Appendix C

Virtual Physical Link (VPL)

Ontology to Support Virtual Models Linked with Physical Components

Virtual Physical Link (VPL)

Ontology to Support Virtual Models Linked with Physical Components

Kristian Birch Sørensen

Ramboll Denmark

Abstract: *This appendix is a supplement to Paper I and Paper II and demonstrates how an ontology to support virtual models linked with physical components in construction can be developed. Ontology development methods, tools and recommendations are described and demonstrated by initialising the development of the ontology: "Virtual Physical Link".*

It is also the author's expectation that this appendix will inspire researchers and developers in the development of future ontologies for the construction industry (e.g. classification systems) to have focus on modern information and communication technology (ICT) usage rather than looking backwards for solutions to support outdated information handling methods.

1 Introduction

Development and use of commonly accepted ontoloiges, "*... explicit specification of a conceptualization*" (Gruber, 1993), are important to enable reuse of information in construction, share common understanding of the structure of information among people or ICT systems, make assumptions explicit, and analyse knowledge (Noy & McGuinness, 2001).

In Paper I and Paper II the author identified a need for new ontology development to enable a link between virtual models and physical components in construction. However, it is beyond the resources of this research project to develop a full-fledged ontology for this comprehensive purpose. Therefore, this appendix will present the early first steps, methods and ideas towards the development of such an ontology named: "Virtual Physical Link" (VPL).

Various terminologies, languages and tools can be used for ontology development and implementation. There is not a single best approach because it depends on the intended use of the ontology and competences of the developers. When the ontology is implemented as a hierarchical classification system, it is common to use documents or spreadsheets with tables of relevant terms. The main advantage of this approach is that experienced practitioners can read the tables immediately and contribute to the development. Some of the disadvantages are that the spreadsheet and document based authoring tools often lack automation in use and in evaluation of the consistency, correctness and redundancy of the ontology.

Support and use of Industry Foundation Classes (IFC) is critical, in connection with virtual building models ontology development. IFC is described in the EXPRESS language for product models and graphical tools exist (such as EDMvisualExpress) that can assist the ontology developer in the ontology development and validation. This is a major advantage compared to the spreadsheet and document based method. However, one of the main challenges with this approach is the transformation from EXPRESS to other programming languages, integrated development environments (IDE), ICT systems and relational databases.

Another possibility is to implement the ontology as Web Services described in the XML-based Web Services Description Language (WSDL) (Christensen et al., 2001). Graphical user-interfaces and parsers for XML and WSDL assisting the developer in quick code generation are widely implemented in modern IDEs such as Netbeans and Eclipse. From a business perspective this approach can be efficient, but a challenge is not to loose the full context and only focus on short-termed problems.

Ontologies are in relation to knowledge engineering and Semantic Web development often implemented in the languages RDF Schema (RDF-S) or the more expressive Web Ontology Language (OWL). Several graphical authoring tools exist that can assist the developer of Semantic Web ontologies. Protégé is one of them and was also chosen for the ontology development described in this appendix, see Horridge (2009) for a practical user guide. It supports visualisation, automatic taxonomy classification, consistency check, Java class generation, and database connection etc. A critical issue in relation to use of Semantic Web in the construction industry is the integration of IFC based virtual models. Various researchers have focused on transforming IFC into OWL such as El-Diraby et al. (2005) and Beetz et al. (2009). In another research project, a set of ontologies (SemanticSTEP) and derived Express schemas, that define the mapping between ISO 10303 (STEP) models and ontologies in OWL/RDF were developed (S-TEN, 2008). Converters were also developed in relation to the project that might be useful for a future IFC to OWL conversion. When developing generic applicable Semantic Web ontologies for the construction industry, a challenge is not being specific enough to support any business case.

For the ontology development described in this appendix it was chosen to use the OWL approach. The aim is to initialise an ontology development that can be used to define a basis for new ICT applications in construction supporting a digital link between virtual models and physical components. The appendix can also be used for inspiration to an ICT supported method for developing future classification systems for the construction industry.

2 Methodology

No single best methodology exists for ontology development, because there is no "correct" way to model a domain (Gasevic et al., 2006). Inspiration for the development process of ontologies can be found in software engineering and knowledge engineering. Based on literature studies and best practice in ontology development for the medical science domain Noy & McGuinness (2001) propose a methodology for ontology creation. This methodology was adapted for the work presented in this appendix and comprises the steps of:

1. Determine the domain and scope of the ontology

2. Consider reusing existing ontologies
3. Enumerate important terms in the ontology
4. Define the classes and the class hierarchy
5. Define the properties of classes – slots
6. Define the facets of the slots
7. Test by creating instances

The application of this methodology is described in the following sections.

3 Ontology Development

3.1 Domain and Scope of the Ontology

In Paper I and Paper II it was argued and illustrated by future working scenarios that there is a need for ontologies, and particularly business process ontologies, to support project progress management, work instruction delivery, quality inspection, inventory management, construction planning, procurement and facility management. Therefore, these fields of work will define the domain of the VPL ontology under development.

3.1.1 Competency Questions

One way to establish the scope of an ontology is to sketch a list of competency questions an ICT system based on the ontology should be able to answer (Grünninger and Fox, 1995). In relation to the construction industry, and in particular to the establishment of a digital link between the virtual models and physical components, the following are possible competency questions:

- Which working tasks are related to component XX and performed by person YY?
- What is the best choice of maintenance of component XX?
- How many bolts are needed to perform the installation of steel frame ZZ?
- Which one of our partner companies make the fewest errors in the installation of windows?
- How long time does the erecting of interior walls take in average and what is the standard deviation?
- Do we have enough materials in stock for next weeks planned activities?
- Is project KK running according to the schedule and which activities are not?

In order to answer these questions, classification of construction components and how they are related to working processes, persons and best practices is crucial for the ontology.

3.2 Consider Reusing Existing Ontologies

VPL is not a brand new ontology but mostly a combination of several other ontologies to fit a new domain. As a result of the evaluations, experiences and conclusions described in Paper I it is chosen to base VPL on the meta-ontology consisting of the upper classes: Business Process, Technical Service, Resource and Organisation. In addition, VPL draws on aspects from the ontology specifications of IFC (IAI, 2006), Dublin Core (DCMI, 2008), FOAF (Friend of a Friend) Vocabulary (FOAF, 2007), Physical Markup Language (PML) (Floerkemeier et al., 2003), OmniClass Table 32 (OCCS, 2006) and OWL-S, Semantic Markup for Web Services (Martin et al., 2004). To give a brief explanation these ontologies can be used to:

IFC: Resource ontology defining virtual building model related information.
Dublin Core: Resource ontology for document meta-data.
FOAF: Organisation ontology for social web-sites.
PML: Technical service ontology for automatic identification equipment and other sensors.
OmniClass Table 32: Business process ontology for services in construction.
OWL-S: Upper ontology for technical services and business processes.

3.3 Enumerate Important Terms in the Ontology

The VPL ontology is not developed from scratch, so the important terms are already enumerated in the existing ontologies described above. However, some translation or mapping between the terms in the ontologies will be needed. A Person e.g. in IFC as well as in FOAF inherits from Actor but that is the only parallel of the Actor definition in the two ontologies. In IFC the Actor is a class that enables selection between Person, Organisation or Person and Organisation. In FOAF the Actor is a super class from which the sub classes inherit common attributes such as age, e-mail address and IDs etc. Several similar challenges must be solved when integrating the six ontologies described in Section 3.2.

3.4 Define the Classes and the Class Hierarchy

In Figure 1 and Figure 2 the illustrations show how parts of the six ontologies described in Section 3.2 are used to define classes and class hierarchy within each of the four domains of the upper ontology in VPL. The figures show screen dumps from the class browser in Protégé with different classes expanded and collapsed in each view. For each entity in IFC a class is defined in VPL. The FOAF Dublin Core, and OWL-S ontologies are already described in RDF/OWL and are therefore used directly in VPL. PML is described in the six XML schemas:

- PmlCore.xsd
- Identifier.xsd
- RFIDReaderAndTags.xml
- RFIDReaderAndTagsWithMemory.xml
- RFIDReaderAndTagsWithSensor.xml
- SensorAndData.xml

For each element in the PML-XML schemas a class is defined in VPL. OmniClass Table 32 is a text based document defining names and hierarchy of working processes in construction. For each item in OmniClass Table 32 a class is defined in VPL. In addition a WorkPackage class is defined to contain the information of a business process. It is expected that for the execution works a more detailed specification will be required than the one currently available in OmniClass Table 32.

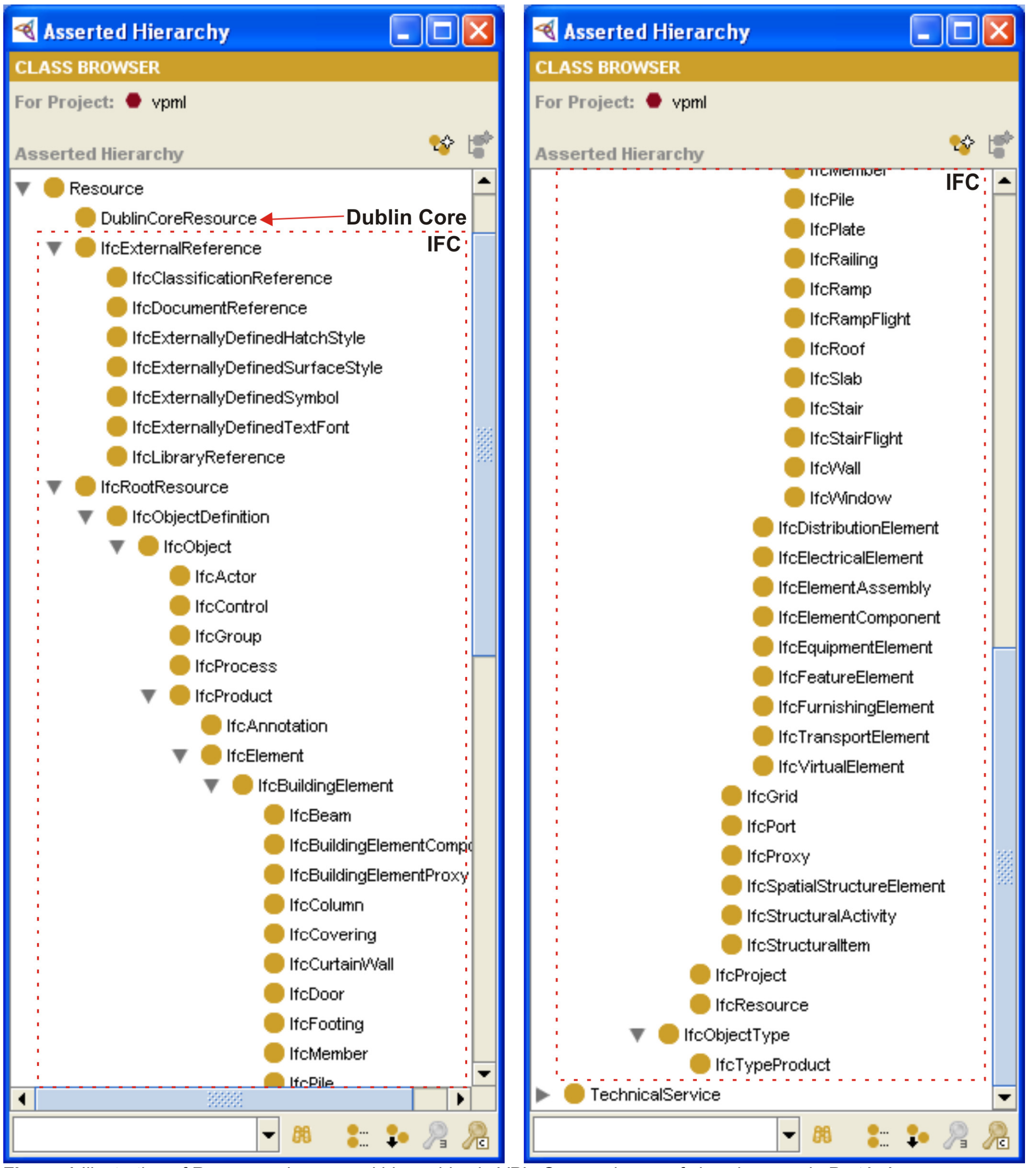

Figure 1 Illustration of Resource classes and hierarchies in VPL. Screen dumps of class browser in Protégé.

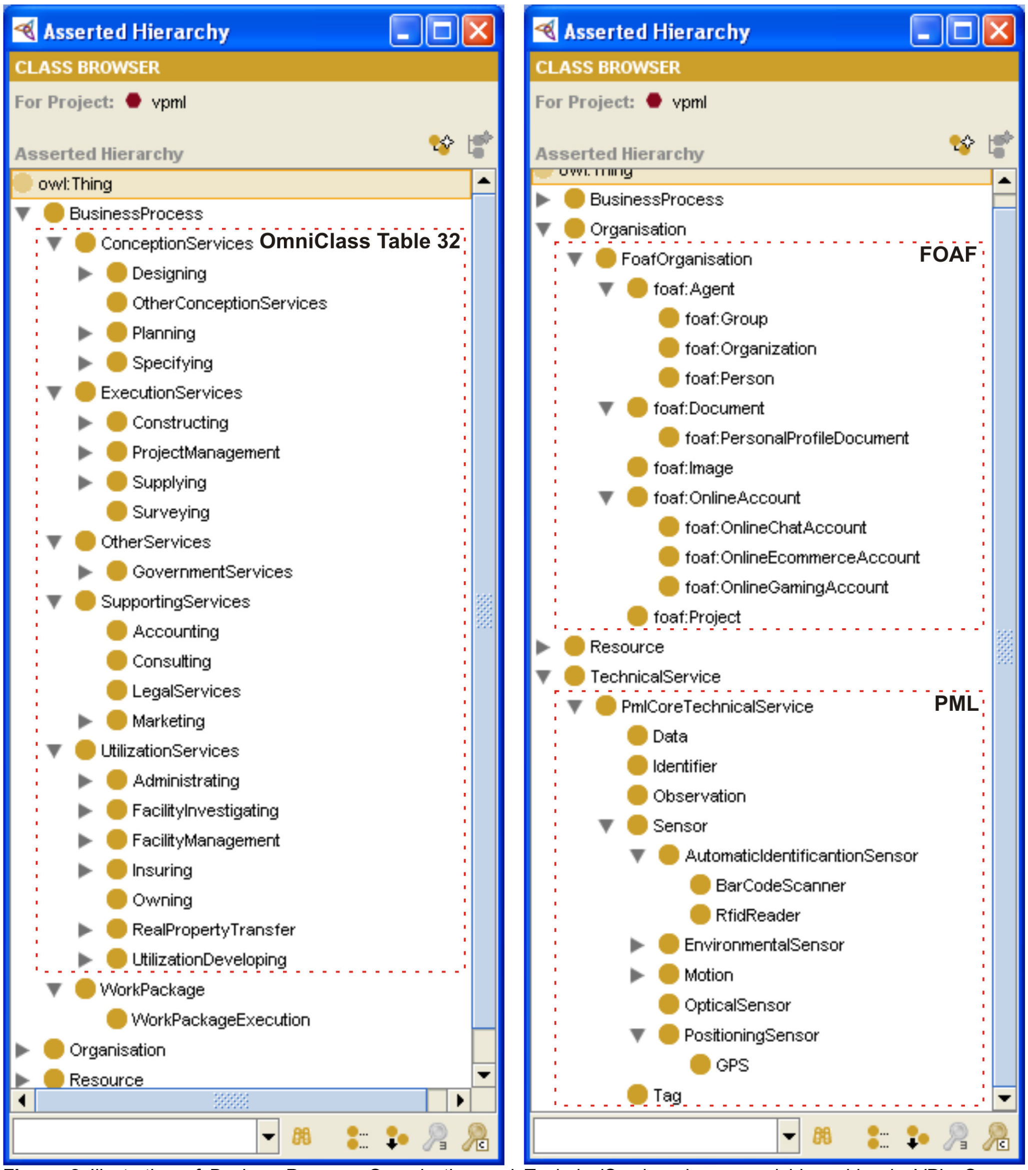

Figure 2 Illustration of BusinessProcess, Organisation and TechnicalService classes and hierarchies in VPL. Screen dumps of class browser in Protégé.

The OWL description of the IfcBeam class in VPL is shown below as an example of a class definition:

```
<owl:Class rdf:about="#IfcBeam">
  <owl:disjointWith rdf:resource="#IfcStairFlight"/>
  <owl:disjointWith rdf:resource="#IfcPlate"/>
```

```
  <owl:disjointWith rdf:resource="#IfcWindow"/>
  <owl:disjointWith rdf:resource="#IfcCurtainWall"/>
  <owl:disjointWith rdf:resource="#IfcDoor"/>
  <owl:disjointWith rdf:resource="#IfcRamp"/>
  <owl:disjointWith rdf:resource="#IfcBuildingElementComponent"/>
  <owl:disjointWith rdf:resource="#IfcFooting"/>
  <owl:disjointWith rdf:resource="#IfcRoof"/>
  <owl:disjointWith rdf:resource="#IfcMember"/>
  <owl:disjointWith rdf:resource="#IfcPile"/>
  <owl:disjointWith rdf:resource="#IfcColumn"/>
  <owl:disjointWith rdf:resource="#IfcCovering"/>
  <owl:disjointWith rdf:resource="#IfcRampFlight"/>
  <owl:disjointWith rdf:resource="#IfcStair"/>
  <owl:disjointWith rdf:resource="#IfcRailing"/>
  <owl:disjointWith rdf:resource="#IfcBuildingElementProxy"/>
  <owl:disjointWith rdf:resource="#IfcSlab"/>
  <rdfs:subClassOf>
    <owl:Class rdf:about="#IfcBuildingElement"/>
  </rdfs:subClassOf>
  <owl:disjointWith rdf:resource="#IfcWall"/>
</owl:Class>
```

3.5 Define the Properties of Classes – Slots

Properties of classes, also termed slots, describe the internal structure of the concepts. Object properties describe the relationship between the classes, and datatype properties are simple literals (e.g. an Integer, String, and others). In Figure 3, the screen dump from Protégé shows how the hasIdentifier property is used to define the relationship between instances of the Identifier class and several other classes such as the IfcClassificationReference, WorkPackage, Sensor, Tag etc. The IFC Entities (classes and subclasses) of IfcRelationship are in this "first try" implemented as object properties rather than classes, and define the relationship internally in the IFC data structure as well as the link to associated resources and processes. To recap from Paper I, the IfcRelAssociatesClassification relationship is used to link building components described in IFC with identification numbers of RFID tags. It is common to place a verb as prefix in object property names such as "has" or "is" but for the FOAF, IFC and Dublin Core ontologies the orginal names were kept.

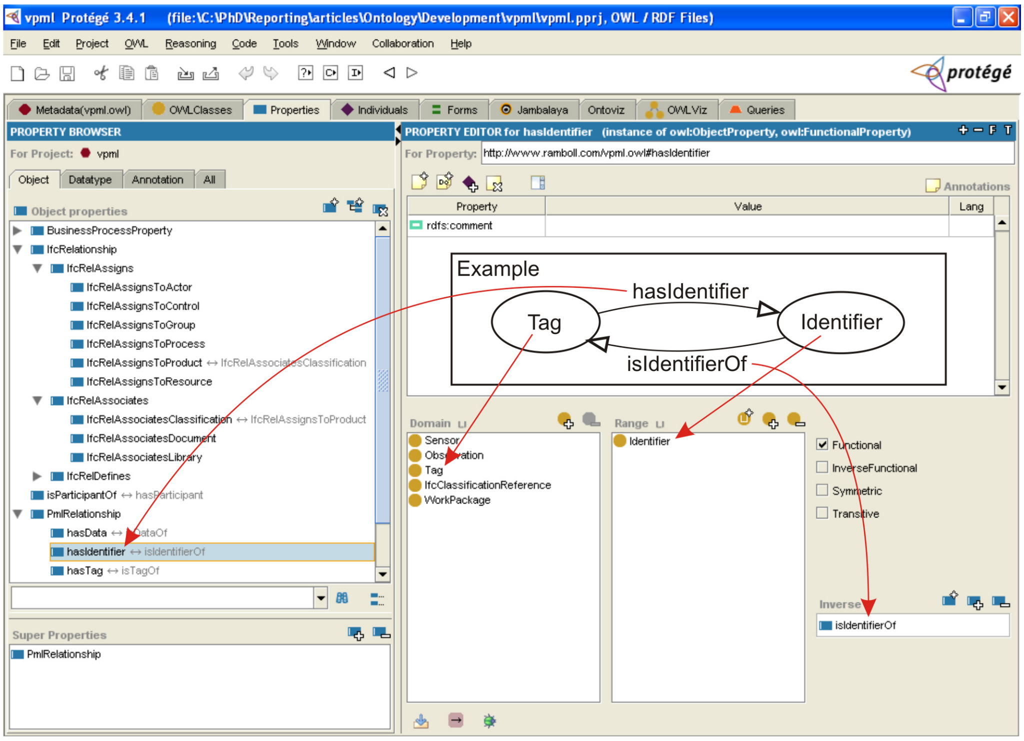

Figure 3 Illustration of object property definition in VPL. Screen dump of properties tab in Protégé.

The OWL description of the hasIdentifier in VPL is shown below as an example of a property definition:

```
<owl:FunctionalProperty rdf:ID="hasIdentifier">
  <owl:inverseOf>
    <owl:InverseFunctionalProperty rdf:ID="isIdentifierOf"/>
  </owl:inverseOf>
  <rdfs:subPropertyOf rdf:resource="#PmlRelationship"/>
  <rdfs:range rdf:resource="#Identifier"/>
  <rdfs:domain>
    <owl:Class>
      <owl:unionOf rdf:parseType="Collection">
        <owl:Class rdf:about="#Sensor"/>
        <owl:Class rdf:about="#Observation"/>
        <owl:Class rdf:about="#Tag"/>
        <owl:Class rdf:about="#IfcClassificationReference"/>
        <owl:Class rdf:about="#WorkPackage"/>
      </owl:unionOf>
    </owl:Class>
```

```
  </rdfs:domain>
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#ObjectProperty"/>
</owl:FunctionalProperty>
```

3.6 Define the Facets of the Slots

Facets of slots define restrictions of properties such as allowed values and other features properties can have. It was beyond the resources of this project to define facets of the slots.

3.7 Test by Creating Instances

Protégé enables the developer to populate the ontology with data (instances of classes). It actually makes Protégé the core of a complete knowledge management system, but it is probably mostly used for testing and evsaluation purposes. It is not recommended to make such a mix of class definitions and instances of classes in the same OWL-file but it can be useful for testing. In this development a small dataset was created for demonstration purposes. A part of these instances are shown in Table 1.

Table 1 Instances used for the evaluation of VPL.

Identifier			
Instance name	**hasID**	**isIdentifierOf**	
Identifier_12		IfcClassitionReference_14	
		WorkPackageExecution_17	
Identifier_14	"urn:epc:1:304:769"	Sensor_44	
Identifier_45	"urn:epc:1:304:768"	IfcClassificationReference_13	
Identifier_21	"urn:epc:1:304:770"	IfcClassificationReference_20	
Identifier_25	"urn:epc:1:304:771"	IfcClassificationReference_24	
Identifier_27	"urn:epc:1:304:772"	IfcClassificationReference_26	
Identifier_28		IfcClassitionReference_29	
		WorkPackageExecution_5	
Identifier_31	"urn:epc:2:102:356"	IfcClassificationReference_30	
Identifier_33	"urn:epc:2:102:911"	IfcClassificationReference_32	
Identifier_34		IfcClassificationReference_35	
		WorkPackageExecution_6	
Identifier_39	"urn:epc:3:007:996"	IfcClassificationReference_38	
Identifier_41	"urn:epc:3:007:997"	IfcClassificationReference_40	
Sensor			
Instance name	**has_Identifier**		
Sensor_44	Identifier_14		
Person			
Instance name	**firstName**	**isParticipantOf**	
Person_10	"Kristian"	WorkPackageExecution_6	
Person_11	"Michael"	WorkPackageExecution_17	
Person_18	"Peter"	WorkPackageExecution_17	
		WorkPackageExecution_5	
Person_7	"John"	WorkPackageExecution_5	
Person_8	"Jane"	WorkPackageExecution_6	
Person_9	"Paul"	WorkPackageExecution_6	
IfcBuildingElement			
Instance_Name	**hasName**	**hasDescription**	**IfcRelAssociatesClassification**
IfcDoor_12			IfcClassificationReference_13
			IfcClassificationReference_14
IfcDoor_16			IfcClassificationReference_20
			IfcClassificationReference_14

IfcDoor_17			IfcClassitionReference_24
			IfcClassificationReference_14
IfcDoor_19			IfcClassificationReference_26
			IfcClassificationReference_14
IfcWindow_22	"Window22"	"This is window 22"	IfcClassitionReference_29
			IfcClassificationReference_30
IfcWindow_23	"Window23"	"This is window 23"	IfcClassitionReference_29
			IfcClassificationReference_32
IfcWall_36	"Wall 36"		IfcClassificationReference_35
			IfcClassificationReference_38
IfcWall_37	"Wall 37"		IfcClassificationReference_35
			IfcClassificationReference_40
IfcClassificationReference			
Instance_Name	**hasIdentifier**	**IfcRelAssignsToProduct**	
IfcClassificationReference_13	Identifier_45	IfcDoor_12	
IfcClassificationReference_14	Identifier_12	IfcDoor_12	
		IfcDoor_16	
		IfcDoor_17	
		IfcDoor_19	
IfcClassitionReference_20	Identifier_21	IfcDoor_16	
IfcClassitionReference_24	Identifier_25	IfcDoor_17	
IfcClassificationReference_26	Identifier_27	IfcDoor_19	
IfcClassitionReference_29	Identifier_28	IfcWindow_22	
		IfcWindow_23	
IfcClassificationReference_30	Identifier_31	IfcWindow_22	
IfcClassificationReference_32	Identifier_33	IfcWindow_23	
IfcClassificationReference_35	Identifier_34	IfcWall_36	
		IfcWall_37	
IfcClassificationReference_38	Identifier_39	IfcWall_36	
IfcClassificationReference_40	Identifier_41	IfcWall_37	
WorkPackageExecution			
Instance_Name	**hasIdentifier**	**hasParticipant**	**hasMethod**
WorkPackageExecution_17	Identifier_12	Person_18	"Installation of doors 1st floor"
		Person_11	
WorkPackageExecution_5	Identifier_28	Person_18	"Installation of windows 1st floor"
		Person_7	
WorkPackageExecution_6	Identifier_34	Person_8	"Installation of prefab walls"
		Person_9	
		Person_10	

It requires some effort to immediately overview how the instances in Table 1 are related. This is better illustrated with Figure 4. The figure illustrates how the VPL ontology can be used to retrieve information about a work process when the craftsman “Peter” reads an RFID tag with his mobile phone. The RFID tag with the identification number “1:304:7682” in the EPC namespace is used to identify IfcDoor_12, with the related WorkPackageExecution_17, the method “Installation of doors 1st floor” and “Peter” as participant.

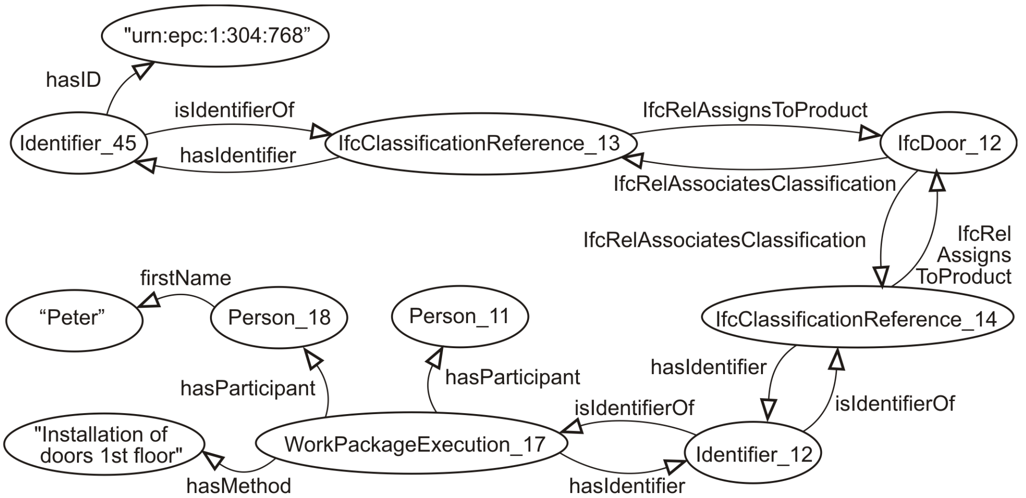

Figure 4 Illustration of how an ID of an RFID tag is related to a building component an instruction of a work method and a craftsman. Instance names are shown in the bubbles, object and datatype properties next to the arrow and string values in quotation marks (" ").

The OWL description of the IfcWall_36 in VPL is shown below as an example of an instance:

```
<IfcWall rdf:ID="IfcWall_36">
  <hasName rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >Wall 36</hasName>
  <IfcRelAssociatesClassification rdf:resource="#IfcClassificationReference_35"/>
  <IfcRelAssociatesClassification>
    <IfcClassificationReference rdf:ID="IfcClassificationReference_38">
      <hasIdentifier>
        <Identifier rdf:ID="Identifier_39">
          <hasID rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
          >urn:epc:3:007:996</hasID>
          <isIdentifierOf rdf:resource="#IfcClassificationReference_38"/>
        </Identifier>
      </hasIdentifier>
      <IfcRelAssignsToProduct rdf:resource="#IfcWall_36"/>
    </IfcClassificationReference>
  </IfcRelAssociatesClassification>
</IfcWall>
```

3.7.1 Demonstration of Ontology Usage

A "quick and dirty" method for evaluating the ontology is to populate an ICT system based on the ontology with some instances and run queries to answer the competency questions listed in Section 3.1.1 of this appendix. In Protégé it can be done either with the query tool illustrated in Figure 5 or by executing SPARQL Queries (Prud'hommeaux et al., 2008). The query tool was primarily used in this project and a set of queries was defined to answer the competency question:

“Which working tasks are related to component XX and performed by person YY?” The queries described in Table 2 were defined to answer the question.

Table 2 Queries to answer the competency question: ”Which working tasks are related to component urn:epc:1:304:7682 and performed by Peter?”

Query name	**Class**	**Slot**	**Expression**	**Value/Query**	**Returns**
Find Identifier	Identifier	hasID	contains	**“urn:epc:1:304:7682”**	Identifier_45
Find IfcClas.Ref. for Identifier	IfcClas.Ref.	hasIdentifier	contains	Find Identifier	IfcClas.Ref._13
Find Product	IfcProduct	IfcRelAssociatesClassification	contains	Find IfcClas.Ref. for Identifier	IfcDoor_12
Find Person	Foaf:Person	firstName	contains	**“Peter”**	Person_18
Find IfcClas.Ref. assigned to Product	IfcClas.Ref.	IfcRelAssignsToProduct	contains	Find Product	IfcClas.Ref._13 and IfcClas.Ref._14
Find Identifier of WorkPackage	Identifier	isIdentifierOf	contains	Find IfcClas. Ref. assigned to Product	Identifier_45 and Identifier_12
Find WorkPackage for Component and Person	WorkPackageExecution WorkPackageExecution	hasIdentifier hasParticipant	contains contains	Find Identifier of WorkPackage Find Person	**WorkPackageExecution_17**

A SPARQL expression equivalent to the first query in Table 2 can be written as:

```
PREFIX model: <http://www.ramboll.com/VPL.owl#>
SELECT ?Indentifier
From <http://www.ramboll.com/VPL.owl>
WHERE { ?Indentifier model:hasID  "urn:epc:1:304:768"}
```

The execution of the queries is illustrated in Figure 4.

Another possible usage of the VPL ontology is implementation in description logic based expert systems. This is supported in Protégé by means of the Semantic Web Rule Language (SWRL) or e.g. by translating the VPL ontology into Prolog, see Samuel et al. (2007).

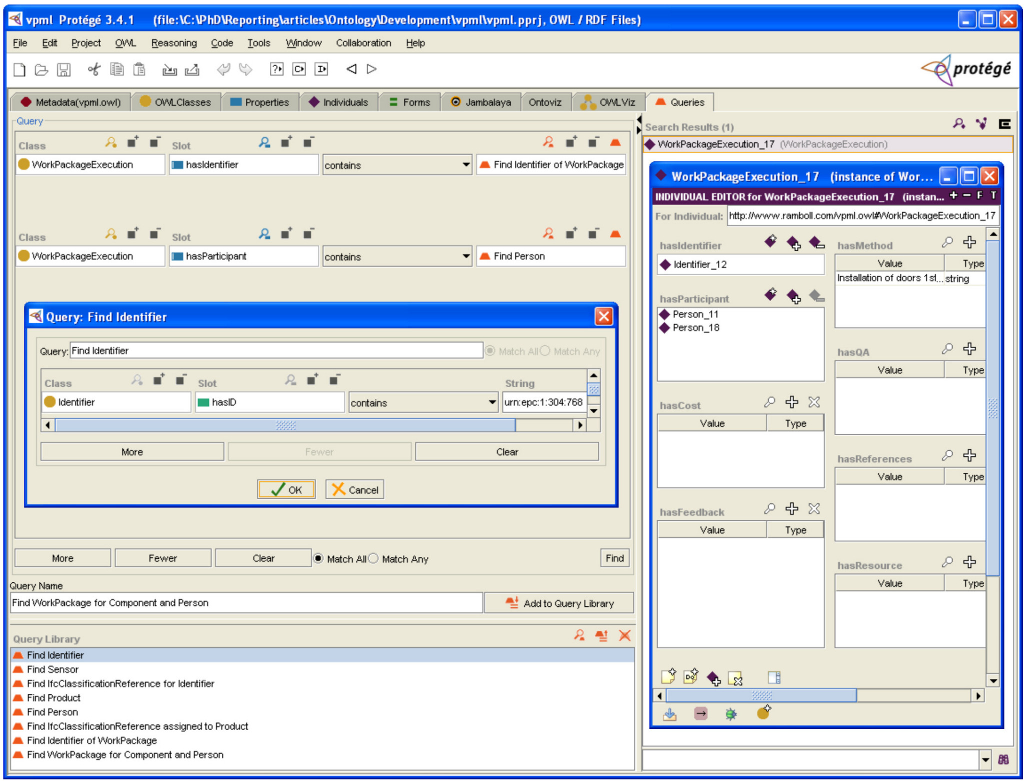

Figure 5 Illustration of queries in Protégé to answer the competency question: "Which working tasks are related to component urn:epc:1:304:7682 and performed by Peter?"

4 Conclusion

ICT systems to support the scenarios described in Paper I and Paper II can be developed as proprietary solutions where the data capture services extract values from the virtual models on basis of the software providers APIs. This approach may lead to situations were users become locked to software providers or reuse of information across tools, and disciplines becomes infeasible. Methods, tools and recommendations on an ontology based approach to overcome these challenges were demonstrated in this paper. However, a significant amount of work is still required before a full-fledged Virtual Physical Markup Language ontology is developed and implemented in the construction industry.

5 References

Beetz, J., Leeuwen, J.V., Vries, B.D (2009). IfcOWL: A case of transforming EXPRESS schemas into ontologies, Artificial Intelligence for Engineering Design, Analysis and Manufacturing (2009), 23, pp. 89–101.

Christensen, E., Curbera, F., Meredith, G., Weerawarana, S. (2001). Web Services Description Language (WSDL) 1.1, W3C Note 15.

DCMI (2008). Dublin Core Metadata Element Set, Version 1.1, Dublin Core Metadata Initiative (DCMI), available at http://dublincore.org/documents/dces/

El-Diraby, T.E., Lima, C., Fies, B. (2005). Strategies for Incorporating Data Exchange Standards in E-business Taxonomies. Journal of Information Technology Management Volume XVI, Number 4, pp. 26-38.

Floerkemeier, C, Anarkat, D., Osinski, T., Harrison, M. (2003). PML Core Specification 1.0, Auto-ID Center Recommendation 15 September 2003, available at http://develop.autoidcenter.org/

FOAF (2007). FOAF Vocabulary Specification 0.9, Namespace Document 24 May 2007 - *'Rehydrated' Edition,* available at http://xmlns.com/foaf/0.1/.

Gasevic D., Djuric, D., Devedzic, V. (2006). Model Driven Architecture and Ontology Development, Springer-Verlag Berlin Heidelberg, 311 pp.

Grünninger and Fox (1995): Methodology for the Design and Evaluation on Ontologies. Department of Industrial Engineering, University of Toronto.

Gruber, Tom (1993). A translation approach to portable ontology specifications, Knowledge Acquisition 5, pp. 199-220.

Horridge, M. (2009). A Practical Guide To Building OWL Ontologies Using Protégé 4 and CO-ODE Tools, Edition 1.2, The University of Manchester, 109 pp.

IAI (2006). IFC2x Edition 3, Online documentation, available at http://www.iai-international.org/, (accessed November 2006, currently moving to http://www.buildingsmart.com/)

Martin, D., Burstein, M., Hobbs, J., Lassila, O, McDermott, D., McIlraith, S., Narayanan, S., Paolucci, M., Parsia, B., Payne, T., Sirin, E., Srinivasan, N., Sycara, K. (2004). OWL-S: Semantic Markup for Web Services, W3C Member Submission 22 November 2004, available at http://www.w3.org/Submission/OWL-S/.

Noy, N. F., McGuinness, D.L. (2001). Ontology Development 101: A Guide to Creating Your First Ontology. Stanford University, 25 pp.

OCCS (2006). OmniClass Construction Classification System, OCCS Development Committee Secretariat, available at http://www.omniclass.org.

Prud'hommeaux, E., Seaborne, A. (2008). SPARQL Query Language for RDF, W3C Recommendation 15 January 2008, available at http://www.w3.org/TR/rdf-sparql-query/

Samuel, K., Obrst, L, Stoutenberg, S., Fox, K., Franklin, P., Johnson, A., Laskey, K., Nichols, D., Lopez, S., Peterson, J. (2007). Translating OWL and Semantic Web Rules into Prolog: Moving Toward Description Logic Programs, The MITRE Corporation.

S-TEN (2008). Web-site of the S-TEN Project: Intelligent Self-describing Technical and Environmental Networks, available at http://www.s-ten.eu.