Beetz, J., van Leeuwen, J.P. and de Vries, B. **2005**. "An Ontology Web Language Notation of the Industry Foundation Classes." In Scherer, R.J., Katranuschkov, P., Schapke and S.-E. (eds.): *Proceedings of the 22nd CIB W78 Conference on Information Technology in Construction*. CIB publication no. 304. Dresden: Technical University of Dresden. 193-198.

An Ontology Web Language Notation of the Industry Foundation Classes

J. Beetz, J.P. van Leeuwen, B. de Vries
Eindhoven University of Technology

ABSTRACT: In this paper we describe and discuss an OWL notation of IFCs, its advantages over generic XML schema representation, its various fields of possible application, and our implementation of it in a multi-agent framework.

1INTRODUCTION

1.1*Problem domain*

Formats for integrated building models for the description of buildings covering their complete lifecycle have been a major topic in the ICT/AEC research community (Amor et al. 1995, Eastman 1995). The actively ongoing effort of the International Alliance for Interoperability (IAI) to bring together various software vendors and research institutions has led to a standard that is increasingly accepted by the AEC/FM industry: The Industry Foundation Classes (IFC). While the main means of data modeling for this format have been the ISO certified STEP and EXPRESS (ISO 10303) languages, recent developments introduced eXtensible Markup Language (XML) technologies into the process. Data modeling and processing with XML has been embraced by other industries and a number of initiatives have led to the adoption of XML as core technology for the exchange of business relevant data. Although the AEC/FM industry is traditionally fragmented and slow in embracing new technologies, a number of initiatives (ifcXML, aecXML, BLIS-XML, bcXML and others) have aimed to extend, integrate or complement IFC with XML. However useful and promising these developments are, the use of XML does not constitute a virtue in itself.

Although the biggest advantage of XML is its standardized, well-formed and plain-text nature which enables developers to read and understand it and to work with it in a vast collection of (freely available) tools, it requires a significant amount of engineering, manual work and coordination to enable interoperability. The reason for this is that XML does not solve the problem of semantic interrelationships of data models.

To further automate development processes and enable high-level processing of data, the Semantic Web (SW) initiative has opened the door to a new era of 'intelligent' data exchange "in which information is given well-defined meaning, better enabling computers and people to work in cooperation" (Berners-Lee et al. 2001).

In order to create this machine-readable meaning, the underlying expert knowledge has to be encoded in high-level classifications (ontologies) in standardized ways. The W3C has recommended to use the Ontology Web Language (OWL) (Bechhofer et al. 2004) that is based on the Resource Description Framework (RDF). OWL allows suitable tools to reason on data and to draw conclusions from the statements made about a specific subject, such as a building.

One of the most promising applications of this technology is the creation of web services (McIlraith et al. 2000) and agent based systems (Weiss 2000) to introduce new means of distributed collaboration for the AEC/FM industry. In such a scenario, expert domain knowledge such as building standards, product databases or even structural calculations could be encoded in the form of expert agents or web services that are automatically discovered and consulted. The paper describes how the OWL representation of IFCs forms the basis for the creation of such agent systems that represent building elements and services as well as building expertise. The introduction of formalized rule sets (i.e. using the Semantic Web

Rule Language (SWRL)) and agent based services is within the scope of our future research.
In this paper we present and discuss the work that was done on the derivation of an OWL ontology from the EXPRESS Schema of the Industry Foundation Classes version 2x2.

2MOTIVATION

Several researchers within the AEC/FM community have identified multi agent based systems as a promising technology to assist (distributed) collaboration, decision making and design. Several promising prototypes have been implemented and demonstrated the potential of this technology in various fields (Meißner et al. 2004).
The classic architectures of software agents rely on the ability of any single agent to reason about the outside world based on an internal representation that it keeps of the outside world. Furthermore, in multi agent systems agents must be able to communicate with each other about some aspects of their environment, beliefs or plans. This makes it necessary to rely on common concepts about the outside world and the ability to communicate about them in shared language.
An ontology is “a formal specification of a shared conceptualization” (Gruber 1993, Borst et al. 1997). Data models – even if they are as feature-rich as the ones described in EXPRESS and UML - should not be confused to be the same as axiomatic theories about “the things that are” (ontologies). However, creating a hierarchical, relational model to store data as has been done in the IFCs constructs such an ontology as a side-effect.
Rather than crafting special purpose ontologies and language codecs for multi agent system, the large body of knowledge that has been built up by the IFC community over the years could be used.
Outside of agent systems, the notation of AEC/FM content in an XML-based ontology description format like OWL can be used for various applications within the Semantic Web context:

- Information discovery and retrieval. Distributed information such as building related product catalogues, maintenance information for FM purposes etc. can be indexed and searched including their semantic relations.
- Semantic Web services. Complex services, such as building physics simulations could be wrapped into web services describe how to interface and interpret themselves.
- Distributed data storage: Unlike EXPRESS based models, information in XML can be stored in different physical locations and later be linked together.
- Mapping into other description formats: with several existing standards and data models in existence, mapping in an n-n manner is much more demanding than mapping to a pivot ontology. An IFC based ontology of AEC/FM content with its large set of nodes could be a candidate for such an pivot ontology.

Furthermore, an OWL-based description of the IFCs has some advantages over EXPRESS-based and XML Schema based methods:.

- User base: Compared to STEP EXPRESS based technologies with their small user base and its niche market character, XML is a widespread technology with a vast set of existing tools that could significantly ease the development of new tools by smaller companies and research institutions.
- Predefined relations, restricted constructs and expressiveness: Although there is often more than one solution to model a certain concept in OWL in different ways, the set of possibilities is more limited than on the lower levels of the language stack (XML, XMLS, RDF/RDFS). This helps to establish a commonly used “best practice” model and hence helps avoiding fragmentation of standards.

3RELATED WORK AND RESEARCH

Creating an XML notation of EXPRESS part 28/21 has been on the roadmap of the ISO TC184/SC4 for quite some time now, and several implementations are available. However, one of the weaknesses of an XML schema approach, is that some of the expressiveness of the EXPRESS Schema definition language is lost. Another approach lies in the translation of STEP part 21 into an XML notation of the Unified Modeling Language. Although much more straightforward and strict, some language constructs just cannot be translated.
An open source implementation of an EXPRESS-OWL transformer has been created for the Oil and Gas production facilities ISO 15926 in the context of the OMPEK project by (Batres et al. 2005) Based on the open source “osexpress” parser, a basic transformation into OWL using the Jena API has been created. Although some language constructs are still missing, a basic ENTITY/class hierarchy along with some properties of the IFC schema can be created

with it. To date this is the only Open Source tool available.
Lima et al. (2004) have created an EXPRESS-to-OWL-transformer as part of the FUNSIEC project. Again, the output of the transformer is limited to the purposes of the project (mapping between different ontologies).
The free set of tools that have been created by the exff.org team around David Price and others approach the translation effort by cascading XSLT transformations via UML/XMI (Price & Bodington 2004)
In the context of the e-COGNOS (Lima et al. 2003) project, an ontology based on DAML+OIL (the predecessor of OWL) was created describing building and construction related concepts in a multilingual way.

4UNDERLYING TECHNOLOGY

4.1*STEP EXPRESS*

The Standard for the Exchange of Product Model Data (STEP) that is described in the different parts of the ISO 10303 has been the key technology in the exchange of data within many large industries for a very long time. The definition of objects, their relations to other objects and their constraints are defined in the EXPRESS (ISO 10303 part 11) language. This very powerful means of data modeling was developed in the early 1980s predating UML and XML. It was aimed at being a flexible, extendible and scaleable modeling language easy to be read by human experts. However successful in many industries, only a very limited amount of developers is familiar with it. Simple examples of EXPRESS schemas include classes with primitive types like

```
ENTITY door;
SUBTYPE OF (buildingPart)
   height: REAL;
WHERE
   WR : height > 0;
END_ENTITY;
```

Describing a concept door that in addition to everything inherited from an existing concept “buildingPart” has a property height of the primitive type ‘REAL’ that is constraint to values greater than zero.

4.2*RDF/RDFS*

The Resource Description Framework (RDF) and its schema definition extension (RDFS) are XML-based description and modeling formats that have gained increasing popularity in recent times. It is based on simple predicate-subject-object triplets connecting any two resources (subject and object). By making statements about statements, very complex descriptions can be built up. The resources themselves are denoted by uniform resource identifiers (URI). Since URLs are a special form URIs, data models and instance data based on them can be distributed over different physical storage locations. This is a clear advantage over STEP where no distribution is supported. Since RDF is not designed to be understood by human readers, a schema language (RDFS) was added that introduces predefined constructs, such as classes, properties and constraints, making modeling of complex concepts easier.

4.3*Description Logics*

Description Logics (DLs) are a set of formal languages to represent knowledge in a certain domain based on atomic concepts and roles. The roots of DLs lie in first order predicate logic. They are the basis of ontology description languages such as OWL.

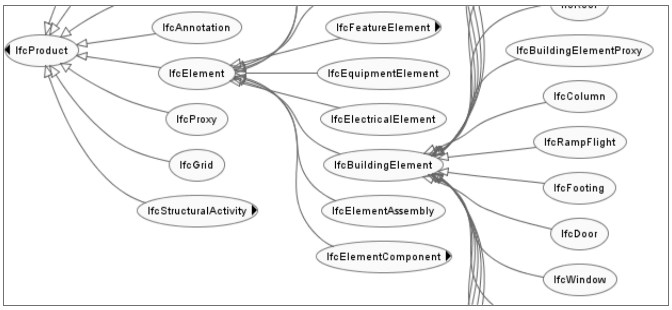

Figure 1. Extract from a simple class hierarchy of the IFCs in OWL.

4.4*OWL*

Based on and extending RDFS the W3C recommendation for the description of ontologies, the Ontology Web Language (OWL) has received a lot of attention over the recent years. Historically it has evolved from the description logic language SHIQ (Horrocks et al 2003). The current grammar and syntax has evolved from the DAML + OIL (McGuinness et al 2003), the result of the joint efforts of American (DARPA Agent Markup Language – DAML) and European (Ontology Integration Language – OIL) projects. It comes in three different flavors of complexity and expressiveness:

- OWL lite, with some basic extensions to RDFS introducing property restrictions, universal and existential quantifiers
- OWL DL, adding enumerated classes, boolean combinations of classes and restrictions, the concept of disjointness and full cardinality.

– OWL full, sharing the same vocabulary as DL but removing some limitations from DL.

With the amount of freedom and expressiveness that each layer adds, the decidability for reasoning engines and the compatibility to other Semantic Web applications is limited down. For a discussion on decidability see (Heflin 2004).

OWL as well as RDF(S) models can be written in two different syntaxes, with the XML-based version intended for actual web use and the N3 notation for easier consumption of human readers.

The small example given in an EXPRESS model above could be expressed in OWL as follows:

```
<owl:Class rdf:ID="buildingPart"/>
  <owl:Class rdf:ID="door">
    <rdfs:subClassOf rdf:resource="#buildingPart"/>
    <rdfs:subClassOf>
      <owl:Restriction>
        <owl:cardinality
rdf:datatype="http://www.w3.org/2001/XMLSchema#int"
        >1</owl:cardinality>
        <owl:onProperty>
          <owl:DatatypeProperty rdf:ID="height"/>
        </owl:onProperty>
      </owl:Restriction>
    </rdfs:subClassOf>
  </owl:Class>
  <owl:DatatypeProperty rdf:about="#height">
    <rdfs:range
rdf:resource="http://www.w3.org/2001/XMLSchema#float"/>
    <rdfs:domain rdf:resource="#door"/>
  </owl:DatatypeProperty>
```

In the above example, each door is said to have exactly one property height. But what about the constraint that is given in the original EXPRESS schema, that every door height must be greater than zero? For some datatypes, these kinds of restrictions can be expressed by setting the datatype to the according XML schema type, such as xs::nonNegativeInteger. In this case, no such restricting type exists for floating point values in the standard. While this easy problem might even be solved on the XML schema level, there are a number of more complex rules used as constraints in the IFCs that require an additional level of definition. For this purpose, the Semantic Web architecture introduces an additional layer: rule languages.

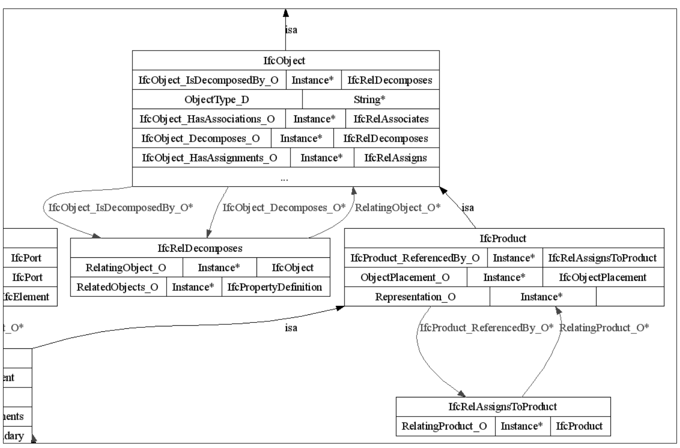

Figure2 Excerpt from a graph visualizing slots of the OWL model of the IFCs

4.5*RuleML/SWRL*

The Semantic Web Rule Language (SWRL) combines OWL with a subset of the Rule Markup Language developed by the RuleML initiative. It has been submitted for comments to the W3C and will eventually become a full recommendation. With SWRL, complex rules can be defined in a standardardized way, enabling reasoners to check constraints or infer 'new' knowledge from a given ontology[1]. The flexibility and expressiveness of SWRL will allow most of the EXPRESS constraint to be modeled. The fact that even some basic numerical computation can be described for a reasoner to solve is promising for this purpose. For the above case a formal notation might look like this:

```
door(?x) ^ hasHeight(?x,?height) ^
swrlb:greaterThan(?height, 0)
-> WR1(?x,true)
```

EXPRESS construct	OWL construct
ENTITY	Class
SUB/SUPERTYPE	subClassOf
SELECT	Class and subClassOf
INVERSE	inverseOf / InverseFunctionalProperty
ENUM	DatatypeProperty […] owl:DataRange […] owl:one of […] rfd:List or enumerated classes.
Cardinality constraints	owl:cardinality, owl:minCardinality, owl:maxCardinality
Simple Types	Simple XML Schema types
WHERE domain rules	Possibly through SWRL rules in future
collections: LIST, SET, BAG, ARRAY	Only unordered (?)

5TRANSFORMATION

Two different approaches to derive an OWL notation of the Industry Foundation Classes have been taken in our research:

5.1*OWL from XML schema via XSLT*

In this approach we created an XSLT file that can be used to transform the Part 28 XML schema (XSD) of the IFCs 2x2 into an OWL file. In the XSD solution, some basic elements and attributes that are common for all concepts are defined externally on a lower level.

One of the obstacles to overcome to achieve a transformation is the difference in element order and nesting between XML Schema and OWL: XML

[1] The classic example of such an inference is the uncleOf property: If an individual x has a father y who has a brother z then the individual z is the uncle of x.

files that are compliant to a certain schema must strictly adhere to the element order and nesting modeled in the schema. In RDF files and consequently in OWL, the order of elements and their position within a single file or even their storage location can be chosen freely (i.e. a property of a class can be defined in another file on another server. On the one hand this has some advantages when creating derived knowledge representations; on the other hand, this introduces many consistency problems. A clear advantage over proprietary parser/transformers (as in the second approach) is the ability to use standard tools for the transformation, as the set of XSLT features that was used was kept compatible to most of the popular engines. Furthermore, the same strategy can be used when handling instance data.
Although the complete ENTITY/class hierarchy found in the XML schema notation of the IFC can be transformed into OWL along with all of the constraints, the information being lost while deriving the XML notation was reason enough to attempt a second approach, trying to capture as much of the underlying knowledge encoded in the original EXPRESS schema as possible.

5.2 *OWL from EXPRESS schema*

In this second approach the OWL notation is derived directly from the original EXPRESS schema format of the IFCs. This made the creation of a proprietary parser necessary, which was build on top of the java/ANTLR-based lexer provided by the open source osexpress (Luebell 2001 and Parr & Quong 1995) project. With this approach, we are able to make use of the full range of EXPRESS constructs. To date, we have implemented most of the basic types, relations (including the reverse relations that are not captured in the XML schema), enums and cardinality constraints. This results in an ontology with over 850 classes and more than 4000 overall frames[2].
The most important advantage of this approach over the intermediate XML schema translation is, that with it, it is possible to maintain all the additional knowledge that is captured in, e.g., the over 300 WHERE domain rules in the core model and that is not transferable into XML schema.

6 CONCLUSIONS AND FUTURE WORK

In this paper we have presented and discussed two approaches to derive an ontology in the OWL notation from the Industry Foundation Classes modeled in EXPRESS schema. We have outlined the potential of such ontology in the context of multi agent systems and the broader context of various future Semantic Web applications. We have shown, that much of the knowledge that has been created by the effort of the IFC community can be captured and used with the large set of (free and open) tools that are under active development by the large community around the Semantic Web. Furthermore, we have shown how upcoming standards such as SWRL are potentially well suited to capture even complex constraints, which is not possible with lower level modeling standards such as XML schema alone.
Ongoing and future research is aimed at using this ontology both as internal representation of domain expert agents in collaborative design scenarios and as a language codec for the inter-agent communication in multi-agent systems. In this context we will also have a look at how this ontology might be used as a pivot ontology for the mapping between various other models such as the ISO 12006-3.

[2] The ontology has been validated by the WonderWeb and the Pellet validators

REFERENCES

Amor, R., Augenbroe, G., Hosking, J.G., Rombouts, W., and Grundy, J.C. Directions in modelling environments. *Automation in Construction* , 4 (1995), pp. 173-187.

Batres, R., West, M., Leal, D. & Nakaa, Y. 2005 An Upper Ontology based on ISO 15926. In L. Puigjaner and A. Espuña (eds.) *Proceedings of the 38th European Symposium on Computer Aided Process Engineering.*

Bechhofer, S. van Harmlelen, F., Hendler, J., Horroks, I., L.McGuiness, D., Patel-Schneider, P., Stein, L. 2004. OWL Web Ontology Language Reference.*W3C Recommendation 10 February 2004”* W3C 10.02.2004 10.01.2005 *http://www.w3.org/TR/owl-ref/*

Berners-Lee, T., Hendler, J. & Lassila, O.. 2001. The Semantic Web.. *Scientific American*, May 2001, pp. 35-43.

Borst, P., Akkermans, H. & Top, J. 1997. Engineering ontologies. *International Journal of Human-Computer Studies*, 46:365-406.

Eastman, C.M. 1992. Modeling of buildings: Evolution and concepts. *Automation in Construction. Vol 1(2): pp. 99-121*

Gruber, T. 1993. A translation approach to portable ontology specifications. *Knowledge Acquistion* 5(2): 199-220

Heflin, J 2004. Web ontology language (owl) use cases and requirements. W3C Recommendation 10 February 2004. http://www.w3.org/TR/2004/REC-webont-req-20040210/

Horrocks, I., Patel-Schneider, P., and van Harmelen, F., 2003 From SHIQ and RDF to OWL: The Making of a Web Ontology Language, *Journal of Web Semantics* 1(1):7--26, 2003

Liebich, T. 2005. 15.12.05. http://www.iai-international.org/iai_international/ Technical_Documents/files/ifcXML_for_IFC2x2.xsd>

Lima, C., Fies, B., El-Diraby, T. A., & Lefrancois, G. 2003. The challenge of using a domain Ontology in KM solutions: the e-COGNOS experience". *In Proceedings of 10th ISPE International Conference on Concurrent Engineering: Research and Applications 2003.*

Lima, C; Fès, B.; Ferreira da Silva, C. 2004 Setting up the open semantic infrastructure for the European construction sector

the FUNSIEC project. In: *Proceedings of the ECPPM 2004.* Rotterdam: Balkema.
Lubell, S. 2001. Open Source EXPRESS Parser Documentation. http://osexpress.sourceforge.net/exparser.html. 04.03.2005
McGuinness, D.L. Fikes, R. Hendler, J. Stein, L.A. , 2003. DAML+OIL: an ontology language for the Semantic Web, *IEEE Intelligent Systems,* 17(5): 72-80
McIlraith, S., Son, T. & Zeng, H. 2000. Semantic web services. *IEEE Intelligent Systems,* 16(2):pp.46–53
Meißner, Udo F.; Rüppel, Uwe; Theiß, Mirko, 2004 : Network-Based Fire Engineering Supported by Agents In: *Proceedings of the Xth International Conference on Computing in Civil and Building Engineering (ICCCBE-X), Weimar , Juni 2004*
Parr,T.J. & Quong, W. 1995. ANTLR: A Predicated-LL(k) Parser Generator. *Software Practice and Experience*, 25(7): 789-810
Price, D. & Bodington, R. 2004 Applying Semantic Web Technology to the Life Cycle Support of Complex Engineering Assets, *Lecture Notes in Computer Science*, 3298: 812 - 822
Weiss, G. (ed.) 2000. Multiagent Systems A Modern Approach to Distributed Artificial Intelligence. Cambridge, MA, USA: MIT Press