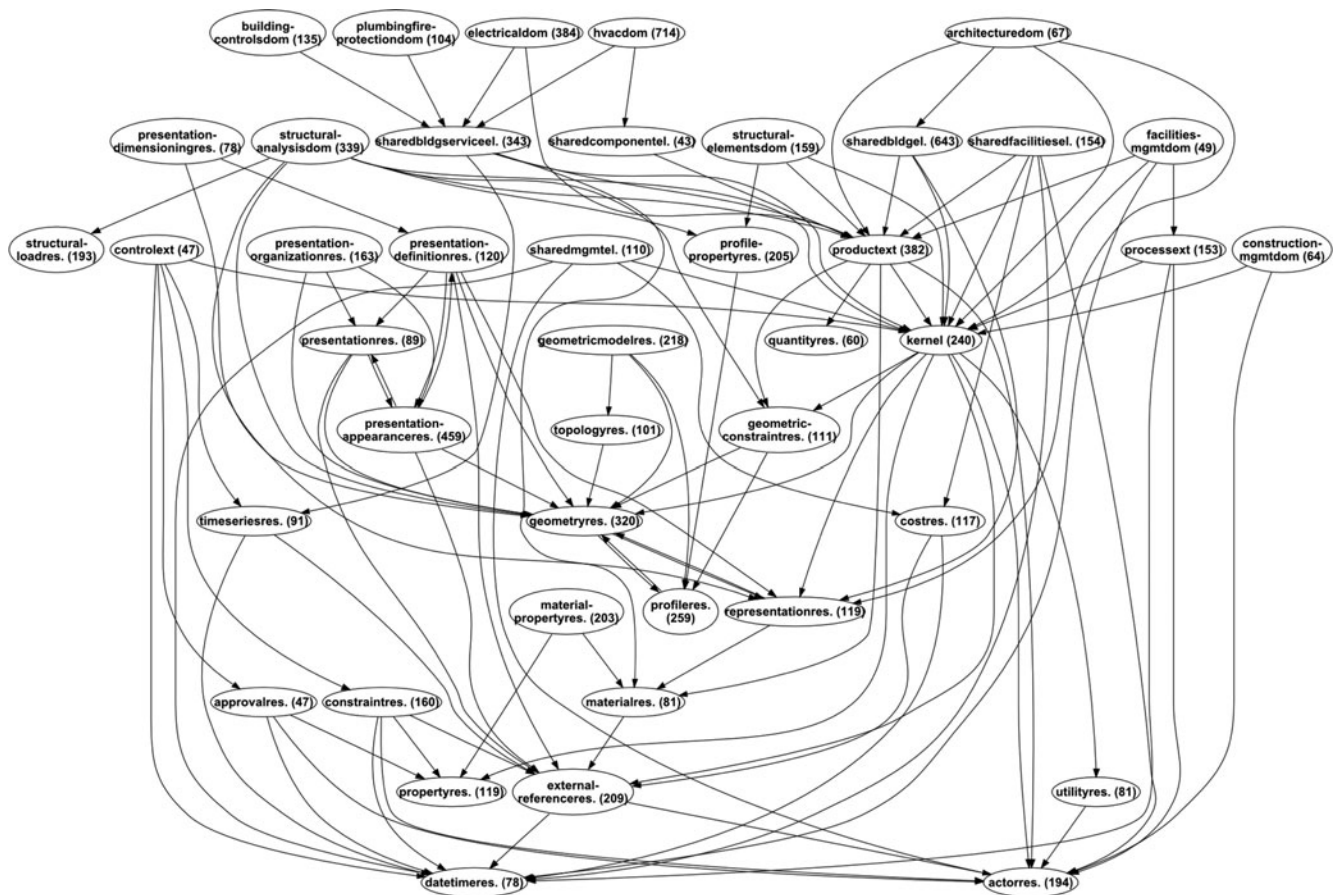


Fig. 1. The resulting ontologies of IfcOWL, which are configured here for the logical compartments according to the original IAI architecture. The IfcMeasureResource namespace is left out for layout reasons.

guage level, “unresolved” references to nodes may be treated as anonymous resources while still being valid RDF. On the ontological level, however, anonymous resources may not be used in statements that compose a DL-complex OWL ontology. This means that at least the particular class, property or restriction has to be resolved, that is, pulled from the external ontology. The problem with the current language specification of OWL is, however, that the mechanism designed to interweave several ontologies only allows the inclusion of complete ontologies via the *owl:imports* statement. To make a truly interwoven semantic net possible, however, mechanisms and standards will have to be developed that allow the retrieval of only parts of certain ontologies without having to work on the complete set of nodes and edges.

In the current design of the compartments of the IFC model, the many, partly cyclic, references through ENTITY attributes that point to external resources (see Fig. 2) prevent the easy extraction of completely autonomous subparts. Although in most cases the references to external resources only include a few or even a single edge to an external node, the effects are dramatic. For example, the Entity *IfcProductType* defined in the *IfcKernel* subschema has an attribute *RepresentationMaps* that stores references to one or more *IfcRepresentationMap* entities that are defined in the *IfcGeo-*

metryResource subschema. Because most interdependencies between the subschemas that enable geometric and topological representations are cyclic in nature, that is, cannot be separated in an easy manner, the single property edge *RepresentationMaps* results in the inclusion of a few hundred additional definitions. A clear advantage of encoding the model using RDF triplets is the global scope of statements over the local scope definitions of attributes within entities. Here, it is possible to move triplets (or nodes and edges of the graph) to an arbitrary external resource. For the case of IfcOWL we eliminated a number of interdependencies moving edges like *RepresentationMaps* into separate partial ontologies, we refer to as *pivot ontologies*: the aim here is to reduce the out-degree of nodes representing sets of concepts, for example, the namespace *IfcKernel*, which is referenced by many namespaces and whose use results in long chains of transitive ontology inclusions (see Fig. 2). By creating a separate namespace that attributes *IfcProductType* with the *RepresentationMaps* owl:ObjectProperty (and its according restrictions) we create independent semantic clusters that can be used in cases where, for example, representational geometry is not of interest (see Fig. 3).

Even greater impact on the capability and performance of reasoning and query tasks than the compartmentalization of

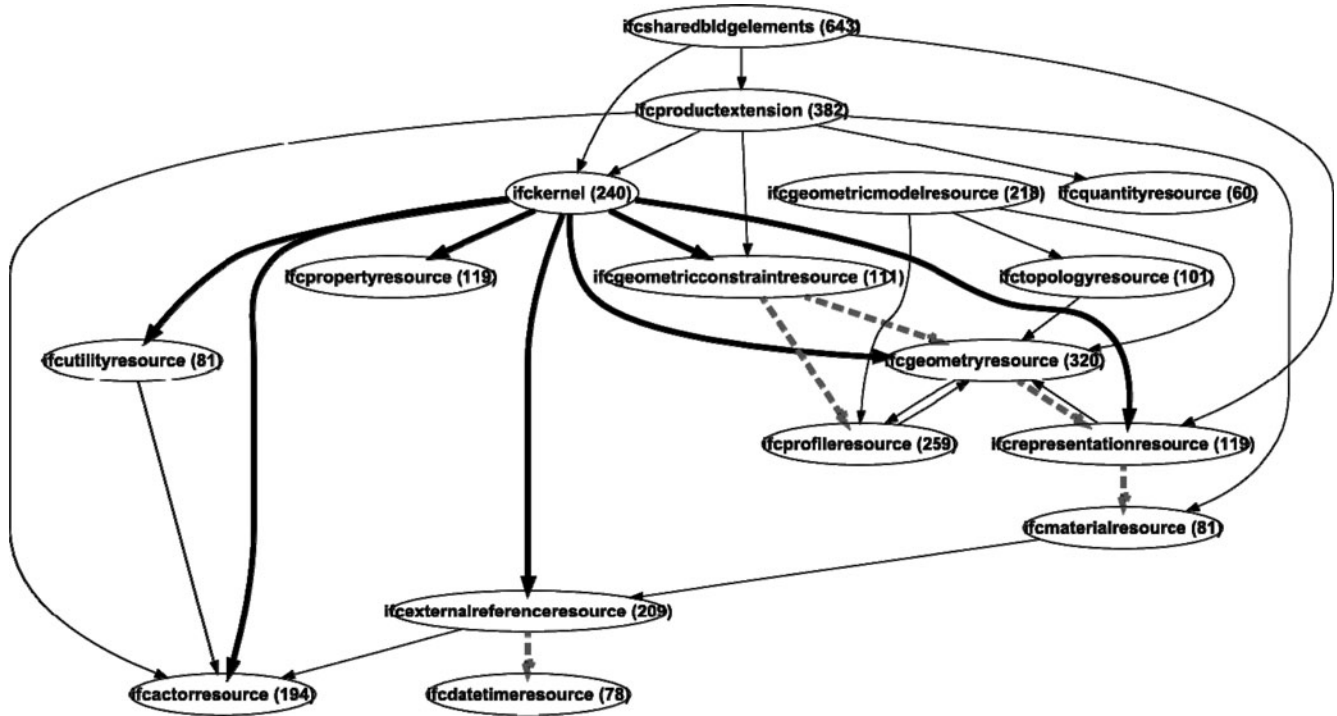


Fig. 2. The Ifc platform subpart. Direct import references of the basic *ifckernel* namespace are in bold, and indirect transitive imports are indicated by dashed lines. Numbers denote the amount of classes and properties.

the TBoxes with its several hundreds or thousands of statements has the compartmentalization of ABoxes, where real-world models easily have millions of statements declaring instances of concepts and roles. We have successfully applied various compartmentalization strategies for IFC instance files converting part 21 physical files into meaningful subparts. The strategies can be roughly categorized as the following combinations of TBox and ABox compartmentalizations:

- *oAoT—one ABox one TBox*: Both definitions of concepts and their assertions as facts reside in a single compartment: concepts from the complete schema (platform and nonplatform, all layers) reside in one namespace. Where necessary, disambiguations of property names (e.g., *Height* typed as *IfcInteger* for *IfcPixelTexture* vs. *Height* typed as *IfcPositiveLengthMeasure[Real]* for *IfcRectangularPyramid*) are done via universal restrictions $\forall R.C$ on ranges.
- *oAnT—one ABox n TBoxes*: Definitions of concepts reside in different namespaces and resources. Chances of necessary disambiguations are relatively low since *ifcgeometricmodelresource:Height* (for pyramid) and *ifcrepresentationappearanceresource:Height* (for texture) have different localized scopes.
- *nAoT—n ABoxes one TBox*: Splitting up the asserted instances of concepts into several compartments has many advantages: irrelevant information can be left out of consideration, that is, reasoning tasks can be limited to required information. For the IFC model several

ABox-partition approaches are configurable in our prototypical transformer resulting in different views of the overall model.

- *nAnT—n ABoxes n TBoxes*: The solution with the greatest degree of freedom, but also most demanding from a managing perspective, is the partitioning into both split TBoxes and split ABoxes.

In the following we give example configurations of the *nAnT* approach to illustrate its use in practical scenarios.

According to our analysis of various reference and example populations such as the HITOS model (Lê et al., 2006), around 80% of all entity instantiations come from the namespaces *ifcgeometryresource* and *ifctopologyresource*. One of the evident choices of semantic borders for ABox partitions are hence the decoupling of these 80% from the rest of the model. The remaining type of information is more applicable for the use in a contemporary logic-based reasoning environment, such as the various properties attached to specific building elements or more general components like material layers, and so forth. To further increase this decoupling and thus increase the ease of use and the performance, we tested configurations where geometric parts were not only outsourced to one node repository per project, but split along the *Representations* property edges that connect one or more geometric/topologic *IfcRepresentations* to an *IfcProduct* such as a wall, window or door. For prospective future reasoning and query engines specialized on spatial processing tasks such as started by Borrmann et al. (2007), further partitions can

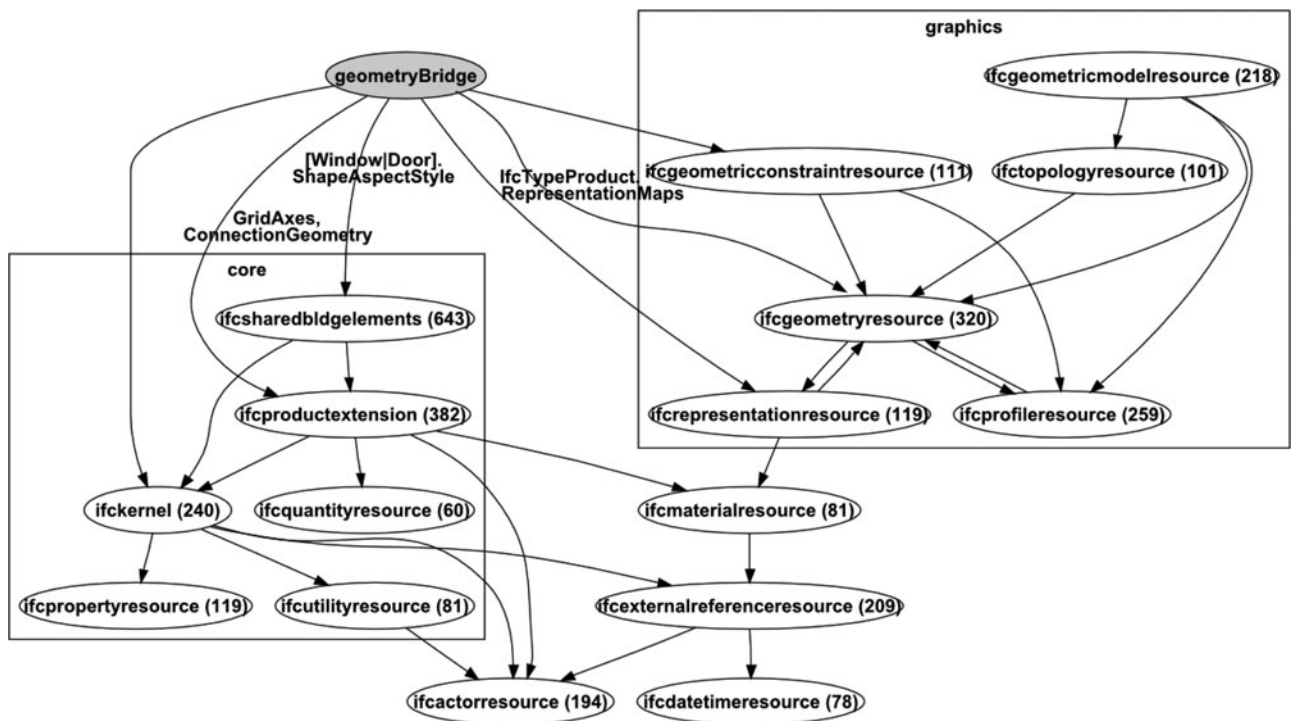


Fig. 3. The clustering of definitions through externalization of critical properties into separate pivot ontologies.

be made by separating, for example, the three common representations of plan view, three-dimensional model, and bounding box into different repositories.

5. A PRACTICAL EXAMPLE: PARTIAL MODEL EXTRACTION USING RDF QUERIES

One of the practical advantages OWL/RDF modeling has over traditional STEP/EXPRESS methods becomes apparent when using it for the creation of functional part (FP) definitions. FPs, defined in the IDM methodology, are the data artifacts used in the exchange of information between two (software) actors. Instead of using error-prone replications of the overall schema to reduce the implementation work necessary for a software vendor, references to the full model(s) are used. By pointing to the corresponding (distributed) schemas, an OWL/RDF-aware application “knows” the semantic definition of a window and its properties by pulling just the defining triplets from the schema resources when their availability becomes necessary. Moreover, the actual expansion of the complete definition of the *IfcWindow* class is not necessary for simple operations such as partial graph extraction, because the equality of the resource uniform resource identifiers (URIs; which function as unique identifiers) is a sufficient comparator for a processor to, for example, extract a “thermal windows” view from a large model. To look for and extract the relevant subgraph we have to formulate a graph pattern to search in the original model (Beetz et al., 2007).

An example of an application interested in a minimal amount of information is a decision support system that needs

all thermal transmittance U-values of all windows from a building project.

Using our IfcOWL ontology configuration we can define the subgraph of the required information as a digraph

$$G = (\{n_{win}, n_{rel}, n_{prop}, n_{val}\}, \{(n_{win}, n_{rel}), (n_{rel}, n_{prop}), (n_{prop}, n_{val})\})$$

with

$$n_{win} = \text{IfcWindow}, \quad n_{rel} = \text{IfcRelDefinesByProperties}, \\ n_{prop} = \text{IfcPropertySet}, \quad n_{val} = \text{IfcPropertySingleValue}$$

The graph consists of an “IfcWindow” node that is connected to an objectified property assignment node (*IfcRelDefinesByProperties*), which in turn, connects the window to a PSet (*IfcPropertySet*) having a some value (*IfcPropertySingleValue*). Using this simplified graph as a basic target pattern, we can generate a small graph pattern matching query in one of the languages such as SPARQL (Arenas et al., 2006), SeRQL (Broekstra & Kampman, 2003), RQL (Karavelakis et al., 2002) for which some fast and efficient free and open source software implementations exist, and pull a fraction from a large model (which itself can be distributed over various locations). Using graph query languages to operate on large ontologies like average IFC models is less complex than the use of complete rule and reasoning engines (whose use we are going to illustrate for the semantic validation of our submodels in later in this section). In cases where standardized query operations do not suffice, many imple-

mentations, such as ARQ (2005), allow the creation of domain specific extensions. One of such useful extensions could be, for example, the implementation of spatial operators such as currently worked on by Borrmann et al. (2007).

For the concrete example we make use of the SPARQL “CONSTRUCT” feature, which allows the creation of new graphs. We extract a subgraph containing all windows that have “ThermalTransmittance” properties attached via IfcPropertySets of an original (distributed) model. Notice that the actual graph matching pattern (in the WHERE part of the query) has some additional nodes and edges compared to the simplified digraph above.⁷

```

CONSTRUCT {
  ?window a ifcsharedbldgelements:IfcWindow .
  ?window ifckernel:IsDefinedBy ?defines .
  ?defines a ifckernel:IfcRelDefinesByProperties .
  ?defines ifckernel:RelatingPropertyDefinition ?propDef .
  ?propDef a ifckernel:IfcPropertySet .
  ?propDef ifckernel:HasProperties ?singleProp .
  ?singleProp a ifcpropertyresource:IfcPropertySingleValue .
  ?singleProp ifcpropertyresource:Name ?name .
  ?singleProp ifcpropertyresource:NominalValue ?value .
}
WHERE {
  ?window a ifcsharedbldgelements:IfcWindow .
  ?window ifckernel:IsDefinedBy ?defines .
  ?defines a ifckernel:IfcRelDefinesByProperties .
  ?defines ifckernel:RelatingPropertyDefinition ?propDef .
  ?propDef ifckernel:HasProperties ?singleProp .
  ?singleProp ifcpropertyresource:Name "ThermalTransmittance".
  ?singleProp ifcpropertyresource:NominalValue ?value .
}

```

This can be read as “Construct a graph using IfcOWL namespaces where each window has a related property set one of whose properties has a thermal transmittance value.”

This results in a partial graph depicted in Figure 4 that only contains the minimal information needed by the target application. At the same time, the graph carries provenance infor-

mation by pointing to the corresponding schema elements and occurrences. The advantage over other partial view generation approaches that cut, transform, or even flatten the model is that the references to the original items are kept intact. To keep the view consistent with the global model, the target application could reevaluate the slot filler values by resolving the URIs. In contrast such consistency maintenance is only possible for certain nodes and relations (i.e., those who have a globally unique ID) in conventional IFC populations. However, for this to work additional version management over time has to be done (van Leeuwen & Fridqvist, 2003). Several approaches for temporal logic and provenance data

in RDF for the purpose of journaling and model consistency are introduced (Gutierrez et al., 2005; Huang & Stuckenschmidt, 2005).

To ensure that all the windows extracted by the query comply with the requirements for windows of the target application, or in IDM terms form a valid FP, the following DL definition is made:

$$\begin{aligned}
 \text{fp_thermalWindow} \equiv & \\
 & \text{IfcWindow} \\
 \cap & \exists \text{IsDefinedBy.} (\\
 & \quad \exists \text{RelatingPropertyDefinition.} (\\
 & \quad \quad \exists \text{HasProperties.} (\\
 & \quad \quad \quad (\exists \text{Name "ThermalTransmittance"}) \\
 & \quad \quad \quad \cap (\geq 1 \text{NominalValue}) \\
 & \quad \quad \quad \cap (\forall \text{Unit. IFCTHERMALTRANSMSR}) \\
 & \quad \quad) \\
 & \quad) \\
 &)
 \end{aligned}$$

⁷ Note that it is implied that inferred symmetric properties owl:inverseOf have been asserted into the graph beforehand. isDefinedBy in this case has to be explicitly added finding the symmetric closure on the RelatedObjects property that has the domain IfcRelDefinesByProperties class.

This construct can be read as “A ThermalWindow is a window that has some related property definition which has some property by the name of ‘ThermalTransmittance’ and at least

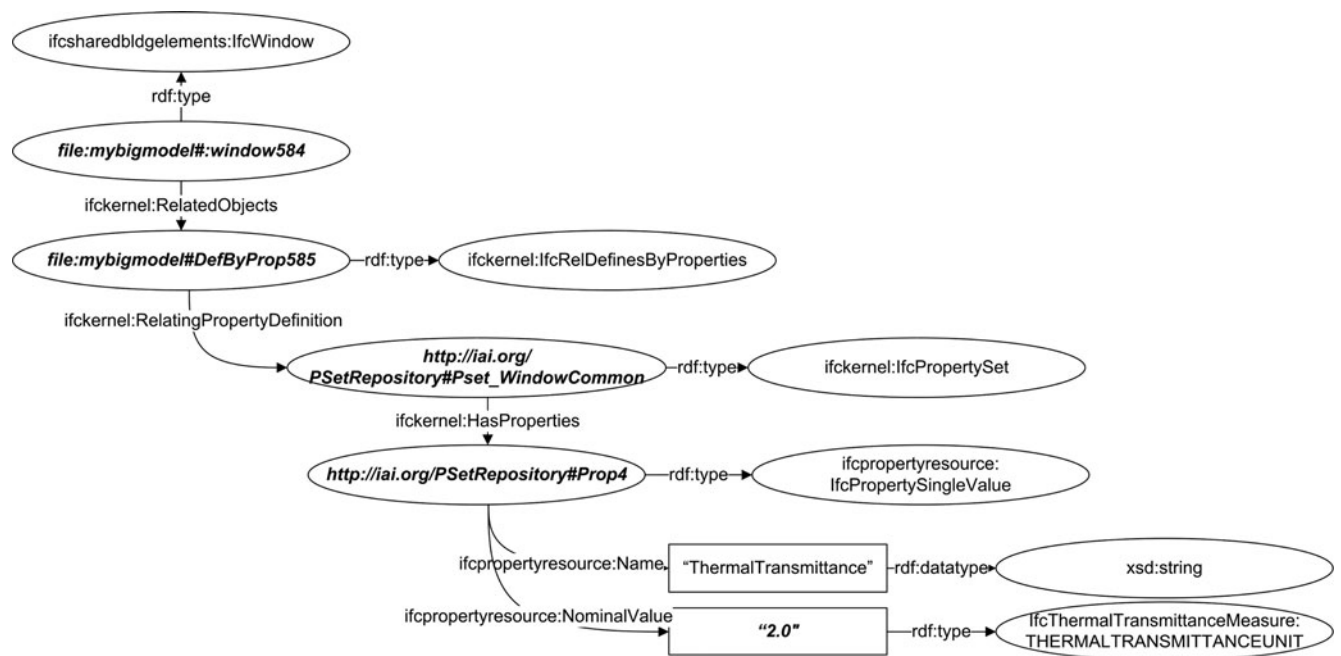


Fig. 4. The minimal subgraph extracted by a query.

one value ‘NominalValue’ all of whose Unit types are ‘ThermalTransmittanceMeasures.’”

To enable a reasoning engine to successfully find entailments of these nested axioms and assert them into the graph, the definitions of all participating concepts have to be known. This adds an additional level of complexity on top of the simple queries in the earlier extraction. Although the latter is handled on the pure RDF graph layer, the semantic meaning of classes, their attributes, and relations have to be known during this validation stage. It might be considered a drawback that the Open World Assumption, which is the basis of all reasoning on OWL, does not allow us to extract all windows that do *not* have a “ThermalTransmittance” value directly. In the Semantic Web world all information is considered incomplete (there might be some value for “ThermalTransmittance” that is not accessible in the current context); hence, an answer to the negation of the second part of the above axiom returns “unknown” rather than “false.” However, in a scenario where we would like to be able to detect these (e.g., in order to prompt the user to fill in the necessary missing values), we could simply iterate over all the windows and mark those who have not been classified as `fp_ThermalWindow` for further processing (falling back to the level of conventional imperative programming).

6. CONCLUSIONS

In this paper we presented a way of semiautomatically lifting legacy EXPRESS schemas onto an ontological level to profit from methods and algorithms in the emerging field of interoperability based on knowledge representation. We have described how we applied this technique to the large standard

product data model for the building and construction industry, the IFCs. We have suggested how ontology compartmentalization can be used and configured to reduce the complexity of implementing tools for the processing of such information. We have exemplified the practical use of such ontologies by addressing the common problem of view definitions and functional part extractions for collaboration scenarios.

7. DISCUSSION AND OUTLOOK

The approach to transforming EXPRESS schemas into ontologies that we have presented in this paper is only one of many possible starting points for the future uses of ontologies in the building and construction field. It might be argued that a clean start from scratch will lead to much more carefully designed engineering ontologies. We argue, however, that the careful augmentation of an established model instead of its replacement is of special importance in the conservative and slow-moving building and construction industry. Vice versa, for some of the semantic information that is present in the IFC schemas, such as WHERE rule constraints, and procedural FUNCTION calls a fully automatic conversion will probably never be reached. However, the wealth of methods, algorithms, and readily available implementations that is actively researched and developed in the fields of knowledge representation and ontology-driven architectures can be a valuable asset for both model designers and end-user software implementers. Despite the convincing argumentation of some researchers that full semantic interoperability can never be reached (Edmonds & Bryson, 2004) we see the introduction of more formal methods to the field of interoperability as a valuable improvement.

The lack of support for some modeling requirements that are specific to engineering domains such as convenient ordering mechanisms for collections and the need for greater support of numerical reasoning will eventually be addressed by the research community once the practical use has reached a critical mass. An example of such improvements based on feedbacks by practitioners is the addition of qualified numerical restrictions in OWL 1.1.

Our own future research will be focused on the use of the ontologies presented here as a grounding for semantically enhanced Web services.

REFERENCES

- Adachi, Y. (2003). Overview of partial model query language. *Proc. ISPE Conf. Concurrent Engineering 2003*, pp. 549–555.
- Arenas, M., Perez, J., & Gutierrez, C. (2006). Semantics and complexity of SPARQL. *Proc. 5th Int. Semantic Web Conf. (ISWC06)*, pp. 30–43.
- ARQ (2005). ARQ—a SPARQL processor for Jena. Accessed at <http://jena.sourceforge.net/ARQ/> on August 2007.
- Augenbroe, G. (1994). An overview of the COMBINE project. *Proc. 1st European Conf. Product and Process Modeling in the Building Industry ECPPM'94*, p. 547.
- Baader, F., Calvanese, D., McGuinness, D., Nardi, D., & Patel-Schneider, P. (2003). *The Description Logic Handbook: Theory, Implementation, and Applications*. New York: Cambridge University Press.
- Bechhofer, S., van Harmelen, F., Hendler, J., et al. (2004). OWL Web Ontology Language Reference. *W3C Recommendation*. Accessed at <http://www.w3.org/TR/owl-ref/> on October 7, 2007.
- Beetz, J., de Vries, B., & van Leeuwen, J.P. (2007). RDF-based distributed functional part specifications for the facilitation of service-based architectures. *Proc. 24th W78 Conf. Maribor 2007*, pp. 183–188.
- Berners-Lee, T. (1998). *Notation 3 (N3): A Readable RDF Syntax*. Technical Report World Wide Web Consortium, 1998–2006. Accessed at <http://www.w3.org/DesignIssues/Notation3.html> on August 2007.
- Björk, B. (1994). RATAS Project—developing an infrastructure for computer-integrated construction. *Journal of Computing in Civil Engineering* 8(4), 401–419.
- Böhms, M., Bonsma, P., Bourdeau, M., Pascual, E., Storer, G., & Sedano, J. (2006). “Semantic Web”-based product modelling: ontologies and configurators. In *Exploiting the Knowledge Economy: Issues, Applications and Case Studies* (Cunningham, P., & Cunningham, M., Eds.), pp. 1099–1107. Amsterdam: IOS Press.
- Borrmann, A., Schraufstetter, S., van Treeck, C., & Rank, E. (2007). An octree-based implementation of directional operators in a 3D spatial query language for building information models. *Proc. 24th W78 Conf. Maribor*.
- Brickley, D., & Guha, R. (2004, February 10). RDF Vocabulary description language 1.0: RDF schema W3C recommendation. Accessed at <http://www.w3.org/TR/2004/REC-rdf-schema-20040210/>
- Broekstra, J., & Kampman, A. (2003). SeRQL: a second generation RDF query language. *Proc. SWAD—Europe Workshop on Semantic Web Storage and Retrieval*.
- Chen, P.P. (1976). The entity-relationship model—toward a unified view of data. *ACM Transactions on Database Systems* 1(1), 9–36.
- Corcho, O., Fernández-López, M., & Gómez-Pérez, A. (2003). Methodologies, tools and languages for building ontologies. Where is their meeting point? *Data & Knowledge Engineering* 46(1), 41–64.
- Dolenc, M., Katranuschkov, P., Gehre, A., Kurowski, K., & Turk, Z. (2007). The InteliGrid platform for virtual organisations interoperability. *ITcon 12*, 459–477.
- Drummond, N., Rector, A., Stevens, R., Moulton, G., Horridge, M., Wang, H., & Seidenberg, J. (2006). Putting OWL in order: patterns for sequences in OWL. *Proc. 2nd OWL Experiences and Directions Workshop*, Athens.
- Edmonds, B., & Bryson, J.J. (2004). The insufficiency of formal design methods—the necessity of an experimental approach for the understanding and control of complex MAS. *Proc. Third Int. Joint Conf. Autonomous Agents and Multiagent Systems*, pp. 938–945.
- Gielingh, W. (1988). General AEC reference model (GARM) an aid for the integration of application specific product definition models. *Proc. CIB W74 + W78 Workshop*, pp. 165–178.
- Gruber, T. (1993). A translation approach to portable ontology specifications. *Knowledge Acquisition* 5(2), 199–220.
- Gutierrez, C., Hurtado, C., & Vaisman, A. (2005). Temporal RDF. *Proc. 2nd European Semantic Web Conf. (ESWC '05)*, pp. 93–107.
- Hietanen, J. (2006). *IFC Model View Definition Format*, Technical Report. International Alliance for Interoperability.
- Horrocks, I., Patel-Schneider, P.F., & van Harmelen, F. (2003). From SHIQ and RDF to OWL: the making of a Web Ontology Language. *Web Semantics: Science, Services and Agents on the World Wide Web* 1(1), 7–26.
- Huang, Z., & Stuckenschmidt, H. (2005). Reasoning with multi-version ontologies: a temporal logic approach. *Int. Semantic Web Conf.*, pp. 398–412.
- IDEF1x. (1993). *Integration Definition for Information Modeling (IDEF1x)*. FIPS Publication 184, National Institute of Standards and Technology.
- ISO. (2007). *ISO10303-28:2007. Industrial Automation Systems and Integration—Product Data Representation and Exchange. Part 28: Implementation Methods: XML Representations of EXPRESS Schemas and Data, Using XML Schemas*. Geneva: ISO.
- Karvounarakis, G., Alexaki, S., Christophides, V., Plexousakis, D., & Scholl, M. (2002). RQL: a declarative query language for RDF. *Proc. 11th Int. Conf. World Wide Web*, pp. 592–603. New York: ACM Press.
- Lassila, O., & McGuinness, D. (2001). The role of frame-based representation on the Semantic Web. *Linköping Electronic Articles in Computer and Information Science* 6.
- Lê, M., Mohus, F., Kvarsvik, O., & Lie, M. (2006). The HITOS project—a full scale IFC test. *Proc. ECPPM 2006*.
- Lee, E.C., & Sacks, R. (2006). Generating IFC VIEWS and conformance classes using GTPPM. *Proc. 11th Int. Conf. Computing in Civil and Building Engineering ICCCBE-XI*, pp. 1715–1724.
- Lima, C., El-Diraby, T., & Stephens, J. (2005). Ontology-based optimisation of knowledge management in e-Construction. *ITcon 10*, 305–327.
- McGuinness, D., Fikes, R., Hendler, J., & Stein, L. (2002). DAML+ OIL: an ontology language for the Semantic Web. *Intelligent Systems* 17(5), 72–80.
- Minsky, M. (1975). A framework for representing knowledge. In *The Psychology of Computer Vision* (Winston, P., Ed.), pp. 211–277. New York: McGraw-Hill.
- Nijssen, G., & Halpin, T. (1989). *Conceptual Schema and Relational Database Design: A Fact Oriented Approach*. Upper Saddle River, NJ: Prentice-Hall.
- OSExpress. (2001). Project homepage. Accessed at <http://osexpress.sourceforge.net/> on August 2007.
- Parr, T., & Quong, R. (1995). ANTLR: a predicated-LL(k) parser generator. *Software—Practice and Experience* 25(7), 789–810.
- Schenck, D., & Wilson, P. (1994). *Information Modeling: The EXPRESS Way*. New York: Oxford University Press.
- Seidenberg, J., & Rector, A. (2006). Web ontology segmentation: analysis, classification and use. *WWW '06: Proc. 15th Int. Conf. World Wide Web*, pp. 13–22. New York: ACM Press.
- Sowa, J. (2000). *Knowledge Representation: Logical, Philosophical, and Computational Foundations*. Pacific Grove, CA: MIT Press.
- Spyns, P., Meersman, R., & Jarrar, M. (2002). Data modeling versus ontology engineering. *ACM SIGMOD Record* 31, 12–17.
- Ushold, M., & King, M. (1995). Towards a methodology for building ontologies. *Proc. Workshop on Basic Ontological Issues in Knowledge Sharing in Conjunction With IJCAI95*, Artificial Intelligence Applications Institute, University of Edinburgh.
- van Leeuwen, J.P., & Fridqvist, S. (2003). Object version control for collaborative design—characteristics of the concept-modelling framework. *E-Activities in Building Design and Construction—9th EuroPIA Int. Conf.* (Tunçer, B., Özariyildiz, S.S., & Sariyildiz, S., Eds.), pp. 129–139. Paris: EuroPIA Productions.
- Weise, M., Katranuschkov, P., & Scherer, R.J. (2003). Generalised model subset definition schema. *Proc. CIB-W78 Conf. 2003—Information Technology for Construction*. Accessed at <https://www.cs.auckland.ac.nz/w78/papers/>
- Wix, J., Christensen, L., Mohus, F., Stangeland, B., & Espedokken, K. (2005). A framework of information for construction—creating an industry-wide enterprise architecture for building information modeling

use. *Proc. 6th European Conf. Product and Process Modelling*, pp. 665–671, Valencia, Spain, September 13–15.

Woods, W.A. (1975). What's in a link: foundations for semantic networks. In *Representation and Understanding: Studies in Cognitive Science* (Bobrow, D., & Collins, A., Eds.), pp. 35–82. New York: Academic Press.

Jakob Beetz is an Assistant Professor in the Design Systems Group in the Department of Architecture, Building and Planning at Eindhoven University of Technology. He will receive his PhD in 2009. He is actively involved in teaching activities and several national and international research initiatives. Mr. Beetz is the author of several publications and has been a peer reviewer for international conferences and journals. His research interests are focused on building information modeling, Semantic Web technologies, and agent systems.

Jos van Leeuwen has been an Associate Professor at the University of Madeira, Portugal, since 2006, where he teaches in the areas of computational design and interaction design in several undergraduate and graduate engineering programs. He previously taught at Eindhoven University of Technology.

He is the founder and Director of the undergraduate Interactive Media Design Program and Assistant Director of the Laboratory for Usage-Centered Software Engineering (www.labuse.org). Dr. van Leeuwen is also an adjunct faculty of Carnegie Mellon University, teaching in the joint master's program of human–computer interaction at Madeira. His current research interests are in interaction design, social networks, and innovating communication.

Bauke de Vries is Chair of the Design Systems Group of the Department of Architecture, Building and Planning at Eindhoven University of Technology. He is the author of publications that have been presented at international conferences and scientific articles that have been published in international journals. Dr. de Vries has been a member of many PhD committees and project leader in European research projects. His main research topics are computer-aided architectural design, product and process modeling, virtual reality technology and interfaces, and knowledge-based systems.